



System i

การจัดการระบบ

การจัดการระบบงาน

เวอร์ชัน 6 รีลีส 1





System i

การจัดการระบบ

การจัดการระบบงาน

เวอร์ชัน 6 รีลีส 1

หมายเหตุ

ก่อนที่จะใช้ข้อมูลนี้และผลิตภัณฑ์ที่ข้อมูลนี้สนับสนุน โปรดอ่านข้อมูลใน “คำประกาศ”, ในหน้า 247

การพิมพ์ครั้งนี้ใช้ได้กับ IBM i5/OS เวอร์ชัน 6 รีลีส 1 โมดิฟิเคชัน 0 (หมายเลขผลิตภัณฑ์ 5761-SS1) และใช้กับรีลีสและโมดิฟิเคชันถัดจากนี้ไปจนกว่าจะมีการระบุเป็นอย่างอื่นในการพิมพ์ครั้งใหม่ เวอร์ชันนี้ไม่สามารถรันบนโมเดล reduced instruction set computer (RISC) และโมเดล CISC ได้ทุกรุ่น

© ลิขสิทธิ์ของ International Business Machines Corporation 2004, 2008. สงวนลิขสิทธิ์ทั้งหมด.

สารบัญ

การจัดการระบบงาน	1
ไฟล์ PDF สำหรับการจัดการระบบงาน	1
บทนำการจัดการระบบงาน	2
ระบบของคุณเปรียบเสมือนธุรกิจ	2
ช่วงชีวิตของงาน	3
การส่งงาน	4
งานเข้าสู่คิวงาน	4
งานเข้าสู่ระบบย่อย	4
ระบบย่อยใช้หน่วยความจำจากพูลหน่วยความจำเพื่อ รันงาน	5
งานเสร็จสิ้นและย้ายไปที่เอาต์พุตคิว	5
วิธีการทำงานให้เสร็จสมบูรณ์	6
งานคืออะไร	6
สิ่งที่เกิดขึ้นก่อนงานเข้าสู่ระบบ	6
งานเข้าสู่ระบบได้อย่างไร	6
งานถูกประมวลผลอย่างไร	7
งานออกจากระบบอย่างไร	7
หลักการ	8
โครงสร้างระบบของคุณ	8
ระบบย่อยที่ส่งมาพร้อมกับระบบ	8
โปรแกรมเริ่มต้น	9
สิ่งที่เกิดขึ้นในระหว่าง IPL	9
ชนิดของ start-ups	10
การปิดเครื่อง	10
System i Navigator	11
ระบบย่อย	11
ระบบย่อยการควบคุม	11
เพราะเหตุใดจึงพิจารณาการใช้หลายระบบย่อย	12
Subsystem description	13
Subsystem description attributes	13
Work entry	14
routing entry	17
วิธีการเริ่มระบบย่อย	21
วิธีการจัดสรรอุปกรณ์เวิร์กสเตชัน	22
สถานการณ์จำลอง: การจัดสรรเวิร์กสเตชัน	23
พูลหน่วยความจำ	24
ชนิดของพูลหน่วยความจำ	25
แบบแผนการกำหนดหมายเลขพูล	27
การจัดสรรพูลหน่วยความจำ	29
activity level ของพูลหน่วยความจำ	30
งาน	31
สิทธิในการใช้งานที่เหมาะสม	31

ลักษณะงาน	32
ไวยากรณ์ชื่องาน	32
แอตทริบิวต์งาน	33
รายละเอียดของงาน	33
รายละเอียดของงานและการรักษาความปลอดภัย	34
Call stacks	35
คลาสอ็อบเจกต์	35
Job user identity	37
ตัวอย่าง Job user identity	38
thread	38
อ็อบเจกต์ที่ถูกล็อก	40
ชนิดของงาน	41
งานแบบ autostart	41
งานแบตช์	42
งานสื่อสาร	43
งานแบบโต้ตอบ	45
งานแบบ prestart	51
งาน reader และงาน writer	57
งานเซิร์ฟเวอร์	57
งานระบบ	58
อ็อบชันการจัดตารางเวลางาน	62
Management Central Scheduler	62
Advanced Job Scheduler	63
Job schedule entries	63
ตัวอย่าง: job schedule entry	64
คำสั่งการส่งงาน	65
ข้อควรพิจารณาเกี่ยวกับตารางเวลางาน	66
การจัดตารางเวลางานและสภาพพร้อมใช้งานระบบ	67
คิวงาน	68
รายการที่จัดลำดับ	69
วิธีการทำงานของคิวงาน	69
วิธีการนำงานมาจากคิวงาน	70
Job queue entry	71
วิธีการจัดสรรคิวงานที่ระบบย่อย	71
หลายคิวงาน	72
วิธีการนำงานมาจากหลายคิวงาน	73
การรักษาความปลอดภัยคิวงาน	74
เอาต์พุตคิว	74
แอตทริบิวต์ของเอาต์พุตคิว	75
ลำดับของไฟล์	76
สพูลไฟล์	76
การสพูลเอาต์พุต	77
เอาต์พุตคิวและสพูลไฟล์	78

ดีพอลต์เอาต์พุตคิวของระบบ	78	การกำหนดจำนวนของระบบย่อยที่ใช้พูลหน่วยความ จำ	116
Spooling writers	79	System i Navigator	116
คำสั่ง spooling writer	79	อินเทอร์เฟซแบบอักษร	116
Input spooling	80	การดูสถิติประสิทธิภาพของงาน	116
คำสั่งของการอินพุตงาน	81	การดูสถานะระบบโดยรวม	117
ไฟล์ข้อมูลออนไลน์	82	การตรวจสอบสถานะของดิสก์	117
ข้อควรพิจารณาในการเปิดไฟล์ข้อมูลออนไลน์	83	การจัดการงาน	118
บันทึกการใช้งาน	84	ภารกิจงานทั่วไป	118
วิธีการสร้างบันทึกการใช้งาน	85	การเริ่มงาน	118
การคงค้างบันทึกการใช้งาน	86	การจบงาน	119
เซิร์ฟเวอร์บันทึกการใช้งาน	87	การหางาน	121
ลักษณะการแสดงของบันทึกการใช้งาน	88	การดูงานบนคิวงาน	122
หัวข้อบันทึกการใช้งาน	88	การดูงานบนระบบย่อย	123
ข้อความ	89	การดูแอ็ททริบิวต์งาน	123
ไฟล์บันทึกงานแบบโต้ตอบ	90	การดู call stack	124
บันทึกประวัติ (ที่ผ่านมา) QHST.	91	การวางแผนบนคิวงาน	125
การจัดรูปแบบบันทึกประวัติ (ที่ผ่านมา)	91	การย้ายงานไปที่คิวงานอื่น	125
ข้อมูลประสิทธิภาพและ QHST	93	การเลื่อนระดับความสำคัญของงานขึ้นภายในคิว งาน	126
สพูลไฟล์	94	คำแนะนำสำหรับการตั้งค่าระดับความสำคัญงาน	127
Job accounting.	94	การส่งงานครั้งเดียว	127
วิธีการทำงานของ job accounting	94	การดูข้อมูล job affinity	128
ลักษณะการดำเนินการ Job Accounting	96	การจัดการรายละเอียดของงาน	128
การประมวลผล Accounting Journal.	96	การสร้างรายละเอียดของงาน	129
ใช้ job accounting เมื่อใด	97	การเปลี่ยนรายละเอียดของงาน	129
การรักษาความปลอดภัยและ job accounting	98	การใช้รายละเอียดของงาน	129
เกี่ยวกับ accounting code	99	การควบคุมซอร์สของแอ็ททริบิวต์งาน	130
Resource accounting	99	การลบรายละเอียดของงาน	131
ข้อมูล Resource accounting	100	การจัดการงานแบ็ตซ์	131
งานแบบ prestart และ job accounting	101	การส่งงานแบ็ตซ์	131
การประมวลผลงานระบบสำหรับ job accounting	102	การเริ่มงานแบ็ตซ์ที่กำลังรออยู่ในคิวงาน	133
การประมวลผลแบบแบ็ตซ์และ job accounting	102	การจัดการงานแบบโต้ตอบ	134
การประมวลผลแบบโต้ตอบและ job accounting	103	การควบคุมงาน และเวิร์กสเตชันที่ inactive	134
Printer file accounting	103	การจบงานแบบโต้ตอบ	135
Journal entry สำหรับ job accounting	104	การตัดการเชื่อมต่องานทั้งหมดจากอุปกรณ์	136
ข้อมูลฟิลด์ Job accounting journal entry	104	ข้อควรพิจารณาการตัดการเชื่อมต่องาน	136
Printer file accounting data สำหรับ direct print และ spooled print	108	การหลีกเลี่ยงฟังก์ชันที่ใช้เวลารันนานจากเวิร์กส เตชัน	137
การจัดการงาน	110	การจัดการงานแบบ prestart	137
การเรียกโปรแกรมการกู้คืน IPL พิเศษ	111	การเริ่มงานแบบ prestart	138
การมอนิเตอร์ system ระบบ	111	การจัดคิว หรือการปฏิเสธคำขอการเริ่มโปรแกรม	138
การตรวจสอบการใช้พูลหน่วยความจำ	112	การปรับ prestart job entry	139
การควบคุมระดับของ activity ระบบ	112	การเปลี่ยนแอ็ททริบิวต์งานสำหรับงานแบบ prestart	142
ตัวอย่าง: ความสัมพันธ์ของการควบคุม activity	114	การจบงานแบบ prestart job	144
การกำหนดสถานะของงาน	115	การจัดการคลาสอ็อบจ็กต์งาน	144
การมอนิเตอร์ระบบย่อย	115		
System i Navigator	115		

การสร้างคลาสอ็อบเจ็กต์	144	การลบ job queue entry	191
การเปลี่ยนคลาสอ็อบเจ็กต์	145	การลบ prestart job entry	192
การจัดการ thread	145	การลบ routing entry.	192
การดู thread ที่กำลังรันภายใต้งานเฉพาะ	145	การลบ workstation entry	193
คุณสามารถทำอะไรกับ thread	146	การตั้งค่าระบบย่อยแบบโต้ตอบ	193
การดูคุณสมบัติของ thread	147	การสร้างไลบรารี.	193
การจบ หรือการลบ thread	148	การสร้างคลาส	194
การจัดการการจัดตารางเวลา	149	การสร้าง subsystem description	194
การจัดการตารางเวลาโดยใช้ System i Navigator	149	การสร้างคิวงาน	194
การจัดการตารางเวลาโดยใช้ Management Central Scheduler	149	การใส่เพิ่ม routing entry	194
Advanced Job Scheduler	150	การใส่เพิ่ม workstation entry	195
Advanced Job Scheduler for Wireless.	150	การปรับ QINTER	195
การจัดการตารางเวลาด้วย Advanced Job Scheduler	150	การตั้งค่าคอนโซล	196
การทำงานกับ job schedule entries.	175	การกำหนดผู้ใช้ที่ระบบย่อยเฉพาะ.	196
การใส่เพิ่ม job schedule entry	175	การสร้างระบบย่อยการควบคุม.	198
การเปลี่ยน job schedule entry	175	การวางระบบในสภาพที่ถูกจำกัด	199
การพัก job schedule entry	176	การจัดการพูลหน่วยความจำ	199
การพิมพ์รายการของ job scheduled entry.	176	การดูข้อมูลพูลหน่วยความจำ	200
การรีลีส job schedule entry	177	System i Navigator	200
การลบ job schedule entry	177	อินเตอร์เฟซแบบอักษร	200
การจัดการระบบย่อย	177	การกำหนดจำนวนของระบบย่อยที่ใช้พูลหน่วยความจำ	201
ภารกิจระบบย่อยทั่วไป	178	System i Navigator	201
การดูแอ็ททริบิวต์ระบบย่อย	178	อินเตอร์เฟซแบบอักษร	201
การหยุดระบบย่อย	178	การกำหนดจำนวนของงานในพูลหน่วยความจำ	201
การเริ่มระบบย่อย	180	การกำหนดพูลที่งานเดี่ยวกำลังรันอยู่	202
การสร้างรายละเอียดของระบบย่อย	181	System i Navigator	202
การใส่เพิ่ม autostart job entries.	182	การจัดพารามิเตอร์การปรับสำหรับแชร์พูล	203
การใส่เพิ่ม communications entries	182	System i Navigator	203
การใส่เพิ่ม job queue entries	183	อินเตอร์เฟซแบบอักษร	203
การใส่เพิ่ม prestart job entries	183	การจัดการ configuration ของพูล	204
การใส่เพิ่ม routing entries	184	System i Navigator	204
การใส่เพิ่ม workstation entries.	184	อินเตอร์เฟซแบบอักษร	204
การสร้างไฟล์แสดงผล sign-on.	185	การเปลี่ยนขนาดของพูลหน่วยความจำ	204
การระบุการแสดงผล sign-on ใหม่	186	System i Navigator	205
การเปลี่ยนรายละเอียดของระบบย่อย	186	อินเตอร์เฟซแบบอักษร	205
การเปลี่ยน autostart job entries	187	การเปลี่ยนขนาดของ shared pool	206
การเปลี่ยน communication entries	187	การสร้างพูลหน่วยความจำส่วนตัว.	206
การเปลี่ยน job queue entry	188	การจัดการคิวงาน	207
การเปลี่ยน prestart entry	188	การกำหนดคิวงานที่ระบบย่อย	207
การเปลี่ยน routing entry	189	ระบบย่อยจัดการกับหลายคิวงานได้อย่างไร	207
การเปลี่ยน workstation entry	189	การเปลี่ยนจำนวนของงานที่รันได้พร้อมกันในคิวงาน	208
การเปลี่ยนการแสดงผล sign-on	190	การลบคิวงาน	209
การลบรายละเอียดของระบบย่อย	190	System i Navigator	209
การลบ autostart job entry	191	อินเตอร์เฟซแบบอักษร	209
การลบ communication entry	191	การสร้างคิวงาน	209
		การลบคิวงาน	210

การกำหนดวาระบทย่อยใดมีคิวงานที่ถูกจัดสรร	210	วิธีการแสดงบันทึกการใช้งาน	220
System i Navigator	211	System i Navigator	220
อินเทอร์เน็ตเฟสแบบอักษร	211	จะทำอะไรเมื่อบันทึกการใช้งานไม่แสดงขึ้น	221
การพักคิวงาน	211	การระบุเอาต์พุตคิวงานสำหรับบันทึกการใช้งาน	222
System i Navigator	211	การหยุดการจัดทำบันทึกการใช้งานเฉพาะ	223
การรีเซ็ตคิวงาน	211	การป้องกันการจัดทำบันทึกการใช้งาน	223
System i Navigator	212	การควบคุมข้อมูลในบันทึกการใช้งาน	224
อินเทอร์เน็ตเฟสแบบอักษร	212	การเปลี่ยนระดับไฟล์บันทึกของงาน	225
การย้ายงานไปที่คิวงานอื่น	212	การควบคุมข้อมูลบันทึกการใช้งานแบ็คซ์	225
System i Navigator	212	การลบไฟล์เอาต์พุตบันทึกการใช้งาน	226
การวางแผนคิวงาน	213	การจัดทำพรินเตอร์เอาต์พุตจากบันทึกการใช้งานที่	
System i Navigator	213	คงค้าง	227
อินเทอร์เน็ตเฟสแบบอักษร	213	การล้างบันทึกใช้งานที่คงค้าง	228
การค้นหาคิวงานทั้งหมดสำหรับงานเฉพาะ	213	System i Navigator	228
System i Navigator	213	อินเทอร์เน็ตเฟสแบบอักษร	228
อินเทอร์เน็ตเฟสแบบอักษร	214	การจัดการ job accounting	229
การค้นหาคิวงานถ้าคุณไม่ทราบชื่อของคิวงาน	214	การตั้งค่า job accounting	229
การระบุระดับความสำคัญสำหรับคิวงาน	214	การควบคุมการกำหนดได้คิวงาน	230
การจัดการเอาต์พุตคิวงาน	214	การแสดงผลข้อมูลที่รวบรวม	231
การสร้างเอาต์พุตคิวงาน	215	การแปลง job accounting journal entry	231
การกำหนดเอาต์พุตคิวงานที่งาน หรือรายละเอียดของ		การกู้คืน และ job accounting	233
งาน	215	job accounting journal หรือ journal receiver ที่	
System i Navigator	215	เสียหาย	234
อินเทอร์เน็ตเฟสแบบอักษร	215	การเข้าใช้งานข้อความ CPF1303	234
การเข้าใช้งานพรินเตอร์เอาต์พุต	216	การอ้างอิง	235
System i Navigator	216	Group job	235
การลบเอาต์พุตคิวงาน	216	โปรแกรม Attention key handling	239
System i Navigator	216	เคล็ดลับประสิทธิภาพงานกลุ่ม	241
อินเทอร์เน็ตเฟสแบบอักษร	216	การแก้ปัญหาสำหรับการจัดการระบบงาน	241
การลบเอาต์พุตคิวงาน	216	งานของเงินหยุดทำงาน	241
การดูเอาต์พุตคิวงานระบบ	217	งานของเงินประสบปัญหาทำงานช้า	243
การจัดการบันทึกการใช้งาน	217	การสืบสวนงานแบบ prestart	244
การจัดการเซิร์ฟเวอร์บันทึกการใช้งาน	217	ข้อมูลที่เกี่ยวข้องกับการจัดการระบบงาน	246
Reconfiguring เซิร์ฟเวอร์บันทึกการใช้งาน	218	ภาคผนวก. คำประกาศ	247
การจบเซิร์ฟเวอร์บันทึกการใช้งาน	218	เครื่องหมายการค้า	249
การเริ่มเซิร์ฟเวอร์บันทึกการใช้งาน	219	ข้อกำหนดและเงื่อนไข	249
System i Navigator	219		
อินเทอร์เน็ตเฟสแบบอักษร	220		

การจัดการระบบงาน

การจัดการระบบงานเป็นพื้นฐานสำคัญภายในระบบปฏิบัติการ i5/OS®

ฟังก์ชันของมันเป็นพื้นฐานที่งานทั้งหมดถูกป้อนเข้าสู่ระบบ, ประมวลผล, รัน และแล้วเสร็จบนผลิตภัณฑ์ System i™ Navigator ไม่ว่าคุณจะรัน งานแบ็คซ์แบบธรรมดาสัปดาห์ละครั้งหรือเรียกใช้แอปพลิเคชันประจำวัน (เช่น Lotus Notes®) การจัดการระบบงานช่วยในการบริหารงานและอ็อบเจกต์ต่างๆ ที่รันบนระบบของคุณ นอกจากนี้ ยังสนับสนุนคำสั่งและฟังก์ชันภายในที่จำเป็นในการควบคุมการปฏิบัติการของระบบ และจัดสรรรีซอร์สให้กับแอปพลิเคชันเมื่อต้องการอีกด้วย

ผลิตภัณฑ์ System i ได้รับการตั้งค่า และพร้อมใช้ ผู้ใช้ส่วนใหญ่ไม่จำเป็นต้องเปลี่ยนการตั้งค่าที่กำหนดไว้ อย่างไรก็ตาม หากคุณจำเป็นต้องปรับเปลี่ยนส่วนการจัดการระบบงานให้เหมาะสมกับบริษัทของคุณ คุณจำเป็นต้องเข้าใจคำศัพท์และแนวคิดที่เกี่ยวข้องรวมทั้งวิธีผสมผสานเข้าด้วยกัน เพื่อให้คุณได้ผลการปฏิบัติงานที่ดีที่สุดจากระบบของคุณ

ไม่ว่าคุณจะมีประสบการณ์ในฐานะ System i ผู้ใช้หรือเพิ่งหัดใช้ก็ตาม กลุ่มหัวข้อนี้ช่วยให้คุณเข้าใจภาพการจัดการระบบที่เข้าใจง่าย หัวข้อนี้มี entry point ต่างๆ เพื่อให้คุณสามารถเลือกจุดเพื่อเริ่มต้นเรียนรู้เกี่ยวกับการจัดการระบบงาน

หมายเหตุ: นอกจากนี้ คุณสามารถทำงานกับการจัดการระบบงานด้วยการใช้ภารกิจต่างๆ ของ System i Navigator บนเว็บ สิ่งนี้ช่วยให้คุณสามารถทำงานกับฟังก์ชันการจัดการระบบงานโดยใช้เว็บเบราว์เซอร์ได้ สำหรับข้อมูลเพิ่มเติมให้ดูที่ภารกิจต่างๆ ของ System i Navigator บนเว็บ

ไฟล์ PDF สำหรับการจัดการระบบงาน

คุณสามารถดู และพิมพ์ไฟล์ PDF ของข้อมูลนี้

- ในการดู หรือดาวน์โหลดเวอร์ชัน PDF ของเอกสารนี้ ให้เลือก การจัดการระบบงาน (ประมาณ 3300 KB).
- ในการดู หรือดาวน์โหลดเวอร์ชัน PDF ของบางส่วนของ Advanced Job Scheduler ของการจัดการระบบงาน ให้เลือก Advanced job scheduler (ประมาณ 390 KB)

คุณสามารถดูหรือดาวน์โหลดหัวข้อที่เกี่ยวข้องเหล่านี้:

- ประสิทธิภาพ (2000 KB) มีหัวข้อดังต่อไปนี้:
 - การวางแผนสำหรับประสิทธิภาพ
 - การจัดการประสิทธิภาพระบบ
 - แอปพลิเคชันสำหรับการจัดการประสิทธิภาพ
- Management Central (2500 KB) ประกอบด้วยข้อมูลที่ช่วยให้คุณดำเนินการภารกิจในการจัดการระบบในระบบเดียวหรือมากกว่าในเวลาเดียวกัน

การบันทึกไฟล์ PDF


ถ้าต้องการบันทึก PDF บนเวิร์กสเดชันของคุณ สำหรับการดูและการพิมพ์ให้ทำดังนี้:

1. คลิกขวาที่ลิงก์ PDF ในเบราว์เซอร์ของคุณ
2. คลิกอ็อปชันที่จะบันทึก PDF แบบโลคัล

3. นำทางไปที่ไดเร็กทอรีซึ่งคุณต้องการบันทึก PDF
4. คลิก Save.

การดาวน์โหลด Adobe Reader

คุณต้องติดตั้ง Adobe® Reader บนระบบของคุณเพื่อดู หรือพิมพ์ PDF เหล่านี้ คุณสามารถดาวน์โหลดก็อปปีฟรีจาก Adobe

Web site (www.adobe.com/products/acrobat/readstep.html) 

บทนำการจัดการระบบงาน

การจัดการระบบงานสนับสนุนคำสั่งและฟังก์ชันภายในที่จำเป็น สำหรับการควบคุมการดำเนินการระบบและเวิร์กโฟลด์ประจำวันบนระบบ. นอกจากนี้, การจัดการระบบงานยังมีฟังก์ชันที่คุณต้องใช้ในการแจกจ่ายรีซอร์สสำหรับแอปพลิเคชันของคุณ เพื่อให้ระบบของคุณสามารถจัดการแอปพลิเคชันได้.

วัตถุประสงค์ของระบบของคุณคือการปฏิบัติงาน. งานเข้าสู่, งานถูกประมวลผล, และงานออกจากระบบ. ถ้าคุณคิดถึงการจัดการระบบงานเป็นสามระยะเหล่านี้, จะทำให้เข้าใจการจัดการระบบงานได้ง่ายขึ้น. การจัดการระบบงานอธิบายว่า งานจะเข้าสู่ระบบที่ใด, งานจะถูกประมวลผลที่ใดและด้วยรีซอร์สใด, และเอาต์พุตจากงานจะไปที่ไหน.

คุณยังไม่เคยใช้การจัดการระบบงานใช่หรือไม่? กลุ่มหัวข้อข้างใต้เรื่อง บทนำการจัดการระบบงาน ได้รับการออกแบบมาเพื่อนำเสนอข้อมูลภาพรวมในด้านต่างๆ ของการจัดการระบบงาน. ในวิธีนี้ ช่วยให้คุณสามารถ เข้าใจพื้นฐานในหลักการของการจัดการระบบงาน ไม่ว่าคุณจะมีแบ็คกราวด์เกี่ยวกับระบบมากเพียงใดก็ตาม

ระบบของคุณเปรียบเสมือนธุรกิจ

เพื่อให้เข้าใจหลักการโดยรวมของการจัดการงานได้ง่ายขึ้น ให้ลองเปรียบเทียบระบบของคุณเป็นเสมือนธุรกิจอย่างหนึ่ง

ระบบพื้นฐานสามารถเปรียบเป็นธุรกิจขนาดเล็ก และระบบที่ซับซ้อนสามารถ เปรียบได้เป็นห้างสรรพสินค้า สมมุติว่ามีร้านค้าขนาดเล็กที่ทำธุรกิจ การประกอบเฟอร์นิเจอร์ไม้แกะสลัก *งานที่เข้าสู่ระบบ*, เช่น ไม้สั่งซื้อ โต๊ะขนาดเล็ก, เก้าอี้, และชั้นหนังสือ *งานที่ประมวลผล* ช่างไม้โทรหาลูกค้า เพื่อให้รับรองไม้สั่งซื้อ และปรึกษเกี่ยวกับการออกแบบ เช่น สีไม้, ขนาด, และสีข้างไม้ ออกแบบเฟอร์นิเจอร์แต่ละชิ้น รวบรวมวัสดุที่จำเป็น, แล้วผลิตเฟอร์นิเจอร์ หลังจากเฟอร์นิเจอร์เสร็จสมบูรณ์แล้ว เฟอร์นิเจอร์จะถูกจัดส่งให้ลูกค้า: *งานออกจากระบบ*.

เนื่องจากระบบที่ซับซ้อนคือระบบที่ประกอบไปด้วยระบบพื้นฐานหลายระบบ สิ่งที่เปรียบเทียบได้กับระบบที่ซับซ้อนคือห้างสรรพสินค้า ซึ่งมีธุรกิจขนาดเล็กและขนาดใหญ่หลายธุรกิจในพื้นที่หนึ่ง อาจเป็นไปได้ที่ช่างไม้มีร้านทำเฟอร์นิเจอร์อยู่ในมุมด้าน ตะวันตกเฉียงเหนือของห้างสรรพสินค้า และคนทำขนมมีร้านเบเกอรี่อยู่ทางทิศตะวันออก คนทำขนม และช่างไม้มีอินพุตและเอาต์พุตที่แตกต่างกัน นั่นคือ ไม้สั่งและ ผลิตภัณฑ์ของพวกเขาจะไม่เหมือนกัน ยิ่งกว่านั้น เวลาในการทำงานของ แต่ละธุรกิจค่อนข้างแตกต่างกัน และผู้ใช้ของพวกเขาทราบและ เข้าใจเรื่องนี้ดี

คำศัพท์ในการจัดการระบบงาน

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

ชิ้นงานใดๆ ภายในธุรกิจ ถือว่าเป็น *งาน*. ตัวอย่างของชิ้นงานอาจจะเป็น จดหมายของลูกค้า, การโทรศัพท์, ไม้สั่งซื้อ หรือการทำ ความสะอาดตอนกลางคืน เช่นเดียวกับ System i ผลิตภัณฑ์ บนระบบ งานแต่ละงานมีชื่อที่ไม่ ซ้ำกัน

รายละเอียดของงาน อธิบายวิธีการจัดการงาน ที่เข้ามาในระบบย่อย รายละเอียดของงานมีชั้นของข้อมูล เช่น user IDs, คิวงาน และข้อมูลการเรด ข้อมูลในรายละเอียดของงานอาจ เปรียบเทียบได้กับรายละเอียดของงานในธุรกิจขนาดเล็ก

ธุรกิจ มีลักษณะอย่างไร? ทุกร้านค้ามีพิมพ์เขียวหรือแผนผังร้านค้า แผนผังเหล่านี้ใช้สำหรับการอธิบาย โดยมีรายละเอียดแตกต่างกันไปถึง องค์ประกอบทางกายภาพของธุรกิจ ธุรกิจอาจมีร้านค้าที่มีสิ่งดังต่อไปนี้: พื้น 2 ชั้น, ประตู 5 บาน, ตู้ไปรษณีย์ 3 ตู้ และโทรศัพท์ 2 เครื่อง บนระบบของคุณ *subsystem description* มีข้อมูลทั้งหมดเกี่ยวกับระบบย่อย

งานมาจากไหน? สำหรับช่างไม้ งานมาจากโทรศัพท์ของลูกค้า, จากการอ้างอิง และจากบุคคลที่แวะชมเฟอร์นิเจอร์ บนระบบของคุณ งานสามารถมาได้จากหลายที่ ตัวอย่างเช่น คิวงาน, เวิร์กสเตชัน, การสื่อสาร, งานแบบ autostart และงานแบบ prestart

งานหาพื้นที่ว่างได้อย่างไร? ภายใน ห้างสรรพสินค้า แต่ละธุรกิจ (ระบบย่อย) มีพื้นที่ร้านค้าจำนวนหนึ่ง บนระบบ *พูลหน่วยความจำ* ช่วยให้คุณสามารถควบคุมแหล่งเก็บข้อมูลหลัก (หรือพื้นที่ร้านค้า) ที่แต่ละระบบย่อย (ธุรกิจ) สามารถใช้ทำงานของตนได้ ร้านค้ายังมีพื้นที่ร้านค้า (ระบบย่อย) มาก ยิ่งสามารถรองรับลูกค้า หรืองานได้มากขึ้น

งานเข้ามาได้อย่างไร? ลูกค้าที่ไม่สามารถหาร้านค้าที่ต้องการได้ อาจพบว่าบูธข้อมูล สามารถช่วยให้ข้อมูลทิศทางและทำให้เขาไปยังร้านค้าที่ต้องการได้ เช่นเดียวกับบนระบบของคุณ *Routing entry* คล้ายกับไดเรกทอรีร้านค้าหรือบูธข้อมูล หลังจากที่ได้พบ *routing entry*, *entry* จะช่วยแนะนำให้งานทราบถึงตำแหน่งที่ถูกต้องของงาน อย่างไรก็ตาม ต้องพบ *routing entry* ก่อนสามารถพบได้โดยใช้ *ข้อมูล การเรด*. ข้อมูลการเรดคือสิ่งที่งานใช้ในการค้นหา *routing entry* ที่ถูกต้อง

งานถูกจัดการอย่างไร? ช่างไม้ต้องจัดระดับความสำคัญของแต่ละงาน แก้อัปเดตที่ต้องจัดส่งลูกค้าตอนปลายสัปดาห์ควรจะทำก่อนชั้นวางหนังสือที่มีกำหนดส่ง ตอนสิ้นเดือน บนระบบของคุณ *classes* ให้ข้อมูลเกี่ยวกับวิธีการจัดการงานในขณะที่อยู่ในระบบย่อย ข้อมูลนี้รวมถึง ระดับความสำคัญในขณะที่ยัง, หน่วยเก็บสูงสุด, เวลา CPU สูงสุด และการแบ่ง เวลา แต่ละแอ็ดทริบิวต์เหล่านี้ประกอบกันเป็นวิธีการและเวลาที่งานจะถูกประมวลผล

เช่นเดียวกับที่มีกฎที่ส่งผลต่อร้านค้าทั้งหมดในห้างบนระบบของคุณ ตัวอย่างของกฎ เหล่านี้คือ *ค่ากำหนดของระบบ*. ค่ากำหนดของระบบคือชั้นของข้อมูลที่ใช้กับระบบทั้งระบบ ค่าระบบรวมข้อมูล เช่น วันที่ และเวลา, ข้อมูลค่าติดตั้ง, ข้อมูล sign-on, ความปลอดภัยของระบบ และการจัดหน่วยความจำ

ลูกค้าแต่ละรายในห้างสรรพสินค้ามีข้อมูลเฉพาะของตน บนระบบของคุณ *โปรไฟล์ผู้ใช้* พักข้อมูลเฉพาะสำหรับผู้ใช้เฉพาะ ในลักษณะคล้ายกับบัตรเครดิตของลูกค้า โปรไฟล์ผู้ใช้ให้สิทธิเฉพาะของผู้ใช้ และกำหนดแอ็ดทริบิวต์ผู้ใช้สำหรับงานของผู้ใช้นั้น *แอ็ดทริบิวต์งาน* เหล่านี้นำเสนอข้อมูลที่รวมถึง แต่ไม่จำกัด เฉพาะรายละเอียดของงาน, เอาต์พุตคิวหรืออุปกรณ์การพิมพ์, message queue, ใค้ดบัญชีผู้ใช้, และระดับความสำคัญในการจัดตารางเวลา

ช่วงชีวิตของงาน

เพื่อให้เข้าใจพื้นฐานของ System i การจัดการระบบงาน ให้ลองติดตามงานแบ็ตซ์ธรรมดาเมื่องานดำเนินการผ่านระบบ

ช่วงชีวิตของงานแบ็ตซ์แบบธรรมดาจะเริ่มต้นขึ้นเมื่อคุณได้ส่งงานดังกล่าวเข้าสู่ระบบ จากนั้นงานจะถูกส่งเข้าไปในคิวงานที่ซึ่งงานจะไปรอเข้าสู่ระบบย่อยเพื่อการรันต่อไป หลังจากทำงานย้ายไปที่ระบบย่อย งานจะได้รับการจัดสรรหน่วยความจำเพื่อใช้รัน จากนั้น printer output file (เรียกอีกอย่างว่าสพูลไฟล์) จะถูกส่งไปที่ เอาต์พุตคิว เพื่อรอคำสั่งต่อไปว่าจะทำอะไร (เช่น การพิมพ์) แม้ว่างานทั้งหมดไม่ได้มีพาทที่เหมือนกับพาทนี้ทุกอย่าง แต่คุณสามารถเข้าใจวิธีการที่จะทำให้งานอื่นเสร็จสมบูรณ์บนระบบได้ดีขึ้น โดยการเรียนรู้เพิ่มเติมเกี่ยวกับวงจรชีวิตงานปกติ

ส่งงาน → งานเข้าสู่คิวงาน → งานเข้าสู่ระบบย่อย → พูล หน่วยความจำจัดสรรหน่วยความจำไปที่ระบบย่อย → งานเสร็จสิ้นและย้ายไปที่เอาต์พุตคิว

การส่งงาน

เมื่องานถูกส่งงานจะถูกสร้างขึ้นและเข้าสู่ระบบ ในเวลานี้ แอ็ททริบิวต์จะถูกกำหนดให้กับงาน

รายละเอียดของงานมี *แอ็ททริบิวต์* ที่งานจะใช้ เพื่อดำเนินการผ่านวงจรชีวิตการจัดการระบบงาน แอ็ททริบิวต์เหล่านี้รวมถึง โปรไฟล์ผู้ใช้ที่งานจะเริ่มรันอยู่ภายใต้, ข้อมูลคำขอ (ซึ่งบอกงานว่าจะทำอะไร) และส่วน *initial user* ของรายชื่อไลบรารีและอื่นๆ รายละเอียดของงานยังเก็บข้อมูลที่จะบอกให้งานรู้ว่าจะต้องเข้าไปอยู่ในคิวงานใด และข้อมูลการเรอต์ ข้อมูลการเรอต์จะถูกใช้งานโดยระบบย่อยในภายหลังเพื่อค้นหา *routing entry* ที่ประกอบไปด้วยข้อมูลที่งานจำเป็นต้องใช้ในการสตาร์ทเพื่อจะทำงาน เอ็ททริบิวต์จะถูกกำหนดเอาไว้ใน รายละเอียดของงานด้วย เอ็ททริบิวต์เป็นตัวบอกว่าพรินเตอร์เอ็ททริบิวต์ (หรือเรียกว่า *ไฟล์สพูล*) จากงานจะออกไปที่จุดใด

หลังจากที่งานได้รับค่า (การกำหนดค่าเริ่มต้น, การปรับแต่งค่า) สำหรับแอ็ททริบิวต์ งาน จะย้ายไปที่คิวงานที่ซึ่งงานจะรอเพื่อเข้าสู่ระบบย่อย

งานเข้าสู่คิวงาน

คิวงาน เป็น *entry point* ของการทำงานสำหรับงานแบ็คที่ที่จะเข้าสู่ระบบ. โดยอาจเปรียบเทียบกับเป็น "ห้องรับรอง" ของระบบย่อย.

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

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

งานเข้าสู่ระบบย่อย

ระบบย่อย คือสถานะแวดล้อมการปฏิบัติการที่ระบบ จัดการรีซอร์สที่งานใช้และควบคุมงานที่รันอยู่ภายใน ระบบย่อย หลังจาก ที่งานรันอยู่ในระบบย่อยแล้ว งานระบบย่อยจะดำเนินการตาม คำขอของผู้ใช้ เช่น การพัก, การรีลีส และการจบงาน เมื่องาน เข้าสู่ระบบย่อย งานจะกลายเป็นแอ็คทีฟ

เช่นเดียวกันกับงาน ระบบย่อยจะมี *description* ที่เก็บข้อมูลสำคัญเพื่อใช้ในการทำงานให้สำเร็จ ใน *subsystem description* คือ *routing entry* *routing entry* อ้างอิงคลาสอีอบเจ็กต์ ซึ่งมีแอ็ททริบิวต์ ที่ควบคุมสถานะแวดล้อมรันไทม์ อย่างไรก็ตาม ก่อนที่งานจะได้รับ *routing entry*, ข้อมูลการเรอต์จะต้องจับคู่กับค่าเปรียบเทียบที่อยู่ใน *routing entry* เสียก่อน หากไม่เกิดการจับคู่ดังกล่าวแล้ว งานจะไมรันได้

หลังจากสร้างความสัมพันธ์ระหว่างข้อมูลการเรอต์ และ *routing entry* แล้ว จะมีการกำหนดคลาสอีอบเจ็กต์ที่งานจะใช้ แอ็ททริบิวต์บางอย่าง ที่ควบคุมสถานะแวดล้อมรันไทม์รวมถึง ระดับความสำคัญการรัน, การแบ่งเวลา, ช่วงเวลารอสูงสุด, เวลาประมวลผลสูงสุด, หน่วยเก็บชั่วคราวสูงสุด และจำนวนสูงสุดของ *thread*

subsystem description จะกำหนดพูลหน่วยความจำที่จะถูกจัดสรรให้กับระบบย่อย *subsystem description* จะประกอบด้วยจำนวนงานแอ็คทีฟสูงสุดซึ่งก็คือจำนวนสูงสุดที่สามารถมีได้ของงานแอ็คทีฟในช่วงเวลาหนึ่งในระบบย่อย

งานไม่สามารถรันได้ จนกว่างานจะได้ activity level และถูกกำหนดพูลหน่วย ความจำ subsystem description ซึ่งเหมือนกับ job description จะเก็บข้อมูล เช่น พูลหน่วยความจำที่จะใช้, routing entry, จำนวนงานแอนด์ที่สูงสุด และจำนวนงานแอนด์ที่สูงสุด ที่มีอยู่ในระบบย่อยในปัจจุบัน

ระบบย่อยใช้หน่วยความจำจากพูลหน่วยความจำเพื่อรันงาน

หน่วยความจำ คือรีซอร์สจากพูลหน่วยความจำที่ ระบบย่อยจะใช้ในการรันงาน. จำนวนของหน่วยความจำในพูลหน่วยความจำ, และ จำนวนงานอื่นที่แย่งกันใช้หน่วยความจำ, มีผลกระทบต่อประสิทธิภาพในการ รันงาน.

พูลหน่วยความจำ จัดสรรหน่วยความจำซึ่งใช้ในการรันให้งาน. มีปัจจัย หลายอย่างส่งผลต่อวิธีการรันงานในพูลหน่วยความจำ, เช่น ขนาดของพูล หน่วยความจำ, activity level ในพูลหน่วยความจำ, และการเพจและ faulting. activity level ในพูลหน่วยความจำมีความสัมพันธ์โดยตรงกับจำนวนของ threads ที่ได้รับอนุญาตให้รันในพูลหน่วยความจำในเวลาหนึ่ง. จำไว้ว่า, งานทุกงานจะต้องมี thread แอนด์ที่พื่อน้อยหนึ่งอัน, แต่บางงานสามารถมี thread ได้หลายอัน. Thread จะช่วยให้งานสามารถทำงานหลายๆ อย่างได้ ในเวลาเดียวกัน. ตัวอย่างเช่น, หนึ่ง thread สามารถออกไปและทำการคำนวณ ในขณะที่ thread อื่นจะรอข้อมูลเพิ่มเติม เพื่อประมวลผล.

การเพจ เป็นการเคลื่อนที่เข้าออกหน่วยความจำของข้อมูล, ซึ่งเป็นได้ ทั้งแบบซิงโครนัสและอะซิงโครนัส. สามารถเขียนเพจไปยังหน่วยเก็บหรือย้ายออกจากหน่วยความจำโดยไม่ต้องถูกเขียนหากไม่มีการปรับเปลี่ยนใดๆ. Faulting จะทำให้เกิดการเพจบนเซิร์ฟเวอร์. Faulting จะเกิดขึ้นเมื่อน้ำที่อ้างอิง, หรือชิ้นของข้อมูล, ไม่ได้อยู่ในหน่วยความจำ. ข้อผิดพลาดนี้อาจทำให้โปรแกรมหยุดทำงานเพราะโปรแกรมต้องรอการเพจข้อมูล.

ระบบย่อย จะใช้พูลหน่วยความจำต่างๆ เพื่อสนับสนุนประเภทต่างๆ ของงานที่จะรันภายในระบบย่อยนั้น.

งานเสร็จสิ้นและย้ายไปที่เอาต์พุตคิว

พรินเตอร์เอาต์พุตของงาน (เรียกอีกอย่างว่าสพูลไฟล์) ถูกส่งไปที่ เอาต์พุตคิวที่ซึ่งเอาต์พุตจะรอเพื่อส่งต่อไปที่พรินเตอร์หรือไฟล์ เอาต์พุตคิวจะเหมือนกับคิวงานตรงที่เอาต์พุตคิวจะ ควบคุมการส่งเอาต์พุตไปให้กับอุปกรณ์การพิมพ์ เอาต์พุตคิวอนุญาตให้ผู้ใช้ควบคุมว่าไฟล์ใดที่จะถูกพิมพ์ก่อนหลัง

เอาต์พุตคิว คือพื้นที่ที่ printer output file รอที่จะถูก ประมวลผลและส่งไปที่พรินเตอร์ พรินเตอร์เอาต์พุตจะถูกสร้างขึ้นทั้งโดยระบบหรือโดยผู้ใช้โดยใช้ไฟล์พิมพ์ อย่างไรก็ตามหนึ่งไฟล์พิมพ์คล้ายกับเท็มเพลต หรือต้นแบบที่ซึ่งมีการตั้งค่าตีพอลต์สำหรับแอตทริบิวต์ของพรินเตอร์เอาต์พุต ไว้แล้ว ไฟล์พิมพ์เป็นจุดเริ่มต้นของวงรอบช่วงชีวิตของพรินเตอร์เอาต์พุต

ไฟล์พิมพ์มีแอตทริบิวต์เอาต์พุตคิว (OUTQ) และอุปกรณ์การพิมพ์ (DEV), ซึ่งบ่งชี้ว่าพรินเตอร์เอาต์พุตจะถูกกำหนดทิศทางอย่างไรโดยปกติ การตั้งค่า ตีพอลต์คือ *JOB, ซึ่งหมายความว่าแอตทริบิวต์งานของเอาต์พุตคิว และอุปกรณ์การพิมพ์เป็นตัวกำหนดทิศทางของพรินเตอร์เอาต์พุต แอตทริบิวต์งานของเอาต์พุตคิวและการตั้งค่าอุปกรณ์การพิมพ์คือข้อมูลที่ได้นำเมื่องานถูกสร้างขึ้น ข้อมูลนี้สร้างขึ้นจากข้อมูลที่ได้นำจาก โปรไฟล์ผู้ใช้ที่งานกำลังรันอยู่ภายใต้, รายละเอียดของงาน, workstation device description และค่ากำหนดของระบบ Default printer (QPRTDEV)

เมื่อพร้อมที่จะสร้างพรินเตอร์เอาต์พุต ระบบจะตรวจสอบไฟล์พิมพ์และแอตทริบิวต์งาน (ในลำดับนี้) เพื่อดูว่าเอาต์พุตคิวใดจะประมวลผลพรินเตอร์เอาต์พุตและอุปกรณ์การพิมพ์ใดที่ระบบจะใช้ งาน หากไม่พบ เอาต์พุตคิวที่ระบุ พรินเตอร์เอาต์พุตจะถูกกำหนดทิศทาง ไปที่ QGPL/QPRINT

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

วิธีการทำงานให้เสร็จสมบูรณ์

หัวข้อนี้อธิบายว่างานคืออะไร, ต้องตั้งค่าอะไรก่อนจึงสามารถเริ่มงานได้, งานเดินทางผ่านระบบอย่างไร และเกิดอะไรขึ้นกับงานหลังจากที่รันแล้ว

งานคืออะไร

บนผลิตภัณฑ์ System i ของคุณ งานจะถูกทำอยู่เสมอ ไม่ว่าคุณจะเริ่มงาน หรือระบบจะเริ่มงานก็ตาม action ใดๆบนระบบมีงานบางชนิดที่กำลังปฏิบัติเพื่อให้ action สมบูรณ์.

งานจะถูกทำเมื่อคุณเปิดระบบ ขึ้น เมื่อคุณเปิดไฟล์ หรือเมื่อคุณเคอร์รี่ฐานข้อมูล ขึ้นงาน (work) แต่ละชั้นบนระบบจะถูกกระทำผ่านทางงาน (job) งานอาจเป็นสิ่งที่ง่าย ๆ เช่น แอปพลิเคชันที่รอให้ผู้ใช้เรียก หรืออาจเป็นสิ่งที่ซับซ้อนเช่นเคอร์รี่ระบบที่รันอย่างต่อเนื่องซึ่งจะมอนิเตอร์ จำนวนของผู้ใช้บนระบบทุกชั่วโมง. งานบางงาน โดยเฉพาะงานแบ็คซ์ และงานแบบโต้ตอบ มีรายละเอียดของงานที่เชื่อมโยงซึ่งบอกว่า งานจะรันเมื่อไรและรันที่ใด

งานประกอบด้วยโปรแกรมต่างๆ ที่จะทำฟังก์ชันบางอย่างเฉพาะ ไม่มีขีดจำกัดถึงจำนวน ฟังก์ชันที่งานจะสามารถใช้ปฏิบัติงานได้งานประกอบด้วยคำสั่งตามขั้นตอนที่จะต้อง กระทำเพื่อให้งานเสร็จสมบูรณ์ หรือก็คือโปรแกรมที่ทำให้งานรันไปตามลำดับอย่างเจาะจง (ตัวอย่างเช่น โปรแกรม A จะต้องรันก่อน โปรแกรม B จึงจะสามารถเริ่มได้) Thread ช่วยให้งานสามารถทำงานของตน ได้เสร็จสมบูรณ์ งานแ็คทีฟ จะประกอบด้วย thread อย่างน้อยหนึ่งตัว เมื่องานมี thread อยู่หลายตัว ก็จะทำให้งานสามารถทำสิ่งต่างๆ ได้มากขึ้นในเวลาเดียวกัน ตัวอย่างเช่น thread อันหนึ่งสามารถออกไปทำการคำนวณ ในขณะที่อีกอันหนึ่งจะรอข้อมูลเข้ามามากขึ้นเพื่อทำการประมวลผลต่อไป

สิ่งที่เกิดขึ้นก่อนงานเข้าสู่ระบบ

งานทั้งหมด ยกเว้นงานระบบ จะรันภายในระบบย่อย เพื่อให้งานเริ่มในระบบย่อยที่แ็คทีฟ จำเป็นต้องสร้างพูลหน่วยความจำ และซอร์สของ work entry point อย่างน้อยหนึ่งซอร์ส คิวงานคือตัวอย่างของ ซอร์สของงาน

ตัว System i ผลิตภัณฑ์ประมวลชุดดีฟอลต์ของคิวงาน, ระบบย่อย และพูลหน่วยความจำ ซึ่งช่วยให้งานสามารถเริ่มต้นได้ในทันทีที่เปิดเครื่อง

คุณสามารถปรับแต่งระบบย่อย และ configuration ของพูลหน่วยความจำ เพื่อให้ความสามารถ และประสิทธิภาพของ System i ผลิตภัณฑ์สูงสุด ตัวอย่างเช่น หากงานแบ็คซ์ มีความสำคัญต่อความสำเร็จของธุรกิจ คุณอาจต้องการจัดสรรหน่วยความจำให้กับงานแบ็คซ์มากขึ้น หรือคุณอาจกำหนดจำนวนงานที่จะรันในเวลาหนึ่งในระบบย่อย Qbatch ของคุณ ให้น้อยลง เพื่อให้งานสามารถใช้ปริมาณรีซอร์สในการรันได้มากขึ้น นอกจากนี้ คุณสามารถสร้างคิวงาน, ระบบย่อย และพูลหน่วยความจำ โดยออกแบบให้ทำงานชนิดใดชนิดหนึ่งเป็นการเฉพาะ ตัวอย่างเช่น คุณสามารถสร้างคิวงานชื่อ Nightreps เพื่อให้ รายงานแบบแบ็คซ์เป็นรายคืนถูกส่งไปยังระบบย่อยที่ชื่อ Nightrep ซึ่งจะจัดสรร หน่วยความจำไว้เป็นการเฉพาะเพื่อใช้รันงานแบ็คซ์เหล่านี้

งานเข้าสู่ระบบได้อย่างไร

work entry จะบอกถึงต้นทางที่งานได้เข้ามาสู่ระบบย่อยเพื่อรอให้เรียกทำงาน งานแต่ละชนิดมีชนิดของ work entry ที่จะใช้แตกต่างกัน

ตัวอย่างเช่น งานแบ็คซ์ส่วนใหญ่จะใช้คิวงานเพื่อเข้าสู่ระบบย่อย Job queue entry เป็นกลไกหนึ่งที่คิวงานจะถูกกำหนดให้เป็นต้นทางของงาน ที่จะส่งเข้าไปสู่ระบบย่อย

Work entry ถูกเก็บอยู่ใน subsystem description ถ้า subsystem description ไม่มี work entry สำหรับชนิดของงานที่กำลังจะถูกรัน งานดังกล่าวจะไม่สามารถรันในระบบย่อยนั้นได้ ระบบย่อยที่จัดส่งมาจาก IBM มี work entry ดีพอลต์อยู่ใน subsystem description แล้ว แต่ต้องไม่ลืมว่า work entry ดีพอลต์บางตัวที่จัดส่งมาพร้อมกับระบบย่อยนั้นได้ถูกจัดสรรให้ทำการรันงานแบบเจาะจงเท่านั้น

งานถูกประมวลผลอย่างไร

เมื่อระบบถูกเริ่มต้น งานการมอนิเตอร์ระบบย่อยจะเริ่มการรัน งานการมอนิเตอร์ระบบย่อยควบคุมงานภายในระบบย่อย นอกจากนั้น ยังเป็นตัวช่วยสตาร์ทและจบการทำงานให้ พร้อมกับจัดการรีซอร์สให้กับการทำงานนั้นในระบบย่อยอีกด้วย

งาน (หรือหลายงาน) เข้าสู่ระบบย่อยผ่านทาง work entry ที่ซึ่งงาน จะกลายเป็นแอสคทีฟและพร้อมที่จะรัน งานสามารถทำให้เสร็จสมบูรณ์ได้เฉพาะเมื่อ ระบบย่อยได้จัดสรรหน่วยความจำให้รันเท่านั้น หน่วยความจำถูกจัดสรรให้กับระบบย่อยโดย พูลหน่วยความจำ

subsystem description ช่วยประมวลผลงานได้อย่างไร

เช่นเดียวกับงาน ระบบย่อยมี description ที่เรียกว่า subsystem description subsystem description มีข้อมูลสำคัญที่บอกวิธี, ตำแหน่ง และจำนวนงานที่สามารถแอสคทีฟในระบบย่อยในเวลาเดียวกัน และรีซอร์สที่สามารถเรียกใช้เพื่อทำงานได้

Routing entry

routing entry มีอยู่ใน subsystem description และบอกให้ระบบย่อยทราบ ถึงโปรแกรมที่จะรันสำหรับงาน, พูลหน่วยความจำที่จะรันงานในนั้น และคลาสอ็อบเจกต์ที่จะใช้ในการรันงาน

คลาสอ็อบเจกต์

คลาสอ็อบเจกต์กำหนดระดับความสำคัญการรัน, เวลาอดีพอลต์, การแบ่งเวลา และแอตทริบิวต์อื่น ระดับความสำคัญการรันมีความสำคัญ เพราะเป็นตัวกำหนดว่าเมื่อไรที่งานจะได้เวลาของตัวประมวลผลเพื่อรันงาน ระดับความสำคัญในการรันอยู่ระหว่าง 0 ถึง 99 โดยที่ 0 มีความสำคัญสูงสุด (เฉพาะงานระบบเท่านั้นที่จะได้รับระดับความสำคัญ 0 เนื่องจากเป็นงานที่รันระบบ)

เมื่องานเข้าสู่ระบบย่อย ระบบย่อยจะพยายามจับคู่ข้อมูลการเรดท์กับค่าเปรียบเทียบใน routing entry หากข้อมูลการเรดท์และค่าเปรียบเทียบใน routing entry จับคู่กันได้ routing entry จะถูกกำหนดไว้ให้กับงานนั้น แต่ถ้าจับคู่กันไม่ได้ใน routing entry ใดๆ งานจะสิ้นสุดลง

ปัจจัยอื่นที่มีผลกระทบในกรณีที่งานรันอยู่ในระบบย่อย คือ จำนวนงานที่ได้รับ อนุญาตให้มีสถานะแอสคทีฟในระบบย่อยได้พร้อมกัน (หรือรู้จักในอีกชื่อว่างานแอสคทีฟสูงสุดในระบบย่อย) เมื่องานที่แอสคทีฟในระบบย่อยมีมากถึงจำนวนสูงสุดแล้ว งานอื่นๆ จะไม่สามารถเข้ามาสู่ระบบย่อยได้จนกว่างานแอสคทีฟจะรันเสร็จสิ้น หน่วยความจำจะต้องถูกจัดสรรให้กับระบบย่อย เพื่อให้งานรัน Memory pool activity level บอกระบบถึงจำนวน thread ที่สามารถแอสคทีฟได้ภายในพูลหน่วยความจำ จำไว้ว่างานแอสคทีฟจะประกอบด้วย thread อย่างน้อยหนึ่งอัน เมื่อ activity level ของพูลหน่วยความจำถึงระดับสูงสุด งานจะต้องรอให้ thread อื่นใช้งาน activity level ให้เสร็จสิ้นเสียก่อน ดังนั้น งานจึงสามารถแอสคทีฟในระบบย่อยและไม่ได้รัน

หมายเหตุ: อย่าสับสนระหว่างงานแอสคทีฟสูงสุดในระบบย่อย และ memory pool activity level

งานออกจากระบบอย่างไร

เอาต์พุตคิวทำงานคล้ายกับคิวงานตรงที่มันจะจัดลำดับเอาต์พุตที่จะพิมพ์ให้ ทั้งพริ้นเตอร์เอาต์พุตและเอาต์พุตควมมีแอตทริบิวต์ ที่จะใช้ในการพิมพ์ข้อมูล.

พริเตอร์เอาต์พุตมีข้อมูลเอาต์พุตที่รอประมวลผล เช่น ข้อมูลที่รอพิมพ์ พริเตอร์เอาต์พุตยังมีข้อมูลที่สำคัญที่ใช้เพื่อจัดลำดับการพิมพ์ด้วย แอ็ททริบิวต์พริเตอร์เอาต์พุตมีเอาต์พุตคิวที่มีพริเตอร์เอาต์พุตอยู่, ระดับความสำคัญ, สถานะ และตารางเวลาของพริเตอร์เอาต์พุต

เอาต์พุตคิวมีแอ็ททริบิวต์ของตัวเองที่กำหนดลำดับการประมวลผลไฟล์พริเตอร์เอาต์พุต นอกจากนั้นเอาต์พุตคิวยังมีสิทธิในการใช้งาน ที่จำเป็นสำหรับทำการเปลี่ยนแปลงกับพริเตอร์เอาต์พุตและเอาต์พุตคิว

เมื่อพริเตอร์เอาต์พุตพร้อมที่จะถูกส่งไปยังพริเตอร์งานแบบ writer จะดึงพริเตอร์เอาต์พุตขึ้นมา งานแบบ writer จะนำข้อมูลจากพริเตอร์เอาต์พุตและเตรียมข้อมูลเอาต์พุตนั้นเพื่อพิมพ์

หลักการ

ไม่ว่าคุณจะใช้เคยใช้การจัดการระบบงาน หรือใช้เครื่องมือการจัดการระบบงานมาเป็นเวลาหลายปีแล้ว คุณจะพบว่าแนวคิดการจัดการระบบงานเหล่านี้อาจเป็นประโยชน์สำหรับคุณ

โครงสร้างระบบของคุณ

หลังจากรับผลิตภัณฑ์ System i ของคุณ คุณอาจต้องการทราบว่ามียุทธวิธีในระบบ ไม่ว่าคุณต้องการเปลี่ยนโปรแกรมเริ่มต้นใดหรือไม่ และอินเตอร์เฟซผู้ใช้ชนิดใดที่คุณจะทำงานด้วย

ระบบย่อยที่ส่งมาพร้อมกับระบบ

subsystem configuration สมบูรณ์ที่ได้มาจาก IBM มีอยู่สองชุด และสามารถใช้ได้โดยไม่ต้องเปลี่ยนแปลงใดๆ

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

ระบบย่อย	คำอธิบาย
Qbase (ระบบย่อยการควบคุม)	Qbase จะสนับสนุนงานโต้ตอบ, แบตช์ และสื่อสาร และ Qbase จะมีงาน autostart, ซึ่งจะสตาร์ทระบบย่อย Qusrwrk, Qserver และ Qspl โดยอัตโนมัติ
Qserver	เป็นระบบย่อยของไฟล์เซิร์ฟเวอร์
Qspl	นี่คือระบบย่อยสพูลที่สนับสนุนงานเครื่องอ่านและงาน writer
Qsyswrk	นี่คือระบบย่อยของงานระบบซึ่งประกอบด้วยงานที่สนับสนุนฟังก์ชันระบบทั้งหลายที่จะสตาร์ทขึ้นโดยอัตโนมัติในการเริ่มเปิดระบบและเมื่อระบบได้ออกจากสภาพถูกจำกัดการทำงาน
Qusrwrk	นี่คือระบบย่อยของงานผู้ใช้ซึ่งประกอบด้วยงานที่ถูกเรียกทำงานโดยเซิร์ฟเวอร์เพื่อทำงานในนามของผู้ใช้

Configuration อื่น ซึ่งได้มาจาก IBM ประกอบด้วย subsystem description ต่อไปนี้:

ระบบย่อย	คำอธิบาย
Qctl (controlling subsystem)	Qctl มีงานแบบ autostart, ที่จะสตาร์ทระบบย่อย Qinter, Qbatch, Qcmn, Qusrwrk, Qserver และ Qspl ให้โดยอัตโนมัติ

Qinter	นี่คือระบบย่อยที่สนับสนุนงานแบบโต้ตอบ ยกเว้นในส่วนของคนโซล
Qbatch	นี่คือระบบย่อยที่สนับสนุนงานแบตช์
Qcmm	นี่คือระบบย่อยที่สนับสนุนงานสื่อสาร, เว้นแต่งานสื่อสารในโปรโตคอล TCP/IP งานสื่อสารเหล่านี้เป็นสิ่งที่จำเป็นสำหรับโปรโตคอลสื่อสารต่างๆ ที่ระบบ i5/OS สนับสนุน
Qserver	เป็นระบบย่อยของไฟล์เซิร์ฟเวอร์
Qspl	นี่คือระบบย่อยสพูลที่สนับสนุนงานเครื่องอ่านและงาน writer
Qsyswrk	นี่คือระบบย่อยของงานระบบ ซึ่งประกอบด้วยงานที่สนับสนุนฟังก์ชันระบบทั้งหลายที่จะสแตร์ทขึ้นโดยอัตโนมัติในการเริ่มเปิดระบบและเมื่อระบบได้ออกจากสภาพถูกจำกัดการทำงาน
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 นี่คือการกำหนดของระบบซึ่งระบุชื่อของโปรแกรมที่จะถูกเรียกจากงานแบบ autostart เมื่อระบบย่อยการควบคุม เริ่มต้น โปรแกรมนี้ปฏิบัติฟังก์ชันการตั้งค่า เช่น การเริ่ม ระบบย่อยและพริ้นเตอร์ ค่ากำหนดของระบบนี้สามารถเปลี่ยนได้โดยเจ้าหน้าที่ดูแลความปลอดภัยหรือโดยบุคคลที่มีสิทธิเจ้าหน้าที่ดูแลความปลอดภัยเท่านั้น การเปลี่ยนแปลงค่ากำหนดของ ระบบนี้จะมีผลบังคับใช้ในการทำ IPL ครั้งถัดไป

QSTRUPPGM สามารถมีค่าเหล่านี้:

- QSTRUP QSYS: โปรแกรมที่ระบุถูกรัน อันเป็นผลมาจากการถ่ายโอนการควบคุมไปที่โปรแกรมจากงานแบบ autostart ในระบบย่อยการควบคุม
 - *NONE: งานแบบ autostart สิ้นสุดตามปกติโดยไม่มีการเรียกโปรแกรม
- ข้อมูลที่เกี่ยวข้อง**
ค่ากำหนดของระบบที่ควบคุม IPL

สิ่งที่เกิดขึ้นในระหว่าง IPL: โปรแกรมเริ่มต้นดีฟอลต์ QSYS/QSTRUP จะทำดังต่อไปนี้:

- เริ่มระบบย่อย QSPL สำหรับงานที่สพูล
- รีเส็ตคิวงาน QS36MRT และ QS36EVOKE ถ้าคิวงานถูกพักไว้ (คิวงานเหล่านี้ ถูกใช้โดยสถานะแวดล้อม 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) โปรแกรมระบบจะโหลดจากอุปกรณ์ซอร์สการโหลดที่กำหนดในหน่วยความจำรองของระบบ ฮาร์ดแวร์ ระบบจะถูกตรวจสอบด้วย i5/OS คอนโทรลพาเนลแสดงชุดโค้ดอ้างอิงของระบบที่บ่งชี้สถานะปัจจุบัน และเตือนคุณถึงปัญหาใดๆ เมื่อ IPL เสร็จสิ้น อินเทอร์เฟซแบบอักษรจะแสดงบนจอแสดงผล sign-on และผู้ใช้สามารถ sign on ด้วย System i Navigator.

มีหลายอ็อปชันสำหรับการเริ่มระบบของคุณ คุณสามารถ:

- เริ่มระบบโดยไม่เปลี่ยน configuration สิ่งนี้เรียกว่า *unattended IPL*.
- เปลี่ยน configuration ระบบของคุณในระหว่าง IPL สิ่งนี้เรียกว่า *attended IPL*.
จอภาพเพิ่มเติมของ attended IPL ขึ้นอยู่กับอ็อปชันที่คุณเลือกบนจอแสดงผลอ็อปชัน IPL จอเหล่านี้รวมถึงจอแสดงผลที่อนุญาตให้คุณเปลี่ยนค่ากำหนดของระบบและแอ็ดทริบิวต์ระบบอื่นในระหว่าง IPL, reconstruct แอ็คเซสพาธ, ตรวจสอบสถานะของข้อจำกัดฟิสิคัลไฟล์, ตั้งค่าและตั้งชื่ออุปกรณ์ใหม่ และระบุ อ็อปชันสำหรับสภาวะแวดล้อมการปฏิบัติการ
- เปลี่ยนชนิดของ IPL จากคอนโทรลพาเนลระบบของคุณ
- จัดตารางเวลาให้ระบบปิดและรีสตาร์ท

ปัญหาทั่วไปในระหว่าง IPL ถูกเรียกว่าเป็น *IPL ผิดปกติ*

สำหรับข้อมูลเพิ่มเติมเกี่ยวกับ IPL และการปิดระบบ ให้ดูข้อมูลเกี่ยวกับการเริ่ม และหยุดระบบ

ข้อมูลที่เกี่ยวข้อง

การเริ่ม และหยุดระบบ

การปิดเครื่อง

คุณต้องระวังเมื่อปิดระบบของคุณ หากคุณปิดระบบโดยที่ยังทำภารกิจบางอย่างไม่สมบูรณ์ คุณอาจทำให้ข้อมูลเสียหาย หรือทำให้ระบบทำงานในลักษณะที่คาดเดาไม่ได้

หัวข้อศูนย์กลางข้อมูลต่อไปนี้มีข้อมูลเพิ่มเติมเกี่ยวกับการปิดเครื่อง อย่างปลอดภัย

- วิธีการปิดระบบอย่างปลอดภัย เมื่อมี Windows® การแสดงเซิร์ฟเวอร์แบบรวมอยู่
- ปิดเครื่องด้วยโลจิคัลพาร์ติชัน
- ปิดโปรแกรมออกจากระบบ
- โปรแกรมออกสำหรับปรับการปิดเครื่อง

ข้อมูลที่เกี่ยวข้อง

ปิดฮาร์ดแวร์ System i ของคุณเมื่อ integrated Windows servers ปรากฏ

ปิดเครื่องด้วยโลจิคัลพาร์ติชัน

ปิด API โปรแกรมออกจากระบบ

โปรแกรมออกสำหรับปรับการปิด API

System i Navigator

System i Navigator เป็นอินเทอร์เฟซแบบกราฟิกที่เปี่ยมประสิทธิภาพสำหรับโคลเอ็นต์ Windows ด้วย System i Navigator คุณสามารถจัดการ และดูระบบของคุณจากเดสก์ทอป Windows ของคุณ คุณสามารถใช้ System i Navigator เพื่อปฏิบัติภารกิจส่วนใหญ่ที่เชื่อมโยงกับการจัดการระบบงาน

อินเทอร์เฟซนี้ได้รับการออกแบบมาเพื่อให้คุณสามารถทำงานได้ดีขึ้น ดังนั้น ขอแนะนำให้คุณใช้ System i Navigator ซึ่งมีคำอธิบายออนไลน์ช่วยแนะนำคุณ ในขณะที่อินเทอร์เฟซนี้ยังคงอยู่ระหว่างการพัฒนา คุณอาจยังคงต้องใช้โมเด็มเตอร์แบบดั้งเดิม เช่น PC5250 เพื่อทำภารกิจบางอย่างของคุณ ถ้าหัวข้ออธิบายถึงภารกิจเช่นนั้น คุณจะได้รับความแนะนำให้ใช้อินเทอร์เฟซแบบอักษร ภายในขั้นตอนที่แนะนำของหัวข้อ

ข้อมูลที่เกี่ยวข้อง

การเรียนรู้เกี่ยวกับ System i Navigator

System i Navigator สำหรับ Wireless

ภารกิจ System i Navigator บนเว็บ

ระบบย่อย

ระบบย่อย คือ ที่ทำงานถูกประมวลผลบนเซิร์ฟเวอร์ ระบบย่อยเป็นสภาวะแวดล้อมการดำเนินการหนึ่งที่ได้กำหนดไว้ก่อนเพื่อให้ระบบประสานเวิร์กโฟลว์และการใช้รีซอร์สให้เหมาะสมกัน ระบบอาจประกอบด้วยระบบย่อยหลายระบบ โดยที่ระบบทั้งหมดดำเนินการ เป็นอิสระจากกัน ระบบย่อยจะจัดการรีซอร์สเอง

งานทั้งหมด ยกเว้นงานระบบ จะรันภายใต้ระบบย่อย และระบบย่อยแต่ละระบบจะรันการปฏิบัติการที่ไม่เหมือนกัน ตัวอย่างเช่น ระบบย่อยหนึ่งระบบ อาจถูกตั้งค่าให้จัดการกับงานแบบโต้ตอบเท่านั้น ขณะที่ระบบย่อยอื่นจะจัดการกับงานแบตช์เท่านั้น สามารถออกแบบระบบย่อย ให้จัดการงานได้หลายชนิด ระบบจะอนุญาตให้กำหนดจำนวนระบบย่อย และชนิดของงานที่ระบบย่อยเหล่านั้นจะจัดการได้อีกด้วย

ลักษณะรันไทม์ของระบบย่อยถูกกำหนดในอ็อบเจกต์ที่เรียกว่า subsystem description ตัวอย่างเช่น ถ้าคุณต้องการเปลี่ยนปริมาณของงาน (จำนวนของงาน) ที่มาจากคิวงานอย่างถาวรเข้าในระบบย่อย คุณ เพียงแต่เปลี่ยน job queue entry ใน subsystem description เท่านั้น

งานที่เกี่ยวข้อง

“ภารกิจระบบย่อยทั่วไป” ในหน้า 178

ข้อมูลนี้อธิบายภารกิจทั่วไปที่คุณสามารถ ปฏิบัติบนระบบย่อย.

“การสร้างรายละเอียดของระบบย่อย” ในหน้า 181

คุณสามารถสร้าง subsystem description ได้สองวิธี คุณสามารถก๊อปปี้ subsystem description ที่มีอยู่และเปลี่ยนแปลง หรือคุณสามารถสร้าง description ใหม่ทั้งหมด

ข้อมูลที่เกี่ยวข้อง

รายงานจากประสบการณ์: การคอนฟิกูเรชันระบบย่อย

ระบบย่อยการควบคุม

ระบบย่อยการควบคุมคือ ระบบย่อยแบบโต้ตอบที่เริ่มทำงาน โดยอัตโนมัติเมื่อระบบเริ่มขึ้น และเป็นระบบย่อยซึ่งผู้ควบคุมระบบใช้ในการควบคุมระบบผ่านทางคอนโซลของระบบ ระบบย่อยการควบคุมถูกระบุ ในค่ากำหนดของระบบ Controlling subsystem/library (QCTLSBSD).

IBM จัดเตรียม controlling subsystem description ที่สมบูรณ์สองรายการ: QBASE (ระบบย่อยการควบคุมดีพอลต์) และ QCTL ในแต่ละครั้ง สามารถมีระบบย่อยการควบคุมที่แอคทีฟ บนระบบได้เพียงหนึ่งระบบย่อยเท่านั้น

เมื่อระบบอยู่ในสภาพที่ถูกจำกัด activity ส่วนใหญ่บนระบบจะจบลง และมีเวิร์กสเตชันที่แอคทีฟเพียงเวิร์กสเตชันเดียวเท่านั้น ระบบต้องอยู่ในสภาพนี้สำหรับการรันคำสั่ง เช่น Save System (SAVSYS) หรือ Reclaim Storage (RCLSTG) โปรแกรมการวินิจฉัยปัญหา เกี่ยวกับอุปกรณ์บางโปรแกรมอาจต้องการให้ระบบอยู่ในสภาพที่ถูกจำกัดด้วย เพื่อสิ้นสุดสภาพนี้ คุณต้องเริ่มระบบย่อยการควบคุมอีกครั้งหนึ่ง

หมายเหตุ: นอกจากนี้ยังมีข้อจำกัดเกี่ยวกับงานแบ็คชิ่งซึ่งสามารถมีงานแบ็คชิ่งที่แอคทีฟ ได้เพียงหนึ่งงานเท่านั้น

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

สำคัญ: ระบบไม่สามารถเข้าถึงสภาพที่ถูกจำกัด จนกว่าจะมีงานที่เหลืออยู่ในระบบย่อยการควบคุมเพียงงานเดียวเท่านั้น แต่ในบางครั้งอาจเป็นไปได้ที่แม้ว่าจะมีงานเหลืออยู่เพียงงานเดียว แต่ระบบ ไม่ได้เข้าไปในสภาพที่ถูกจำกัด ในกรณีนี้ คุณต้องตรวจสอบว่า ไม่มีงานค่าขอระบบที่ถูกหยุดชั่วคราว, งานกลุ่มที่หยุดชั่วคราว หรือ งานที่ตัดการเชื่อมต่ออยู่ บนจอแสดงผลแอคทีฟที่เหลืออยู่ ใช้คำสั่ง Work with Active Jobs (WRKACTJOB) และกด F14=รวมเพื่อแสดงงานที่ถูกหยุดชั่วคราว หรืองานที่ตัดการเชื่อมต่อ ถ้ามีงานเหล่านี้อยู่ คุณต้องจบงาน เพื่อให้ระบบเข้าถึงสภาพที่ถูกจำกัด ฟังก์ชัน ENDSYS และ ENDSBS จะส่งข้อความ CPI091C ไปให้ผู้ออกคำสั่ง เมื่อมีการตรวจพบสภาพนี้

งานที่เกี่ยวข้อง

“การสร้างระบบย่อยการควบคุม” ในหน้า 198

IBM จัดเตรียม controlling subsystem configuration ที่สมบูรณ์สองรายการ: QBASE (ระบบย่อยการควบคุมดีพอลต์) และ QCTL ในแต่ละครั้ง สามารถมีระบบย่อยการควบคุมที่แอคทีฟ บนระบบได้เพียงหนึ่งระบบย่อยเท่านั้น โดยปกติคอนฟิกูเรชั่นระบบย่อยที่จัดให้โดย IBM ควรเพียงพอสำหรับความต้องการทางธุรกิจส่วนใหญ่ อย่างไรก็ตาม คุณสามารถสร้างระบบย่อยการควบคุม เวอร์ชันของคุณเอง และตั้งค่าให้ตรงกับความต้องการเฉพาะของบริษัทของคุณ ได้มากขึ้น

“การวางระบบในสภาพที่ถูกจำกัด” ในหน้า 199

ถ้าระบบย่อยทั้งหมด รวมถึงระบบย่อยการควบคุม จบลง ระบบจะอยู่ในสภาพถูกจำกัด คุณสามารถวางระบบในสภาพที่ถูกจำกัดได้ โดยการใช้นั่งหรือสองคำสั่งจากเวิร์กสเตชัน แบบโต้ตอบ

ข้อมูลที่เกี่ยวข้อง

รายงานจากประสบการณ์: สภาพะที่ถูกควบคุม

เพราะเหตุใดจึงพิจารณาการใช้หลายระบบย่อย

เมื่อจำนวนของผู้ใช้บนระบบเพิ่มขึ้น ระบบย่อยเพียงระบบเดียว มักจะไม่เพียงพอสำหรับงานต่างๆ โดยการแบ่งผู้ใช้ของคุณลงในหลายระบบย่อย คุณจะได้รับประโยชน์หลายอย่าง

การจัดการงานได้ดีขึ้น

คุณสามารถควบคุมงานที่กำลังรันในแต่ละระบบย่อยได้ดีขึ้น ตัวอย่าง เช่น สำหรับงานเซิร์ฟเวอร์ คุณอาจต้องการแยกงานเซิร์ฟเวอร์ฐานข้อมูล ทั้งหมดไปไว้ที่ระบบย่อยหนึ่ง งานเซิร์ฟเวอร์คำสั่งรีโมตไปไว้ที่อีกระบบย่อยหนึ่ง งานเซิร์ฟเวอร์ DDM ไปไว้ที่อีกระบบย่อยหนึ่ง และอื่นๆ นอกจากนั้น การใช้หลายระบบย่อยยังทำให้คุณสามารถแบ่งกลุ่มของงานด้วยพูลหน่วยความจำ ของงานเองได้ด้วย ในวิธีนี้ งานกลุ่มหนึ่งจะไม่ส่งผลกระทบต่องานอื่น

ผลกระทบ downtime ที่ลดลงสำหรับผู้ใช้

ตัวอย่างเช่น ถ้าทุกตอนบ่ายวันศุกร์ คุณต้องนำระบบเข้าสู่สภาพถูกจำกัดสำหรับการสำรองข้อมูล คุณสามารถนำผู้ใช้ออฟไลน์ได้ทีละน้อย โดยการจบบรรยากาศที่ละหนึ่งระบบ

การกำหนดสเกลและสภาพพร้อมใช้งานที่พัฒนาขึ้น

โดยการใส่ระบบย่อยหนึ่งระบบทำงานให้ผู้ใช้จำนวนน้อยลง ระบบย่อยจึงยุ่ง ลดลงและสามารถตอบสนองต่อคำขอ งานที่ระบบย่อยจัดการได้มากขึ้น

error tolerance ที่พัฒนาขึ้นในระบบย่อยแบบโต้ตอบ

โดยการกระจายงานในหลายระบบย่อย ถ้าเกิดความล้มเหลวของเน็ตเวิร์ก ระบบย่อยหลายระบบสามารถจัดการ กระบวนการกู้คืนอุปกรณ์ได้

เวลาเริ่มทำงานของระบบย่อยแบบโต้ตอบที่พัฒนาขึ้น

คุณสามารถลดเวลาการเริ่มทำงานของระบบย่อยได้โดยการแบ่งงาน ในหลายระบบย่อย

อ็อปชันเพิ่มเติมสำหรับการปรับประสิทธิภาพ

โดยการใส่หลายระบบย่อย คุณจึงสามารถตั้งค่าระบบย่อยโดยมี routing entries จำนวนน้อยได้

ข้อมูลที่เกี่ยวข้อง

รายงานจากประสบการณ์: การคอนฟิกูเรชันระบบย่อย

Subsystem description

Subsystem description คืออ็อบเจกต์ระบบที่มีข้อมูล ซึ่งกำหนดลักษณะของสถานะแวดล้อมการปฏิบัติการที่ควบคุมโดย ระบบ identifier ที่ระบบรับรู้สำหรับชนิดอ็อบเจกต์คือ *SBSD subsystem description กำหนดว่างานจะเข้าสู่ระบบย่อยอย่างไร, ที่ไหน และงานมีจำนวนเท่าใด, และระบบย่อยจะใช้รีซอร์สใดในการปฏิบัติงาน ระบบย่อยแอ็คทีฟจะใช้ชื่อธรรมดาของ subsystem description

เหมือนกับชุดของพิมพ์เขียวที่ละเอียด แต่ละ subsystem description จะไม่ซ้ำกัน, และมีลักษณะเฉพาะตัวที่อธิบายระบบย่อย description ได้แก่งานสามารถเข้าสู่ระบบย่อยได้ที่ใด ระบบย่อยสามารถจัดการงานได้จำนวนเท่าใด จะใช้แหล่งเก็บข้อมูลหลัก (หน่วยความจำ) จำนวนเท่าใด และงานในระบบย่อยสามารถรันได้รวดเร็วเพียงใด

คุณสามารถใช้ subsystem description ที่ได้มาพร้อมกับระบบของคุณ (โดยทำหรือไม่ทำการเปลี่ยนแปลง) หรือคุณสามารถ สร้าง subsystem descriptions ของคุณเองได้

งานที่เกี่ยวข้อง

“การเปลี่ยนรายละเอียดของระบบย่อย” ในหน้า 186

คำสั่ง Change Subsystem Description (CHGSBSD) เปลี่ยนแอ็คทีวิตีการดำเนินการของ subsystem description ที่ระบบ คุณสามารถเปลี่ยน subsystem description ในขณะที่ระบบย่อยแอ็คทีฟ ถ้าต้องการเปลี่ยน subsystem description ให้ใช้อินเตอร์เฟซแบบอ็อกซาระ

“การสร้างรายละเอียดของระบบย่อย” ในหน้า 181

คุณสามารถสร้าง subsystem description ได้สองวิธี คุณสามารถก๊อปปี้ subsystem description ที่มีอยู่และเปลี่ยนแปลง หรือคุณสามารถสร้าง description ใหม่ทั้งหมด

Subsystem description attributes:

Subsystem description attributes คือแอ็คทีวิตีระบบโดยรวมทั่วไป เมื่อคุณสร้างระบบย่อย ขั้นตอนแรกคือการ กำหนดแอ็คทีวิตีระบบย่อย

แอ็ดทริบิวต์ระบบย่อยประกอบด้วย:

- ชื่อของ subsystem description และไลบรารีที่ถูกจัดเก็บอยู่
- definition พูลหน่วยความจำทั้งหมดที่ระบบย่อยนี้ใช้
definition ระบบย่อยหนึ่ง สามารถมี definition พูลหน่วยความจำที่ระบุได้สูงสุด 10 definition ลings ที่ประกอบ อยู่ใน definition ระบบย่อยมีดังนี้:
 - Pool definition identifier: นี่คือ identifier ภายใน subsystem description ของ definition พูลของหน่วยความจำ
 - ขนาด: นี่คือขนาดของพูลของหน่วยความจำที่ระบุเป็นหน่วย KB (1K=1024 ไบต์) และเป็นจำนวนของแหล่งเก็บข้อมูลหลักที่พูลสามารถใช้ได้
 - Activity level: นี่คือจำนวนสูงสุดของ threads ที่สามารถรันพร้อมกัน ในพูล
- จำนวนสูงสุดของงานที่สามารถแอ็ดที่พีในระบบย่อยใน เวลาเดียวกัน
- text description ของ subsystem description
- ชื่อ และไลบรารีของไฟล์แสดงผล sign-on ที่ใช้ในการแสดงจอแสดงผล sign-on ที่เวิร์กสเตชันที่จัดสรรให้กับระบบย่อย
- ชื่อไลบรารีระบบย่อยที่คุณสามารถใช้ได้ ถ้าคุณต้องการระบุไลบรารี ที่ควรป้อนก่อนหน้าไลบรารีอื่นในส่วนรายชื่อไลบรารีของระบบ (พารามิเตอร์นี้ช่วยให้คุณสามารถใช้ไลบรารีภาษาที่สองได้)

นอกจากนี้ subsystem description ยังมีข้อมูลเกี่ยวกับระดับสิทธิในการใช้งาน ระบบย่อยด้วย ข้อมูลนี้ถูกเก็บรักษาโดย การรักษาความปลอดภัย และไม่ถูกจัดเก็บ พร้อมกับแอ็ดทริบิวต์อื่นของ 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

การป้อนข้อมูลงานแบบ Autostart:

การป้อนข้อมูลงานแบบ autostart ระบุให้งานแบบ autostart เริ่มขึ้นในทันทีที่ระบบย่อยเริ่มต้นขึ้น. เมื่อระบบย่อยเริ่มทำงาน, ระบบจะจัดสรรสิ่งต่างๆ ให้ระบบย่อย และเริ่มงานแบบ autostart และ prestart ก่อนที่ระบบย่อยจะพร้อมที่จะ ทำงาน

งานแบบ autostart ที่เชื่อมโยงกับระบบย่อยจะถูกเริ่มต้นโดยอัตโนมัติ ในทุกครั้งที่ระบบย่อยถูกเริ่มขึ้น งานแบบ autostart ในระบบย่อยการควบคุม สามารถใช้ในการเริ่มระบบย่อยอื่น (เช่นระบบย่อยการควบคุม ที่ได้มาจาก IBM) งานแบบ autostart คืองานแบ็คชที่ทำงานซ้ำ

ตัวอย่างเช่น: เพื่อเรียกโปรแกรมการกู้คืนพิเศษถ้า IPL ตรวจสอบว่า ระบบก่อนหน้าจะจบลงอย่างผิดปกติ, คุณสามารถใส่เพิ่ม autostart job entry ที่ subsystem description ของระบบย่อยการควบคุม โปรแกรมนี้ตรวจสอบค่ากำหนดของระบบ Previous system ending status (QABNORMSW) สำหรับการจบระบบตาม ปกติ, ค่าของ QABNORMSW คือ '0', และสำหรับการจบระบบอย่างผิดปกติ, ค่าของ QABNORMSW คือ '1'

งานที่เกี่ยวข้อง

“การใส่เพิ่ม autostart job entries” ในหน้า 182

คุณใช้อินเตอร์เฟซแบบอักษรเพื่อใส่เพิ่ม autostart job entry งานแบบ autostart จะเริ่มโดยอัตโนมัติเมื่อระบบย่อยที่เกี่ยวข้องเริ่มต้นขึ้น โดยทั่วไปงานเหล่านี้จะทำงานการกำหนดค่าเริ่มต้นที่เชื่อมโยงกับระบบย่อยงานแบบ autostart ยังสามารถทำงานที่ซ้ำๆ หรือนำเสนอฟังก์ชัน การให้บริการศูนย์กลางสำหรับงานอื่นที่อยู่ในระบบย่อยเดียวกัน

“การเปลี่ยน autostart job entries” ในหน้า 187

คุณสามารถระบุรายละเอียดของงานอื่นสำหรับ autostart job entry ที่ได้กำหนดไว้ก่อนหน้านี้ ถ้าต้องการเปลี่ยน autostart job entry ให้ใช้อินเตอร์เฟซแบบอักษร

“การลบ autostart job entry” ในหน้า 191

คุณสามารถลบ autostart job entry ออกจาก subsystem description โดยใช้อินเตอร์เฟซแบบอักษร

การป้อนข้อมูลแบบ Communication:

การป้อนข้อมูลงานแบบ communications ระบุให้ระบบย่อยทราบถึงซอร์สสำหรับงานสื่อสารที่ระบบจะประมวลผล การประมวลผลงานเริ่มขึ้นเมื่อระบบย่อยได้รับ คำขอการเริ่มโปรแกรมการสื่อสารจากระบบรีโมตและพบ routing entry ที่เหมาะสมสำหรับคำขอ

แต่เพื่อให้การทำงานมีประสิทธิภาพ, แทนที่จะเริ่มต้นงาน การสื่อสารทุกครั้งที่ได้รับคำขอการเริ่มโปรแกรม, คุณสามารถตั้งค่างานแบบ prestart เพื่อให้จัดการกับคำขอการเริ่มโปรแกรมจาก ระบบรีโมตได้ สำหรับงานแบ็ตซ์การสื่อสารที่จะรันบนระบบ subsystem description ที่มี work entry สำหรับงานสื่อสารต้องมีอยู่บนระบบ

งานที่เกี่ยวข้อง

“การใส่เพิ่ม communications entries” ในหน้า 182

แต่ละ communication entry อธิบายอุปกรณ์สื่อสารหนึ่ง หรือหลายอุปกรณ์, ชนิดของอุปกรณ์ หรือตำแหน่งรีโมตซึ่งระบบย่อยจะเริ่มงาน เมื่อได้รับคำขอการเริ่มโปรแกรม ระบบย่อยสามารถจัดสรรอุปกรณ์สื่อสาร ถ้าอุปกรณ์ไม่ได้ถูกจัดสรรให้กับระบบย่อยหรืองานอื่นอยู่ในปัจจุบัน อุปกรณ์สื่อสารที่ถูกจัดสรรแล้วในปัจจุบันสามารถถูกจัดสรรคืนในภายหลังได้ เพื่อให้อุปกรณ์นั้นพร้อมใช้งานสำหรับระบบย่อยอื่นได้ ถ้าต้องการใส่เพิ่ม communications entry ที่ subsystem description ให้ใช้อินเตอร์เฟซแบบอักษร

“การเปลี่ยน communication entries” ในหน้า 187

คุณสามารถเปลี่ยนแอตทริบิวต์ของ communications entry ที่มีอยู่ใน subsystem description ที่มีอยู่ได้โดยใช้อินเตอร์เฟซแบบอักษร

“การลบ communication entry” ในหน้า 191

คุณสามารถลบ communication entry ออกจาก subsystem description โดยใช้อินเตอร์เฟซแบบอักษร งานทั้งหมดที่แอ็คทีฟผ่านทาง communications entry ที่จะลบออกต้องถูกจบลงก่อน คำสั่งนี้จึงจะสามารถ รันได้

Job queue entry:

Job queue entry ใน subsystem description ระบุ คิวงานซึ่งระบบย่อยจะได้รับงานมา เมื่อระบบย่อยถูกเริ่มขึ้น ระบบย่อยจะพยายามจัดสรรคิวงานแต่ละคิวที่กำหนดใน subsystem job queue entry

ตัวอย่างเช่น job queue entry ใน subsystem description QSYS/QBASE ระบุว่างานสามารถเริ่มโดยใช้คิวงาน QGPL/QBATCH งานสามารถ ถูกวางบนคิวงานได้แม้ว่าระบบย่อยยังไม่ได้ถูกเริ่มต้นขึ้น เมื่อ ระบบย่อย QBASE เริ่มขึ้น ระบบย่อยจะประมวลผลงานบน คิว subsystem description สามารถระบุจำนวนสูงสุดของงาน (แบบแบ็ตซ์ หรือแบบโต้ตอบ) ที่สามารถประมวลผลได้พร้อมกัน จำนวนของงาน ที่สามารถแอ็คทีฟจากคิวงานใดๆ ถูกระบุอยู่ใน job queue entry

งานที่เกี่ยวข้อง

“การใส่เพิ่ม job queue entries” ในหน้า 183

job queue entry ระบุคิวงานซึ่งงานจะถูกเลือกสำหรับการรันในระบบย่อยงานที่เริ่มต้นจากคิวงานคืองานแบ็คคูล์ คุณสามารถใส่เพิ่ม job queue entry โดยใช้อินเตอร์เฟซแบบอักษร

“การเปลี่ยน job queue entry” ในหน้า 188

คุณสามารถเปลี่ยน job queue entry ที่มีอยู่ใน subsystem description ที่ระบุได้ สามารถใช้คำสั่งนี้ในขณะที่ระบบย่อยแอ็คทีฟหรือไม่แอ็คทีฟ ถ้าต้องการเปลี่ยน job queue entry ในระบบย่อย ให้ใช้อินเตอร์เฟซแบบอักษร

“การลบ job queue entry” ในหน้า 191

คุณสามารถลบ job queue entries ออกจาก subsystem description โดยใช้อินเตอร์เฟซแบบอักษร งานบนคิวงานยังคงอยู่บนคิว เมื่อ job queue entry ถูกลบออกจาก subsystem description จะไม่สามารถลบ job queue entry ถ้างานแอ็คทีฟในปัจจุบันใดๆ ถูกเริ่มจาก คิวงาน

Prestart job entry:

คุณกำหนดงานแบบ prestart โดยใช้ prestart job entry prestart job entry ไม่มีผลต่อการจัดสรรอุปกรณ์หรือการกำหนดค่าขอการเริ่มโปรแกรม

แอ็คทีฟชีวิตงานของงานแบบ prestart จะไม่ถูกเปลี่ยนแปลงโดยระบบย่อย เมื่อ คำขอการเริ่มโปรแกรมแนบกับงานแบบ prestart อย่างไรก็ตาม โดยทั่วไปงานเซิร์ฟเวอร์จะเปลี่ยนแอ็คทีฟชีวิตงานเป็นแอ็คทีฟชีวิตของโพรไฟล์ผู้ใช้ที่สลับค่า

คำสั่ง Change Prestart Job (CHGPJ) อนุญาตให้งานแบบ prestart เปลี่ยนแอ็คทีฟชีวิตงานบางอย่างเป็นแอ็คทีฟชีวิตของรายละเอียดของงาน (ที่ระบุในรายละเอียดของงานที่เชื่อมโยงกับโพรไฟล์ของผู้ใช้ของคำขอเริ่มโปรแกรม หรือในรายละเอียดของงานที่ระบุใน prestart job entry)

งานแบบ prestart สำหรับเซิร์ฟเวอร์:

ในต้นแบบงานแบบ prestart จะมีงานการฟังหลักหนึ่งงาน โดยทั่วไปเรียกว่างาน daemon หรืองาน listener และหลายงานเซิร์ฟเวอร์ที่ประมวลผล คำขอไคลเอ็นต์งาน daemon จะรอฟังคำขอการเชื่อมต่ออยู่บนพอร์ต เมื่อได้รับการเชื่อมต่อใหม่ daemon จะทำงานทั่วไป แล้วให้ socket descriptor ที่งานเซิร์ฟเวอร์แบบ prestart ที่กำลังรออยู่

งานแบบ prestart สามารถถูกใช้ใหม่ได้ เมื่องานทำงานสำหรับไคลเอ็นต์หนึ่งเสร็จสมบูรณ์ แล้ว สภาวะแวดล้อมจะถูกรีเซ็ตและงานจะถูกทำให้พร้อมที่จะจัดการ คำขอจากไคลเอ็นต์อื่น

สำหรับงานเซิร์ฟเวอร์ที่รันโค้ดผู้ใช้ (เช่น เซิร์ฟเวอร์คำสั่งรีโมต) โดยปกติงานจะไม่ถูกนำมาใช้ใหม่ที่เป็นเช่นนี้เนื่องจากโค้ดผู้ใช้อาจเปลี่ยนบางสิ่งในงาน และไม่มีวิธีที่แน่นอนในการรีเซ็ตสภาวะแวดล้อมสำหรับไคลเอ็นต์ใหม่ ถ้าเซิร์ฟเวอร์ของคุณนำงานมาใช้ใหม่สามารถใช้ Change Job (QWTCGJB) API เพื่อเปลี่ยนแอ็คทีฟชีวิตของงานกลับไปยังสภาพที่รู้จัก หลังจากคำขอของไคลเอ็นต์ได้เสร็จสมบูรณ์แล้ว

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

หลักการที่เกี่ยวข้อง

“การสืบสวนงานแบบ prestart” ในหน้า 244

หัวข้อนี้นำเสนอขั้นตอนที่ช่วยคุณในการตอบคำถาม “ฉันจะ ค้นหาผู้ใช้จริงของงานแบบ prestart และกำหนดรีซอร์สที่ใช้โดยงานแบบ prestart นั้นได้อย่างไร?”

ข้อมูลที่เกี่ยวข้อง

รายงานจากประสบการณ์: การปรับ prestart job entry

Workstation entries:

งานแบบโต้ตอบคืองานที่เริ่มขึ้นเมื่อผู้ใช้ signs on ที่จอภาพ และสิ้นสุดลงเมื่อผู้ใช้ signs off สำหรับงานที่จะรัน ระบบย่อย จะค้นหารายละเอียดของงาน ซึ่งอาจจะอยู่ใน workstation entry หรือโปรไฟล์ผู้ใช้

workstation entry ช่วยให้ระบบย่อยทราบถึงเวิร์กสเตชันที่เป็นไปได้ หากเวิร์กสเตชันพร้อมใช้งานได้ ระบบย่อยจะส่งจอภาพ sign-on ไปที่จอแสดงผล

หมายเหตุ: subsystem description สำหรับระบบย่อยการควบคุมต้องมี workstation entry สำหรับคอนโซล และ entry นั้นต้องเป็นชนิด *SIGNON (*SIGNON คือ คำสำหรับพารามิเตอร์ AT ที่ระบุบนคำสั่ง Add Work Station Entry (ADDWSE)) คำ *SIGNON บ่งชี้ว่าหน้าจอ sign-on จะถูกแสดงขึ้นที่เวิร์กสเตชันเมื่อระบบย่อยเริ่มต้นขึ้น ข้อกำหนดนี้ทำให้มั่นใจได้ว่า ระบบย่อยมีอุปกรณ์แบบโต้ตอบสำหรับการป้อนคำสั่งระดับระบบ และระบบย่อย คำสั่ง End System (ENDSYS) จะจบ System i โปรแกรมที่ไคไลเซนส์เป็น single session (หรือจอแสดงผล sign-on) ที่คอนโซลในระบบย่อยการควบคุม subsystem description ที่ไม่มี workstation entry สำหรับคอนโซลไม่สามารถถูกเริ่มเป็นระบบย่อยการควบคุม

งานที่เกี่ยวข้อง

“การใส่เพิ่ม workstation entries” ในหน้า 184

workstation entry จะถูกใช้เมื่องานเริ่มต้น เมื่อผู้ใช้ sign on หรือถ่ายโอนงานแบบโต้ตอบจากระบบย่อยอื่น คุณสามารถระบุไอเท็มดังต่อไปนี้ใน workstation entry ชื่อพารามิเตอร์แสดงอยู่ในวงเล็บ ใช้อินเตอร์เฟซแบบอักขระเพื่อใส่เพิ่ม workstation entries

“การเปลี่ยน workstation entry” ในหน้า 189

คุณสามารถระบุรายละเอียดของงานอื่นสำหรับ workstation entry ที่ได้กำหนดไว้ก่อนหน้านี้ได้โดยใช้อินเตอร์เฟซแบบอักขระ

“การลบ workstation entry” ในหน้า 193

คุณสามารถลบ workstation entry ออกจาก subsystem description โดยใช้อินเตอร์เฟซแบบอักขระ ระบบย่อยสามารถแก้ไขที่พีในเวลาที่รัน คำสั่งได้ อย่างไรก็ตาม งานทั้งหมดที่แก้ไขที่ผ่านทาง workstation entry ต้องถูกจบก่อน entry จึงจะสามารถลบออกได้

routing entry:

routing entry ระบุพูลระบบย่อยของแหล่งเก็บข้อมูลหลักที่จะใช้, โปรแกรมการควบคุมที่จะรัน (โดยปกติคือ โปรแกรมที่ระบบจัดให้ QCMD), และข้อมูลรันไทม์เพิ่มเติม (จัดเก็บอยู่ในคลาสอ็อบเจกต์) Routing entries ถูกจัดเก็บใน subsystem description

routing entry สามารถถือเสมือนเป็น entry เดียวในไดเรกทอรี shopping mall ลูกค้าที่ไม่สามารถหาร้านค้าที่ต้องการสามารถใช้ไดเรกทอรี เพื่อช่วยให้ทราบ ทิศทางที่ถูกต้องได้ เช่นเดียวกับบนระบบ Routing entries ช่วยให้งานทราบถึงตำแหน่งที่ต้องการ Routing entries ใน subsystem description ระบุโปรแกรมที่จะถูกเรียกเพื่อควบคุมขั้นตอนการเราต์สำหรับงานที่รันอยู่ในระบบย่อย, พูลหน่วยความจำที่งานจะใช้ และคลาสซึ่งจะเรียกแอ็ททริบิวต์รันไทม์มา ข้อมูลการเราต์ระบุ routing entry สำหรับงานที่จะใช้ในขณะเดียวกัน, routing entries และข้อมูลการเราต์นำเสนอข้อมูลเกี่ยวกับการเริ่มงานในระบบย่อย

Routing entries ประกอบด้วยส่วนเหล่านี้; subsystem description, คลาส, ข้อมูลการเปรียบเทียบ, ขั้นตอนการเรอต์สูงสุดที่แ็คทีฟ, memory pool ID, โปรแกรมที่จะเรียก, thread resources affinity, resources affinity group, และหมายเลขลำดับงานที่เกี่ยวข้อง

“การใส่เพิ่ม routing entries” ในหน้า 184

แต่ละ routing entry ระบุพารามิเตอร์ที่ใช้ในการเริ่มขั้นตอนการเรอต์สำหรับงาน Routing entries ระบุพารามิเตอร์ของแหล่งเก็บข้อมูลหลักที่จะใช้โปรแกรมการควบคุมที่จะรัน (โดยปกติคือ โปรแกรมที่ระบบจัดให้ QCMD), และข้อมูลรันไทม์เพิ่มเติม (จัดเก็บอยู่ในคลาสอ็อบเจกต์) ถ้าต้องการใส่เพิ่ม routing entry ที่ subsystem description, ให้ใช้อินเตอร์เฟซแบบอักษร

“การเปลี่ยน routing entry” ในหน้า 189

คุณสามารถเปลี่ยน routing entry ใน subsystem description ที่ระบุได้โดยใช้อินเตอร์เฟซแบบอักษร routing entry ระบุพารามิเตอร์ที่จะใช้ในการเริ่มขั้นตอนการเรอต์สำหรับงานหนึ่ง ระบบย่อยที่เกี่ยวข้องสามารถแ็คทีฟ เมื่อทำการเปลี่ยนแปลง

“การลบ routing entry” ในหน้า 192

คุณสามารถลบ routing entry ออกจาก subsystem description ที่ระบุได้โดยใช้อินเตอร์เฟซแบบอักษร ระบบย่อยสามารถแ็คทีฟในเวลาที่ยังค้างอยู่ อย่างไรก็ตาม ถ้ามีงานแ็คทีฟในปัจจุบันใดๆ ที่ถูกเริ่มต้นโดยใช้ entry

คลาส:

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

แ็คทีฟวิวด์รันไทม์ที่มีอยู่ใน routing entry class คือ:

Run priority (RUNPTY)

ระดับความสำคัญการรันคือช่วงของค่าตั้งแต่ 1 (ระดับความสำคัญสูงสุด) ถึง 99 (ระดับความสำคัญต่ำสุด) ที่แสดงถึงระดับความสำคัญซึ่งงานแย่งชิงหน่วยการประมวลผลเมื่อเปรียบเทียบกับงานอื่นที่แ็คทีฟในเวลาเดียวกัน สำหรับงานแบบ multi-threaded, ระดับความสำคัญการรันคือระดับความสำคัญการรันสูงสุดที่ใช้ได้สำหรับ thread ใดๆ ภายในงาน แต่ละ threads ภายในงานอาจมี ระดับความสำคัญต่ำกว่า

Time slice (TIMESLICE)

การแบ่งเวลาเป็นตัวกำหนดจำนวนเวลาที่ต้องการโดย thread ในงานในการทำให้การประมวลผลจำนวนหนึ่งเสร็จสมบูรณ์ เมื่อสิ้นสุดการแบ่งเวลา thread อาจถูกกำหนดให้มีสถานะ inactive เพื่อให้ threads อื่นสามารถแ็คทีฟในพูลของหน่วยความจำได้

Default wait time (DFTWAIT)

ค่านีระบุเวลาสูงสุดดีฟอลต์ (ในวินาที) ที่ thread ใน งานจะคอยคำสั่งระบบ เช่นคำสั่ง LOCK machine interface (MI), เพื่อให้ได้รีซอร์ส จะใช้ช่วงเวลารอดดีฟอลต์ นี้ถ้าไม่มีการระบุช่วงเวลารอค่าอื่นสำหรับสถานการณ์ที่กำหนดให้ โดยทั่วไป ช่วงเวลานี้เป็นเวลา ที่ผู้ใช้ระบบเต็มใจที่จะรอระบบก่อนที่จะจบคำขอ ถ้ารอนานเกินช่วงเวลารอสำหรับคำสั่งใดๆ ข้อความแสดงความผิดพลาดอาจถูกแสดงขึ้นหรืออาจถูกจัดการโดยอัตโนมัติโดยคำสั่ง Monitor Message (MONMSG)

Maximum CPU time (CPUTIME)

ค่านี้ระบุเวลาสูงสุดของหน่วยการประมวลผล (ในมิลลิวินาที) ที่งานสามารถใช้ได้ ถ้างานประกอบด้วยขั้นตอนการเรอต์หลายขั้นตอน แต่ละขั้นตอน การเรอต์สามารถใช้เวลาของหน่วยการประมวลผลนี้ได้ ถ้าใช้เกินเวลาสูงสุดงานจะถูกจบ

Maximum temporary storage (MAXTMPSTG)

ค่านี้ระบุจำนวนสูงสุดของหน่วยเก็บชั่วคราว (auxiliary) ที่งานสามารถใช้ได้ ถ้างานประกอบด้วยขั้นตอนการเรอต์หลายขั้นตอน, ค่านี้คือ หน่วยเก็บชั่วคราวสูงสุดที่ขั้นตอนการเรอต์สามารถใช้ได้ หน่วยเก็บชั่วคราวนี้ใช้สำหรับ หน่วยเก็บที่ต้องการโดยตัวโปรแกรมเองและโดยอ็อบเจกต์ระบบภายใน ที่สร้างโดย ทางอ้อมซึ่งใช้ในการสนับสนุนงาน ค่านี้ไม่รวมหน่วยเก็บ ในไลบรารี QTEMP ถ้าใช้เกินหน่วยเก็บชั่วคราวสูงสุด, งานจะถูกจบ พารามิเตอร์นี้ไม่ได้ใช้กับการใช้หน่วยเก็บถาวร ซึ่งถูก ควบคุมผ่านทางโปรไฟล์ผู้ใช้

Maximum threads (MAXTHD)

ค่านี้ระบุจำนวนสูงสุดของ threads ที่งานซึ่งใช้คลาสนี้ สามารถรันได้ทุกเมื่อ ถ้าหลาย thread ถูกเริ่มต้นพร้อมกัน ค่านี้อาจมากเกินได้ ถ้าใช้เกินค่าสูงสุดนี้ thread ที่มากเกินไป จะได้รับอนุญาตให้รันจนเสร็จสมบูรณ์ตามปกติ การเริ่มต้น thread เพิ่มเติมจะถูกห้าม จนกว่าจำนวนสูงสุดของ threads ในงานจะลดลงต่ำกว่าค่าสูงสุดนี้

Text description (TEXT)

ค่านี้ระบุข้อความที่อธิบายอ็อบเจกต์อย่างคร่าวๆ นี้คือแอตทริบิวต์ ของคลาสอ็อบเจกต์เมื่อคลาสอ็อบเจกต์ถูกสร้างขึ้น แต่ไม่ใช่ แอตทริบิวต์รันไทม์สำหรับงาน

Authority (AUT)

ค่านี้ระบุสิทธิที่คุณให้แก่ผู้ใช้ที่ไม่มีสิทธิเฉพาะ สำหรับอ็อบเจกต์, ผู้ใช้ที่ไม่ได้อยู่บน authorization list และผู้ใช้ที่มีโปรไฟล์กลุ่มหรือโปรไฟล์กลุ่มเสริมที่ไม่มีสิทธิเฉพาะ สำหรับอ็อบเจกต์ นี้คือแอตทริบิวต์ ของคลาสอ็อบเจกต์เมื่อคลาสอ็อบเจกต์ถูกสร้างขึ้น แต่ไม่ใช่ แอตทริบิวต์รันไทม์สำหรับงาน

ข้อมูลการเปรียบเทียบ:

พารามิเตอร์ comparison value (CMPVAL) ของ routing entry ระบุข้อมูลที่จะถูกเปรียบเทียบกับข้อมูลการเรอต์ เพื่อกำหนด routing entry ที่จะใช้ (routing entry ยังระบุตำแหน่งเริ่มต้นสำหรับการเปรียบเทียบด้วย) ข้อมูลการเรอต์จะถูกเปรียบเทียบกับค่าการเปรียบเทียบ ของแต่ละ routing entry ตามหมายเลขลำดับจนกว่าจะพบข้อมูลที่ตรงกัน หมายเลขลำดับที่อยู่ใน routing entry กำหนดลำดับในการสแกน routing entries และสามารถใช้เป็น identifier ของ routing entry

เมื่อพบ routing entry ที่มีค่าเปรียบเทียบตรงกับข้อมูลการเรอต์, ขั้นตอนการเรอต์จะถูกเริ่มขึ้น และโปรแกรมที่ระบุใน routing entry จะถูกเรียก แอตทริบิวต์รันไทม์ในคลาสที่เชื่อมโยงกับ routing entry จะถูกใช้สำหรับขั้นตอนการเรอต์, และขั้นตอนการเรอต์จะรันใน พูลของหน่วยความจำที่ระบุใน routing entry

คุณสามารถระบุค่าเปรียบเทียบของ *ANY บน routing entry ที่มีหมายเลขสูงสุด *ANY หมายความว่าบังคับให้จับคู่โดยไม่คำนึงว่าข้อมูลการเรอต์จะเป็นอะไรก็ตาม เพียง routing entry เท่านั้นที่สามารถมีค่าเปรียบเทียบเป็น *ANY และต้องเป็น entry หลังสุด (หมายเลขลำดับสูงสุด) ใน subsystem description

ขั้นตอนการเรอต์แอคทีฟสูงสุด:

พารามิเตอร์ maximum active routing steps (MAXACT) ของ routing entry ระบุจำนวนสูงสุดของขั้นตอนการเรอต์ (งาน) ที่สามารถแอคทีฟพร้อมกันผ่าน routing entry นี้

ในงานหนึ่ง จะมีขั้นตอนการเรอที่แ็คทีฟเพียงหนึ่งขั้นตอนเท่านั้นในแต่ละเวลา เมื่อระบบย่อย แ็คทีฟและใช้งานถึงจำนวนสูงสุดของขั้นตอนการเรอ ความพยายามต่อมา ในการเริ่มขั้นตอนการเรอผ่านทาง routing entry นี้จะล้มเหลวงานที่พยายามเริ่มขั้นตอนการเรอจะถูกจบ และระบบย่อยจะส่งข้อความ ไปที่ไฟล์บันทึกของงาน

โดยปกติ ไม่มีเหตุผลที่ต้องควบคุมจำนวนของขั้นตอนการเรอ ดังนั้น ค่าที่แนะนำคือ *NOMAX

Memory pool ID:

พารามิเตอร์ memory pool ID (POOLID) ของ routing entry ระบุ pool identifier ของพูลหน่วยความจำซึ่งโปรแกรมจะรันใน นั้น pool identifier ที่ระบุที่นี้เกี่ยวข้องกับพูลของหน่วยความจำ ใน subsystem description

โปรแกรมที่จะเรียก

พารามิเตอร์ program to call (PGM) ของ routing entry ระบุชื่อ และไลบรารีของโปรแกรมที่เรียกเป็นโปรแกรมแรกที่จะรันใน ขั้นตอนการเรอที่ไม่สามารถส่งผ่านพารามิเตอร์ใด ไปที่โปรแกรมที่ระบุ ชื่อโปรแกรมอาจถูกระบุโดยทางตรง ใน routing entry หรือดึงข้อมูลมาจากข้อมูลการเรอ อย่างไม่อย่างหนึ่ง

ถ้าชื่อ โปรแกรมถูกระบุใน routing entry การเลือก routing entry นั้นส่งผลให้ routing entry program ถูกเรียก (ไม่คำนึงถึงชื่อ โปรแกรม ที่ส่งผ่านในฟังก์ชัน EVOKE) ถ้าจะเรียกโปรแกรมที่ระบุในฟังก์ชัน EVOKE, ต้องระบุ *RTGDTA ในพารามิเตอร์ นี้ ถ้าโปรแกรมไม่มีอยู่เมื่อใส่เพิ่มหรือเปลี่ยน routing entry ต้องระบุ library qualifier เนื่องจากชื่อโปรแกรมที่ถูกต้อง ถูกจัด เก็บใน subsystem description

หมายเลขลำดับ

พารามิเตอร์ sequence number (SEQNBR) ของ routing entry จะบอกให้ระบบย่อยทราบถึงลำดับในการค้นหาข้อมูลการเรอ ที่ตรงกัน ใน routing entry routing entry จะถูกค้นหา ตามหมายเลขลำดับ เมื่อคุณใส่เพิ่ม routing entry ที่ subsystem description คุณควรจัดลำดับ entry เพื่อให้ entry ที่จะถูกเปรียบเทียบบ่งชี้ที่สุด เป็น entry แรก การทำเช่นนี้จะช่วยลดเวลาการค้นหา

หมายเลขลำดับ	ค่าการเปรียบเทียบ
10	'ABC'
20	'AB'
30	'A'
40	'E'
50	'D'

ในตัวอย่างข้างต้น routing entry จะถูกค้นหาตามหมายเลข ลำดับ ถ้าข้อมูลการเรอคือ 'A' การค้นหาจะจบด้วย routing entry 30 ถ้าข้อมูลการเรอคือ 'AB' การค้นหาจะจบด้วย routing entry 20 ถ้าข้อมูลการเรอคือ 'ABC' การค้นหาจะจบด้วย routing entry 10 เนื่องจาก ข้อมูลการเรอสามารถยาวกว่าค่าเปรียบเทียบของ routing entry การเปรียบเทียบ (ซึ่งทำในลำดับจากซ้าย ไปขวา) จะหยุดเมื่อมาถึง ตอนจบของค่าเปรียบเทียบ ดังนั้น ถ้าข้อมูลการเรอคือ 'ABCD' การค้นหาจะจบด้วย routing entry 10

เมื่อคุณกำหนด routing entry entry ต้องถูกจัดลำดับจาก entry ที่เฉพาะที่สุดไปถึง entry ทั่วไปที่สุด ตัวอย่างต่อไปนี้ แสดงวิธีที่ ถูกต้องและไม่ถูกต้องในการกำหนด routing entry:

	ถูกต้อง		ไม่ถูกต้อง
หมายเลขลำดับ	ค่าการเปรียบเทียบ	หมายเลขลำดับ	ค่าการเปรียบเทียบ
10	'ABC'	10	'ABC'
20	'AB'	20	'ABCD'
30	'A'		
40	'E'		
9999	*ANY		

ในตัวอย่างวิธีที่ไม่ถูกต้อง ไม่สามารถจับคู่ routing entry 20 เนื่องจากข้อมูลการเรดตี้ใดๆ ที่ตรงกับค่าเปรียบเทียบสำหรับ routing entry 20 จับคู่กับ routing entry 10 ก่อน เมื่อเปลี่ยนหรือใส่เพิ่ม routing entry ที่ subsystem description ด้วยค่าเปรียบเทียบที่ทำให้เกิดสถานการณ์นี้ ระบบจะส่งข้อความวินิจฉัยที่ระบุสถานการณ์

โปรแกรม ที่มีชื่อใน routing entry จะถูกควบคุมเมื่อขั้นตอนการเรดตี้ สำหรับงานเริ่มขึ้น พารามิเตอร์ในการควบคุมสถานะแวดล้อมรันไทม์ (ระดับความสำคัญ, การแบ่งเวลา และอื่นๆ) ของขั้นตอนการเรดตี้สำหรับงานจะถูกนำมาจากคลาสที่ได้รับใน routing entry

วิธีการเริ่มระบบย่อย

เมื่อระบบย่อยเริ่มต้น ระบบจะจัดสรรสิ่งต่างๆ ให้ระบบย่อยและเริ่ม งานแบบ autostart และงานแบบ prestart ก่อนที่ระบบย่อยจะพร้อมสำหรับงาน

subsystem description ถูกใช้ในการกำหนดวิธีการจัดการไอเท็มต่างๆ รายการต่อไปนี้แสดงลำดับของเหตุการณ์ที่เกิดขึ้นเมื่อระบบย่อยเริ่มทำงาน:

1. มีการออกคำขอเพื่อเริ่มระบบย่อย มีการออกคำสั่ง Start Subsystem (STRSBS) command is issued ระบุตำแหน่งข้อมูล startup หลักใน subsystem description
2. พูลหน่วยความจำถูกจัดสรร หน่วยความจำจะถูกจัดสรรให้กับพูลที่ได้กำหนดไว้ใน subsystem description หน่วยความจำที่ถูกจัดสรรให้กับพูลที่กำหนดไว้แต่ละอันถูกดึงมาจากพูลหน่วยความจำ หลัก ระบบจะไม่จัดสรรหน่วยความจำไปที่พูล ถ้าจำนวนของหน่วยความจำที่พร้อมใช้งานที่พูลของหน่วยความจำ Base น้อยกว่า ขนาดต่ำสุดที่ระบุโดยค่ากำหนดของระบบ 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 sign-on ปรากฏขึ้น) ถ้ามี workstation entries และอุปกรณ์ถูก vary on และยังไม่ได้ถูกจัดสรรโดยระบบย่อยอื่น ระบบย่อยสามารถจัดสรรอุปกรณ์ และแสดงจอแสดงผล sign-on ถ้าอุปกรณ์ถูก vary on และได้ถูกจัดสรรโดยระบบย่อยอื่นแล้ว และอยู่ที่จอแสดงผล sign-on (จอแสดงผล sign-on ถูกแสดงขึ้นก่อนที่ระบบย่อยที่สองจะเริ่มต้น) ระบบย่อยที่สองสามารถจัดสรรอุปกรณ์จากระบบย่อยแรก และแสดงจอแสดงผล sign-on ถ้าอุปกรณ์ไม่ได้ถูก vary on ระบบย่อยจะไม่สามารถจัดสรรอุปกรณ์ งาน system arbiter (QSYSARB) และงาน QCMNARB พักล็อกบนอุปกรณ์ที่ vary-off ทั้งหมด Workstation entries ให้ข้อมูลว่าอุปกรณ์ใดจะถูกตรวจสอบการจัดสรร

หมายเหตุ: สำหรับอุปกรณ์แสดงผลเสมือน จอแสดงผล sign-on จะปรากฏขึ้นเมื่ออุปกรณ์ถูก vary on ทั้งหมด สิ่งนี้เกิดขึ้นเมื่อผู้ใช้เชื่อมต่อกับ System i โดยใช้ device description (สมมติว่าคำขอการเชื่อมต่อไม่มีข้อมูลที่จะใช้ใน

การข้ามกระบวนการแสดงผล sign-on) อุปกรณ์อาจถูกนำมาจาก พูลของ device descriptions ที่สร้างขึ้น ก่อนหน้านี้และถูก varied on เป็นส่วนหนึ่งของกระบวนการเชื่อมต่อนั้น หรืออุปกรณ์สามารถถูกสร้างและ vary on ได้ เมื่อระบบย่อยเริ่มขึ้น ระบบย่อยจะคงค้างล็อกสำหรับ device description ที่สร้างขึ้นก่อนหน้านี้ซึ่ง ระบบย่อยต้องการ

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

งานที่เกี่ยวข้อง

“การเริ่มระบบย่อย” ในหน้า 180

คำสั่ง Start Subsystem (STRSBS) จะเริ่มระบบย่อย โดยใช้ subsystem description ที่ระบุในคำสั่ง เมื่อระบบย่อย เริ่มต้น ระบบจะจัดสรรรีซอร์สที่จำเป็นและพร้อมใช้งาน (หน่วยเก็บ, เวิร์กสเตชัน และคิวงาน) ที่ถูกระบุใน subsystem description คุณสามารถเริ่มระบบย่อย โดยใช้ System i Navigator อินเทอร์เน็ต หรืออินเทอร์เน็ตแบบอักขระ

วิธีการจัดสรรอุปกรณ์เวิร์กสเตชัน:

ระบบย่อยพยายามจัดสรรอุปกรณ์เวิร์กสเตชันทั้งหมดใน subsystem description สำหรับ AT(*SIGNON) workstation entries

สถานการณ์ต่อไปนี้อาจเกิดขึ้นในระหว่างเวลาที่ระบบย่อยเริ่มต้น:

- ถ้าอุปกรณ์ไม่ได้ถูก vary on ระบบย่อยจะไม่สามารถจัดสรรอุปกรณ์ งาน system arbiter (QSYSARB) และ QCMNARBxx พักล็อกบนอุปกรณ์ที่ถูก vary-off ทั้งหมด
- ถ้าอุปกรณ์ถูก vary on และยังไม่ได้ถูกจัดสรรโดยระบบย่อยอื่น ระบบย่อยสามารถจัดสรรอุปกรณ์ และแสดงจอแสดงผล sign-on
- ถ้าอุปกรณ์ถูก vary on และได้ถูกจัดสรรโดยระบบย่อยอื่นแล้ว และอยู่ที่จอแสดงผล sign-on (จอแสดงผล sign-on ถูก แสดงขึ้นก่อนที่ระบบย่อยที่สองจะเริ่มต้น) ระบบย่อยที่สองสามารถจัดสรรอุปกรณ์จากระบบย่อยแรก และแสดงจอแสดงผล sign-on

ถ้ามีมากกว่าหนึ่งระบบย่อยพยายามจัดสรรเวิร์กสเตชันเดียวกัน (ดังที่ระบุใน workstation entries) และเวิร์กสเตชันถูก vary off จะไม่สามารถคาดการณ์ ระบบย่อยที่เรียกเวิร์กสเตชันเมื่อเวิร์กสเตชันถูก vary on ในลักษณะคล้ายกัน ถ้า workstation entry ระบุชนิดเวิร์กสเตชันแทนชื่อเวิร์กสเตชัน ระบบย่อยอาจจะเรียกเวิร์กสเตชันชนิดนั้นทั้งหมด, บางส่วน หรือไม่เรียก เวิร์กสเตชันชนิดนั้น (สิ่งนี้ใช้กับ workstation entries ที่มีชื่อทั่วไป ด้วย) เพื่อหลีกเลี่ยงสถานการณ์ดังกล่าว คุณสามารถตั้งค่า workstation entries สำหรับระบบย่อย เพื่อไม่ให้หลายระบบย่อยใช้เวิร์กสเตชันเดียวกัน

หลังจากผู้ใช้ได้ sign on

เมื่อผู้ใช้ sign on ที่เวิร์กสเตชัน งานจะรันในระบบย่อยที่ถูกแสดงบนจอแสดงผล sign-on บนเวิร์กสเตชัน (ระบบย่อยถูกระบุ ในจอแสดงผล sign-on ที่ได้มาจาก IBM) สถานการณ์ต่อไปนี้ อาจเกิดขึ้นหลังจากที่ผู้ใช้ได้ sign on:

- ถ้าระบบย่อยที่สองเริ่มขึ้นและพยายามที่จะจัดสรรเวิร์กสเตชัน ซึ่งผู้ใช้ signed on บนนั้น ระบบย่อยที่สองจะไม่สามารถจัด สรรเวิร์กสเตชัน งานของ ผู้ใช้ยังคงรันต่อไปในระบบย่อยแรก
- ถ้าผู้ใช้เลือกอ็อปชัน 1 (Display sign-on for alternative job) บนเมนู System Request หรือออกคำสั่ง Transfer to Secondary Job (TFRSECJOB) งานใหม่จะรันบนระบบย่อยเดียวกันกับงานแรก

- เมื่อผู้ใช้ sign off เวิร์กสเตชันจะยังคงถูกจัดสรรที่ระบบย่อยที่ใช้ เมื่อผู้ใช้ sign on ยกเว้นว่าผู้ใช้ได้ถ่ายโอนเข้าในระบบย่อยโดยใช้คำสั่ง Transfer Job (TFRJOB) และระบุ AT (*ENTER) สำหรับ workstation entry สำหรับเวิร์กสเตชันนี้ จอแสดงผล sign-on ถูกแสดงขึ้น และงานต่อมาได้จากเวิร์กสเตชันนั้นยังคงรันต่อไปในระบบย่อย (ยกเว้นว่ามี การเริ่มต้นระบบย่อยอื่นที่จัดสรรเวิร์กสเตชันในขณะที่มันอยู่ที่จอแสดงผล sign-on)
- ถ้าผู้ใช้ sign off และระบบย่อยซึ่งงานของผู้ใช้เคอร์เนลได้จบลง อุปกรณ์จะถูกจัดสรรคืน จากนั้น ระบบย่อยที่สองสามารถจัดสรรอุปกรณ์ และแสดงจอแสดงผล sign-on

งานที่เกี่ยวข้อง

การกำหนดผู้ใช้ที่ระบบย่อยเฉพาะ

คุณสามารถใช้เทคนิคหลายอย่างในการกำหนดชื่ออุปกรณ์ จากนั้น จึงเชื่อมโยงชื่ออุปกรณ์เหล่านั้นกับผู้ใช้ หลังจากเสร็จสมบูรณ์แล้ว คุณสามารถใช้ workstation entry เพื่อนำผู้ใช้ไปที่ระบบย่อยที่ต้องการ

“การกำหนดผู้ใช้ที่ระบบย่อยเฉพาะ” ในหน้า 196

คุณสามารถใช้เทคนิคหลายอย่างในการกำหนดชื่ออุปกรณ์ จากนั้น จึงเชื่อมโยงชื่ออุปกรณ์เหล่านั้นกับผู้ใช้ หลังจากเสร็จสมบูรณ์แล้ว คุณสามารถใช้ workstation entry เพื่อนำผู้ใช้ไปที่ระบบย่อยที่ต้องการ

ข้อมูลที่เกี่ยวข้อง

รายงานจากประสบการณ์: การคอนฟิกูเรชันระบบย่อย

การใช้ Telnet exit point program

สถานการณ์จำลอง: การจัดสรรเวิร์กสเตชัน:

ตัวอย่างนี้แสดงถึงวิธีการจัดสรรสองเวิร์กสเตชันไปที่ระบบย่อยสองระบบต่างกัน

ในสถานการณ์นี้ ระบบย่อย A และระบบย่อย B มีเวิร์กสเตชัน DSP01 และ DSP02 ใน subsystem description (workstation entries ระบุ AT(*SIGNON))

ชื่ออุปกรณ์	จัดสรรไปที่
DSP01	ระบบย่อย A
DSP02	ระบบย่อย A

สมมติว่าทั้งสองเวิร์กสเตชันถูก varied on เมื่อระบบย่อย A เริ่มต้น

ระบบย่อย A จัดสรรทั้งสองเวิร์กสเตชัน และแสดงจอแสดงผล sign-on บนทั้งสองเวิร์กสเตชัน แม้ว่าระบบย่อย A มีจอแสดงผล sign-on ที่แสดงบนเวิร์กสเตชัน เวิร์กสเตชันสามารถถูกจัดสรรโดยระบบย่อย หรืองานอื่น จากนั้น เวิร์กสเตชันจะไม่พร้อมใช้งานที่ระบบย่อย A อีกต่อไป

ชื่ออุปกรณ์	จัดสรรไปที่
DSP01	USER1
DSP02	ระบบย่อย A

เมื่อผู้ใช้ (USER1) sign on ที่เวิร์กสเตชัน DSP01 อุปกรณ์จะถูกจัดสรรที่งานของ USER1 ซึ่งกำลังรันในระบบย่อย A เวิร์กสเตชัน DSP02 ยังคงอยู่ที่จอแสดงผล sign-on ดังนั้นจึงสามารถถูกจัดสรรโดยระบบย่อยหรืองานอื่นได้ จากนั้น เวิร์กสเตชันจะไม่

พร้อมใช้งานที่ระบบย่อย A อีกต่อไป

ชื่ออุปกรณ์	จัดสรรไปที่
DSP01	USER1
DSP02	ระบบย่อย B

ระบบย่อย B เริ่มต้น เนื่องจาก USER1 ได้ sign on ที่เวิร์กสเตชัน DSP01 ระบบย่อย B จึงไม่สามารถจัดสรรอุปกรณ์ ระบบย่อย B ร้องขอการจัดสรรของ อุปกรณ์เมื่ออุปกรณ์พร้อมใช้งาน DSP02 จะถูกจัดสรรที่ระบบย่อย B เนื่องจากไม่มีใคร sign on ที่อุปกรณ์ในระบบย่อย A งานใดๆที่เริ่มบน DSP02 จะรันในระบบย่อย B

ชื่ออุปกรณ์	จัดสรรไปที่
DSP01	ระบบย่อย A
DSP02	ระบบย่อย B

USER1 sign off เนื่องจากงานผู้ใช้เคยรันอยู่ในระบบย่อย A ที่ระบบย่อยแสดงจอตแสดงผล sign-on เพื่อให้ผู้อื่นสามารถ sign on เวิร์กสเตชัน และรันในระบบย่อย A ถ้าระบบย่อย A ถูกจบ เวิร์กสเตชัน DSP01 จะถูกจัดสรรโดยระบบย่อย B (เนื่องจากมีคำขอที่จะจัดสรรอุปกรณ์ค้างอยู่)

ชื่อของระบบย่อยที่มีเวิร์กสเตชันที่จัดสรรอยู่ในปัจจุบันปรากฏขึ้นในมุมมองของจอแสดงผล sign-on ที่ได้มาจาก IBM งานที่เกี่ยวข้อง

“การกำหนดผู้ใช้ที่ระบบย่อยเฉพาะ” ในหน้า 196

คุณสามารถใช้เทคนิคหลายอย่างในการกำหนดชื่ออุปกรณ์ จากนั้น จึงเชื่อมโยงชื่ออุปกรณ์เหล่านั้นกับผู้ใช้ หลังจากเสร็จสมบูรณ์แล้ว คุณสามารถใช้ workstation entry เพื่อนำผู้ใช้ไปที่ระบบย่อยที่ต้องการ

ข้อมูลที่เกี่ยวข้อง

การใช้ Telnet exit point program

พูลหน่วยความจำ

พูลหน่วยความจำเป็นส่วนโลจิคัลของหน่วยความจำหลักหรือหน่วยเก็บข้อมูลที่สงวนไว้สำหรับการประมวลผลงานหรือกลุ่มของงาน บนระบบของคุณ หน่วยเก็บข้อมูลหลักสามารถจัดสรรแบบโลจิคัลที่เรียกว่าพูลหน่วยความจำได้โดยค่าเริ่มต้น คือ ระบบจะจัดการถ่ายโอนข้อมูลและโปรแกรมเข้าในพูลหน่วยความจำ.

พูลหน่วยความจำที่งานของผู้ใช้ได้หน่วยความจำจะเป็นพูลอันเดียวกันกับที่จำกัด activity level ของพูลนั้นเสมอ (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 หากขนาดของพูลในระบบย่อยยิ่งใหญ่มากขึ้น จำนวนงานที่สามารถทำในระบบย่อยนั้น ยิ่งมากขึ้น

การใช้พูลหน่วยความจำแบบแบ่งใช้ช่วยให้ระบบสามารถกระจายงานสำหรับผู้ใช้งานแบบโต้ตอบไปยังระบบย่อยหลายระบบ ในขณะที่ยังคงอนุญาตให้งานรันในพูลหน่วยความจำเดียวกันได้

พูลหลายพูลในระบบย่อย ช่วยคุณควบคุมการแย่งชิงรีซอร์สของระบบของงาน ประโยชน์ของการมีพูลหลายพูลในระบบย่อยคือ คุณสามารถแบ่งปริมาณของงานที่ทำและเวลาตอบสนองสำหรับงานเหล่านั้นได้ ตัวอย่างเช่น ในระหว่างวัน คุณอาจต้องการให้งานแบบโต้ตอบรันโดยมีเวลาตอบสนองที่ดี สำหรับประสิทธิภาพที่ดีขึ้น คุณสามารถทำให้พูลแบบโต้ตอบใหญ่ขึ้นในตอนกลางวัน คุณอาจรันงานแบ็คชหลายงาน คุณจึงทำให้พูลแบ็คชใหญ่ขึ้น

หมายเหตุ: การปรับ และการจัดการระบบของคุณสามารถช่วยเพิ่มประสิทธิภาพความต่อเนื่องของงานผ่านระบบของคุณ แต่การปรับไม่สามารถช่วยแก้ปัญหาการขาดแคลนฮาร์ดแวร์รีซอร์สได้ ดังนั้นควรพิจารณาเรื่องการอัปเกรดฮาร์ดแวร์ หากความต้องการด้านเวิร์กโหลดมีความสำคัญมาก

วิธีการจัดการข้อมูลในพูลหน่วยความจำ

ถ้าข้อมูลมีอยู่แล้วใน แหล่งเก็บข้อมูลหลัก ข้อมูลสามารถถูกอ้างอิงแยกต่างหากจากพูลหน่วยความจำที่เก็บอยู่ได้ อย่างไรก็ตาม ถ้าข้อมูลที่ต้องการไม่มีอยู่ในพูลหน่วยความจำใดๆ ข้อมูล จะถูกนำกลับมาอยู่ในพูลหน่วยความจำเดียวกันกับงานที่อ้างอิงถึงข้อมูล (สิ่งนี้ เรียกว่า page fault) เมื่อข้อมูลถูกถ่ายโอนเข้าในพูลหน่วยความจำ ข้อมูลอื่นจะถูก สับเปลี่ยน และถ้าเปลี่ยนแปลง จะถูกบันทึกอย่างอัตโนมัติในหน่วยความจำรอง (สิ่งนี้เรียกว่าการเพจ) ขนาดของพูลหน่วยความจำควรใหญ่พอที่จะเก็บ การถ่ายโอนข้อมูล (การเพจ) ที่ระดับที่เหมาะสม เนื่องจากอัตราามีผลต่อประสิทธิภาพ

หลักการที่เกี่ยวข้อง

“การจัดการพูลหน่วยความจำ” ในหน้า 199

การตรวจสอบให้แน่ใจว่างานมีหน่วยความจำเพียงพอที่จะดำเนินงานอย่าง มีประสิทธิภาพถือเป็นสิ่งที่สำคัญ ถ้ามีการให้หน่วยความจำกับระบบย่อย A มากเกินไปและไม่เพียงพอที่ระบบย่อย B งานในระบบย่อย B อาจเริ่มรันได้ไม่ดี ข้อมูลต่อไปนี้อธิบายงานหลากหลายที่เกี่ยวข้องกับการจัดการ พูลหน่วยความจำ

ข้อมูลที่เกี่ยวข้อง

Retrieve System Status (QWCRSSTS) API

การจัดการประสิทธิภาพระบบ

การปรับประสิทธิภาพระดับต้น

แอ็พพลิเคชันสำหรับการจัดการประสิทธิภาพ

รายงานจากประสบการณ์: Performance Adjuster (QPFRADJ)

ชนิดของพูลหน่วยความจำ

บนระบบของคุณ หน่วยเก็บข้อมูลหลักทั้งหมดสามารถจัดสรรแบบโลจิคัลที่เรียกว่า พูลหน่วยความจำ. หน่วยความจำทั้งหมดในระบบสามารถแบ่งเป็นแบบส่วนตัวหรือแบบแบ่งใช้ได้อย่างใดอย่างหนึ่ง มีพูลหน่วยความจำส่วนตัว, พูลหน่วยความจำแบบแบ่งใช้ และพูลหน่วยความจำแบบแบ่งใช้พิเศษ พูลหน่วย ความจำสูงสุดทั้งหมด 64 พูล สามารถแอ็คทีฟได้ในเวลาเดียวกัน ไม่ว่าจะเป็นแบบ ส่วนตัวหรือแบบแบ่งใช้ก็ตาม

พูลหน่วยความจำส่วนตัว

พูลหน่วยความจำส่วนตัว (เรียกอีกอย่างว่าพูลหน่วยความจำที่ผู้ใช้กำหนด) มีจำนวนของแหล่งเก็บข้อมูลหลักที่ระบบย่อยเดี่ยวสามารถใช้ในการรันงานต่างๆ พูลเหล่านี้ไม่สามารถแบ่งใช้โดยหลายระบบย่อย พูลชนิดนี้ถูกระบุใน System i Navigator ตามชื่อระบบย่อย คุณสามารถมีพูลหน่วยความจำส่วนตัวได้มากถึง 62 พูลสำหรับใช้ในระบบย่อยที่แอดคิฟ

พูลหน่วยความจำแบบแบ่งใช้

พูลแบบแบ่งใช้แบ่งเป็นแบบพิเศษหรือทั่วไป อย่างไรก็ตามอย่างหนึ่ง; Machine pool และ Base pool ถือเป็นพูลแบบแบ่งใช้พิเศษ, และพูลแบบแบ่งใช้อื่นทั้งหมดถือเป็นพูลแบบแบ่งใช้ทั่วไป คุณสามารถระบุพูลหน่วยความจำแบบแบ่งใช้ได้ 63 จากทั้งหมด 64 พูลที่ถูกกำหนด บนระบบ สำหรับใช้เมื่อสร้าง subsystem descriptions (machine pool ถูกสงวนไว้สำหรับการใช้งานของระบบ)

Special Shared Pools (*MACHINE และ *BASE)

*MACHINE

พูลหน่วยความจำ Machine ใช้สำหรับโปรแกรม highly-shared Machine และระบบปฏิบัติการ พูลนี้ระบุเป็น Machine ใน System i Navigator. พูลหน่วยความจำ Machine นำเสนอหน่วยความจำสำหรับภารกิจที่ระบบต้องรัน โดยไม่ต้องมีการจัดการของคุณ ขนาดของพูลหน่วยความจำนี้ระบุอยู่ในค่ากำหนดของระบบ 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 System i Navigator มีแหล่งเก็บข้อมูลหลักที่ไม่ได้จัดสรรทั้งหมดบนระบบ (แหล่งเก็บข้อมูลหลักทั้งหมดที่พูลหน่วยความจำอื่นไม่ต้องการ) 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)

พูลแบบแบ่งใช้ทั่วไป

พูลแบบแบ่งใช้ทั่วไปคือ พูลของแหล่งเก็บข้อมูลหลักที่หลายระบบย่อยสามารถใช้ได้พร้อมกัน บน อินเทอร์เน็ตแบบอักษระ พูลถูกระบุดังนี้:

- *INTERACT คือพูลของหน่วยความจำแบบโต้ตอบที่ใช้สำหรับงานแบบโต้ตอบ
- *SPOOL คือพูลของหน่วยความจำที่ใช้สำหรับ spool writer
- *SHRPOOL1 ถึง *SHRPOOL60 คือพูลของหน่วยความจำที่คุณสามารถใช้ สำหรับการใช้งานของคุณเอง

ใน System i Navigator พูลแบบแบ่งใช้ทั่วไปถูกระบุเป็น Interactive, Spool และ Shared 1 – Shared 60

งานที่เกี่ยวข้อง

“การสร้างพูลหน่วยความจำส่วนตัว” ในหน้า 206

พูลหน่วยความจำส่วนตัว (เรียกอีกอย่างว่าพูลหน่วยความจำที่ผู้ใช้กำหนด) สามารถใช้โดยระบบย่อยที่ได้มาจาก IBM หรือโดยระบบย่อยที่ผู้ใช้กำหนด คุณสามารถกำหนด definition พูลหน่วยความจำได้สูงสุดถึง 10 definition สำหรับระบบย่อยหนึ่ง คุณสามารถสร้างพูลหน่วยความจำส่วนตัวใน subsystem description

ข้อมูลที่เกี่ยวข้อง

การจัดการประสิทธิภาพระบบ

การปรับประสิทธิภาพระดับต้น

แอ็พพลิเคชันสำหรับการจัดการประสิทธิภาพ

รายงานจากประสบการณ์: Performance Adjuster (QPFRADJ)

ค่ากำหนดของระบบประสิทธิภาพ: ขนาดพูลหน่วยความจำเครื่อง

ค่ากำหนดของระบบประสิทธิภาพ: ขนาดต่ำสุดของพูลหน่วยความจำหลัก

ค่ากำหนดของระบบประสิทธิภาพ: thread ที่มีลิมิตสูงสุดของพูลหน่วยความจำหลัก

แบบแผนการกำหนดหมายเลขพูล

พูลมีแบบแผนการกำหนดหมายเลขอยู่สองชุด: ชุดหนึ่งใช้ภายใน ระบบย่อยและอีกชุดหนึ่งใช้ในระบบทั้งหมด ระบบย่อยจะใช้ชุดของหมายเลขที่ อ้างอิงถึงพูลที่ระบบย่อยใช้ ดังนั้น เมื่อคุณสร้างหรือเปลี่ยน subsystem description คุณสามารถกำหนดหนึ่งหรือหลายพูลและกำหนดหมายเลขพูลเป็น 1, 2, 3 และต่อไป นี้คือ การกำหนดพูลระบบย่อย และไม่สอดคล้องกับหมายเลขพูลที่แสดงบนจอแสดงผล Work with System Status (WRKSYSSTS)

มีการใช้ชุดของหมายเลขที่แตกต่างอื่นในการเก็บรักษาแตรีกของพูลทั้งหมดบนระบบ จอแสดงผล Work with Subsystems (WRKSBS) เกี่ยวข้องกับ subsystem pool identifier และหัวข้อคอลัมน์ที่ system pool identifier

```
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

Bottom

Parameters or command
==>
F3=Exit F5=Refresh F11=Display system data F12=Cancel
F14=Work with system status
```

ตัวอย่าง: พูลถูกกำหนดหมายเลขอย่างไร

ตัวอย่างต่อไปนี้ สาธิตวิธีการกำหนดหมายเลขพูล

ระบบย่อย		
CRTSBSD QINTER	CRTSBSD NYSBS	CRTSBSD PASBS
Pools (1 *BASE)	Pools (1 *BASE)	Pools (1 *BASE)
(2 1200 25)	(2 500 3)	(2 1000 3)

ระบบย่อย		
	(3 *SHRPOOL2)	(3 *SHRPOOL2)
(พูลระบบ 2, 3)	(พูลระบบ 2, 4, 5)	(พูลระบบ 2, 5, 6)

หลังจากที่ QINTER เริ่มต้น พูลต่อไปนี้จะถูกจัดสรร:

หมายเลขพูลระบบ	คำอธิบาย	QINTER
1	*Machine pool	
2	*BASE pool	1
3	QINTER private pool	2

หลังจากที่ NYSBS เริ่มต้น พูลต่อไปนี้จะถูกจัดสรร:

หมายเลขพูลระบบ	คำอธิบาย	QINTER	NYSBS
1	*MACHINE pool		
2	*BASE pool	1	1
3	QINTER private pool	2	
4	NYSBS private pool		2
5	*SHRPOOL2 shared pool		3

หลังจากที่ PASBS เริ่มต้น พูลต่อไปนี้จะถูกจัดสรร:

หมายเลขพูลระบบ	คำอธิบาย	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

งานที่เกี่ยวข้อง

“การจัดการพารามิเตอร์การปรับสำหรับแชร์พูล” ในหน้า 203

ถ้าต้องการจัดการพารามิเตอร์สำหรับแชร์พูล ให้ใช้ System i Navigator หรือคำสั่งอินเตอร์เฟซแบบอักขระ

“การจัดการ configuration ของพูล” ในหน้า 204

ถ้าต้องการเปลี่ยนขนาดของพูล, activity level หรืออ็อปชันการเพจ ให้ใช้ System i Navigator หรือคำสั่งอินเตอร์เฟซแบบอักษร

“การเปลี่ยนขนาดของพูลหน่วยความจำ” ในหน้า 204

ขนาดของพูลหน่วยความจำมีผลโดยตรงกับปริมาณของงานที่ระบบย่อยสามารถประมวลผลได้ ระบบย่อยยังมีหน่วยความจำมาก, ยิ่งสามารถทำงานให้เสร็จ สมบูรณ์ได้มากขึ้น นับเป็นเรื่องสำคัญที่คุณควรมอนิเตอร์ระบบของคุณอย่างรอบคอบ ก่อนจะเริ่มการเปลี่ยนพารามิเตอร์ของพูลหน่วยความจำของคุณ คุณยังต้องตรวจสอบชำระระดับเหล่านี้เป็นระยะ เนื่องจากอาจต้องมีการปรับบางอย่างใหม่

ข้อมูลที่เกี่ยวข้อง

การจัดการประสิทธิภาพระบบ

การปรับประสิทธิภาพระดับต้น

แอ็พพลิเคชันสำหรับการจัดการประสิทธิภาพ

รายงานจากประสบการณ์: Performance Adjuster (QPFRADJ)

การจัดสรรพูลหน่วยความจำ

เมื่อคุณเริ่มระบบย่อย ระบบจะพยายามจัดสรรพูลของหน่วยความจำ ที่ผู้ใช้กำหนด ซึ่งถูกกำหนดใน subsystem description ของระบบย่อยที่เริ่มต้น

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

Pool ID Specified in SBSDD	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

งานที่เกี่ยวข้อง

“การดูข้อมูลพูลหน่วยความจำ” ในหน้า 200

คุณสามารถดูข้อมูลเกี่ยวกับพูลหน่วยความจำที่มีอยู่บนระบบของคุณได้โดยใช้ System i Navigator หรืออินเตอร์เฟซแบบอักษร

“การกำหนดจำนวนของระบบย่อยที่ใช้พูลหน่วยความจำ” ในหน้า 201

ระบบย่อยจะได้รับการจัดสรรหน่วยความจำจำนวนหนึ่งเพื่อใช้รันงาน เป็นสิ่งสำคัญที่ต้องทราบจำนวนของระบบย่อยที่

กำลังดึงหน่วยความจำออกมาจาก พูลหน่วยความจำเดียวกัน หลังจากที่คุณทราบจำนวนของระบบย่อยที่กำลังส่งงาน ไปที่ พูลและจำนวนของงานที่กำลังรันในพูล คุณอาจต้องการลด contention รีซอร์สโดยการปรับเปลี่ยนขนาดและ activity level ของพูล

“การกำหนดจำนวนของงานในพูลหน่วยความจำ” ในหน้า 201

System i Navigator นำเสนอวิธีการแสดงรายการของงานซึ่งอยู่ในพูลหน่วยความจำอย่างรวดเร็ว

“การกำหนดพูลที่งานเดี่ยวกำลังรันอยู่” ในหน้า 202

ถ้าคุณมีงานที่ไม่ได้ปฏิบัติเนื่องจากคุณคาดว่าคุณอาจต้อง ตรวจสอบพูลหน่วยความจำซึ่งงานกำลังรันอยู่ในการกำหนด พูลที่งานเดี่ยวกำลังรันอยู่ให้ใช้ System i Navigator หรืออินเทอร์เน็ตเฟสแบบอักษร

ข้อมูลที่เกี่ยวข้อง

การจัดการประสิทธิภาพระบบ

การปรับประสิทธิภาพระดับต้น

แอ็พพลิเคชันสำหรับการจัดการประสิทธิภาพ

รายงานจากประสบการณ์: Performance Adjuster (QPFRADJ)

activity level ของพูลหน่วยความจำ

activity level ของพูลหน่วยความจำ คือ จำนวนของ thread ที่สามารถใช้ CPU ในสถานะแอ็คทีฟได้ในเวลาเดียวกันในพูลหน่วย ความจำ สิ่งนี้ช่วยให้สามารถใช้รีซอร์สของระบบได้อย่างมีประสิทธิภาพ ระบบจัดการการควบคุมของ activity level

บ่อยครั้งในระหว่างการประมวลผล ใน thread โปรแกรมจะรอรีซอร์สระบบหรือการตอบกลับจากผู้ใช้งาน เวิร์กสเตชัน ใน ระหว่างการรอนั้น thread จะยกเลิกการใช้งานของ activity level ของพูลหน่วยความจำเพื่อทำให้ thread อื่นที่พร้อมจะประมวล ผลได้สามารถเข้ามาทำงานแทนที่ไปก่อน

เมื่อมี threads ถูกเริ่มมากกว่า thread ที่สามารถรันได้พร้อมกัน threads ที่มากเกินไปต้องรอที่จะใช้หน่วยการประมวลผล (โดย ปกติการรอนี้ใช้เวลาไม่นานนัก) activity level ของพูลหน่วยความจำจะทำให้คุณจำกัดระดับการแย่งใช้ หน่วยความจำหลักในพูล หน่วยความจำทั้งหลายในระบบย่อยของคุณ

จำนวน thread ที่กำลังรันอยู่ (หรือ active thread) คือ จำนวน thread ที่มีสิทธิในการแย่งใช้โพรเซสเซอร์ และมีส่วนใน activity level ของพูลหน่วยความจำ ใน ลักษณะเช่นนี้เอง active thread จะไม่รวมถึง thread ที่กำลังรออินพุต, รอข้อความ, รอได้รับการ จัดสรรอุปกรณ์เพื่อใช้งาน หรือรอไฟล์เพื่อเปิดใช้ active thread จะไม่รวมถึง thread ที่ไม่มีสิทธิ (คือ thread ที่พร้อมจะรัน แต่ activity level ของพูลหน่วยความจำได้ขึ้นไปถึงจุดสูงสุดแล้ว)

activity level ทำงานอย่างไร

thread มากกว่าหนึ่งอันจะแอ็คทีฟได้ในเวลาเดียวกันในพูลหน่วยความจำ เนื่องจากการประมวลผล thread อาจถูกอิน เตอร์รัปต์เพียงชั่วขณะเท่านั้น ในขณะที่ข้อมูลที่ต้องการกำลังถูกเรียกออกมาจากหน่วยความจำสำรอง ในระหว่างช่วงหน่วงนี้ ซึ่งโดยปกติจะใช้เวลาไม่นาน thread อื่นสามารถรันได้ ด้วยการใช้อัตรา activity level เครื่องสามารถประมวลผล thread ในพูลหน่วย ความจำได้เป็นจำนวนมากในขณะที่เดียวกับที่ยังสามารถรักษาระดับของ contention ตามข้อจำกัด ที่คุณระบุ

activity level สูงสุด

หลังจากใช้งานถึง activity level สูงสุดสำหรับพูลหน่วยความจำแล้ว threads เพิ่มเติมที่ต้องการพูลหน่วยความจำจะถู กวางในสภาพที่ไม่มีสิทธิ เพื่อรอให้จำนวนของ active threads ในพูลหน่วยความจำลดลง ต่ำกว่า activity level สูงสุด หรือรอให้ thread ใช้เวลาของการแบ่งเวลาจนหมด เมื่อใดก็ตามที่ thread ไม่ได้ใช้พูลหน่วยความจำแล้ว thread อื่นๆ

ที่ยังไม่แอ็คทีฟก็จะมีสิทธิที่จะรันตามระดับความสำคัญของตัวมัน ตัวอย่างเช่น หาก thread ที่รันอยู่กำลัง รอการตอบกลับมาจากเวิร์กสเตชัน thread นั้นก็จะปล่อย activity level ทิ้งไปทำให้ activity level ลดลงมาไม่ถึงระดับสูงสุด

การกำหนด activity level ของพูลหน่วยความจำ

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

งานที่เกี่ยวข้อง

“การดูข้อมูลพูลหน่วยความจำ” ในหน้า 200

คุณสามารถดูข้อมูลเกี่ยวกับพูลหน่วยความจำที่มีอยู่บนระบบของคุณได้โดยใช้ System i Navigator หรืออินเตอร์เฟซแบบอักษร

“การกำหนดจำนวนของระบบย่อยที่ใช้พูลหน่วยความจำ” ในหน้า 201

ระบบย่อยจะได้รับการจัดสรรหน่วยความจำจำนวนหนึ่งเพื่อใช้รันงาน เป็นสิ่งสำคัญที่ต้องทราบจำนวนของระบบย่อยที่กำลังดึงหน่วยความจำออกมาจาก พูลหน่วยความจำเดียวกัน หลังจากที่ทราบจำนวนของระบบย่อยที่กำลังส่งงาน ไปที่พูลและจำนวนของงานที่กำลังรันในพูล คุณอาจต้องการลด contention รีซอร์สโดยการปรับเปลี่ยนขนาดและ activity level ของพูล

“การกำหนดจำนวนของงานในพูลหน่วยความจำ” ในหน้า 201

System i Navigator นำเสนอวิธีการแสดงรายการของงานซึ่งอยู่ในพูลหน่วยความจำอย่างรวดเร็ว

“การกำหนดพูลที่งานเดี่ยวกำลังรันอยู่” ในหน้า 202

ถ้าคุณมีงานที่ไม่ได้ปฏิบัติเนื่องจากคุณคาดว่าอาจต้อง ตรวจสอบพูลหน่วยความจำซึ่งงานกำลังรันอยู่ในการกำหนดพูลที่งานเดี่ยวกำลังรันอยู่ ให้ใช้ System i Navigator หรืออินเตอร์เฟซแบบอักษร

ข้อมูลที่เกี่ยวข้อง

การจัดการประสิทธิภาพระบบ

การปรับประสิทธิภาพระดับต้น

แอ็พพลิเคชันสำหรับการจัดการประสิทธิภาพ

รายงานจากประสบการณ์: Performance Adjuster (QPFRADJ)

งาน

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

งานแอ็คทีฟแต่ละงานประกอบด้วย thread (initial thread) อย่างน้อยหนึ่ง thread และอาจประกอบด้วย thread สำรองเพิ่มเติมด้วย thread เป็นหน่วยย่อยอิสระของชุดงาน แอ็ตทริบิวต์ของงานถูกแบ่งใช้ระหว่าง thread ต่างๆ ของงาน อย่างไรก็ตาม, thread ยังมีชุดแอ็ตทริบิวต์ของตนเองต่างหาก, เช่น call stack แอ็ตทริบิวต์ของงานประกอบด้วยข้อมูลเกี่ยวกับวิธีที่งานจะถูกประมวลผล งานจะทำหน้าที่เป็นเจ้าของแอ็ตทริบิวต์ที่แบ่งใช้กันระหว่าง thread ภายในงานนั้น การจัดการระบบงานจะจัดเตรียมวิธีควบคุมให้งานดำเนิน ไปโดยสำเร็จสมบูรณ์บนระบบผ่านทางแอ็ตทริบิวต์ของงาน

สิทธิในการใช้งานที่เหมาะสม

ในการเปลี่ยนแปลงแอ็ตทริบิวต์ของงานส่วนใหญ่ คุณต้องมีสิทธิพิเศษในการควบคุมงาน (*JOBCTL) หรือโปรไฟล์ผู้ใช้ของคุณที่ต้องตรงกับ job user identity ของงานที่จะเปลี่ยน

มีสองสามแอตทริบิวต์ที่จำเป็นต้องใช้สิทธิพิเศษ *JOBCTL ในการเปลี่ยนแปลง แอตทริบิวต์เหล่านี้คือ:

- ช่วงเวลารอติฟอลด์
- ระดับความสำคัญของการรัน
- การแบ่งเวลา

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

สำหรับแอตทริบิวต์ของงานใดๆที่อ้างอิงถึงอ็อบเจกต์ i5/OS เช่น คิวงาน, เอาต์พุตคิว และตารางลำดับ คุณต้องมีสิทธิที่ถูกตั้งที่อ็อบเจกต์ สำหรับรายละเอียดเพิ่มเติมเกี่ยวกับ i5/OS สิทธิ ให้ดูที่ สิทธิที่จำเป็นสำหรับอ็อบเจกต์ที่ใช้โดยคำสั่งในกลุ่มหัวข้อ Security reference

หลักการที่เกี่ยวข้อง

“Job user identity” ในหน้า 37

job user identity (JUID) คือชื่อของโปรไฟล์ผู้ใช้ซึ่งทำให้งานนี้เป็นที่รู้จักของงานอื่น ชื่อนี้ใช้สำหรับการตรวจสอบการให้สิทธิ เมื่องานอื่นพยายามดำเนินการกับงานนี้

ลักษณะงาน

การจัดการระบบงานจัดเตรียมวิธีการควบคุมให้งานดำเนิน ไปโดยสำเร็จสมบูรณ์บนระบบผ่านทางแอตทริบิวต์ของงานอย่างไรก็ดี ก่อนที่คุณจะสามารถควบคุมลักษณะหลากหลายของงานได้ คุณต้องเข้าใจลักษณะต่างๆ ของงาน

ข้อมูลต่อไปนี้อธิบายลักษณะของงาน:

ไวยากรณ์ชื่องาน:

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

- สำหรับงานแบบโต้ตอบ ชื่องานคือชื่อเดียวกันกับชื่อของ เวิร์กสเตชันหรืออีเมลเตอร์เซสชันที่คุณได้ signed on สำหรับงานแบ็ตซ์ คุณสามารถระบุชื่องานของคุณเองได้ ชื่องานสามารถยาวได้ถึง 10 อักขระ
- ชื่อผู้ใช้คือชื่อของโปรไฟล์ผู้ใช้ซึ่งงานถูกเริ่มอยู่ภายใต้ สำหรับงานแบบโต้ตอบ ชื่อผู้ใช้ คือ โปรไฟล์ผู้ใช้ที่ใช้ในการ sign on ที่ระบบ นี้คือชื่อผู้ใช้ที่คุณป้อนในฟิลด์ผู้ใช้บนจอแสดงผล sign-on ถ้าคุณใช้ Telnet และข้ามการ sign-on ชื่อนี้ คือ ชื่อผู้ใช้ที่คุณใช้ sign on โดยอัตโนมัติที่ระบบ สำหรับงานแบ็ตซ์ คุณสามารถระบุโปรไฟล์ผู้ใช้ซึ่งงานแบ็ตซ์จะถูกรันอยู่ภายใต้ได้ ชื่อผู้ใช้สามารถยาวได้ถึง 10 อักขระ
- หมายเลขงานคือหมายเลขเฉพาะที่กำหนดโดยระบบ เพื่อให้คุณสามารถระบุถึง งานได้ แม้ว่ามีงานที่มีชื่องานและชื่อผู้ใช้เหมือนกันอยู่มากกว่าหนึ่งงาน หมายเลขงานคือตัวเลข 6 หลักเสมอ

ไวยากรณ์

ไวยากรณ์สำหรับชื่องานที่ถูกต้องเหมือนกับไวยากรณ์ ของชื่อที่ถูกต้องสำหรับอ็อบเจกต์ ตัวอย่างเช่น ถ้าชื่องาน คือ DSP01 ผู้ใช้คือ QPGMR และหมายเลขงาน คือ 000578 ชื่องานที่ถูกต้องที่ป้อนบนคำสั่ง Work with Job (WRKJOB) คือ:

WRKJOB JOB(000578/QPGMR/DSP01)

อีกอย่างหนึ่ง ที่คล้ายกับชื่ออ็อบเจกต์คือ คุณไม่ต้องระบุ qualifier ทั้งหมด เช่น ลองพิจารณาดังนี้:

หรือ

WRKJOB JOB(DSP01)

สิ่งนี้ให้ผลเหมือนกับการป้อนชื่องานที่ต้องทั้งหมด ถ้างานหลายงาน บนระบบมีส่วนที่ตรงกับชื่องานที่คุณป้อน จอแสดงผล Select Job จะปรากฏขึ้น จอแสดงผลนี้อนุญาตให้คุณเลือกงานที่คุณ ต้องการจากรายการของชื่องานที่เข้าได้

แอ็ททริบิวต์งาน:

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

แอ็ททริบิวต์การควบคุมงานช่วยให้คุณเลือกที่จะควบคุมงานที่ระดับงาน, ระดับผู้ใช้ หรือระดับระบบ ตัวอย่างเช่น คุณสามารถให้ระบบของคุณ ตั้งค่ากำหนดของระบบสำหรับแอ็ททริบิวต์งาน (ซึ่งคือค่าดีฟอลต์ของระบบ) จากนั้นถ้าคุณ ต้องการเปลี่ยนค่าสำหรับ งานใหม่ทั้งหมดบนระบบ คุณสามารถจะเปลี่ยนค่ากำหนดของระบบได้

โดยการระบุค่าในรายละเอียดของงาน คุณสามารถเปลี่ยนชนิดของงานทั้งหมดที่ใช้รายละเอียดของงานนั้นได้ ตัวอย่างเช่น ถ้างานแบ็ตช์ทั้งหมดของคุณ ใช้รายละเอียดของงานเดียวกัน การเปลี่ยนรายละเอียดของงานสำหรับงานแบ็ตช์ สามารถส่งผลกระทบต่องานแบ็ตช์ทั้งหมดของคุณและทำให้ไม่มีผลต่องานอื่นทั้งหมดได้

ข้อมูลที่เกี่ยวข้อง

รายงานจากประสบการณ์: แอ็ททริบิวต์งานการจัดการระบบงาน

รายละเอียดของงาน:

รายละเอียดของงานอนุญาตให้คุณสร้างชุดของแอ็ททริบิวต์งาน ที่จะถูกบันทึกและพร้อมใช้งานสำหรับการใช้ในลักษณะต่างๆ รายละเอียดของงานสามารถใช้เป็น ซอร์สสำหรับแอ็ททริบิวต์งานบางอย่างที่บอกให้ระบบทราบถึงวิธีการ รันงาน แอ็ททริบิวต์บอกให้ระบบทราบว่า จะเริ่มงานเมื่อไร เรียกงาน มาจากที่ไหน และจะรันงานอย่างไร คุณสามารถคิดว่ารายละเอียดของงาน เป็นเท็มเพลตที่งานหลายงานสามารถใช้ได้ ดังนั้นจึงลดจำนวนของพารามิเตอร์ เฉพาะที่คุณต้องเซ็ทสำหรับงานแต่ละงาน

รายละเอียดของงานถูกใช้โดยงานแบบ autostart, แบบแบ็ตช์, แบบโต้ตอบ และ งานแบบ prestart คุณสามารถใช้รายละเอียดของงาน เดียวกันสำหรับงานหลายๆ งานได้ เมื่อคุณกำหนดงาน คุณสามารถใช้รายละเอียดของงานในหนึ่งจากสองวิธีดังนี้:

- ใช้รายละเอียดของงานที่ระบุโดยไม่แทนที่ค่าเดิมของแอ็ททริบิวต์. ตัวอย่างเช่น:

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

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

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

หมายเหตุ: คุณไม่สามารถแทนที่ค่าเดิมแอ็ททริบิวต์รายละเอียดของงานใดๆ สำหรับงานแบบ autostart, งานเวิร์กสเตชัน, หรืองานการสื่อสาร

งานที่เกี่ยวข้อง

“การสร้างรายละเอียดของงาน” ในหน้า 129

คุณสามารถใช้อินเตอร์เฟซแบบอักษร คำสั่ง Work With Job Description (WRKJOB) หรือคำสั่ง Create Job Description (CRTJOB) เพื่อเปลี่ยนรายละเอียดของงาน

“การใช้รายละเอียดของงาน” ในหน้า 129

วิธีทั่วไปในการใช้รายละเอียดของงานคือ โดยการระบุรายละเอียดของงาน ในคำสั่ง Submit Job (SBMJOB) พารามิเตอร์ job description (JOB) คือ ที่ที่คุณจะระบุรายละเอียดของงานที่คุณต้องการให้งานนี้ใช้ เมื่อคุณกำหนดงานแบ็ตซ์ คุณสามารถใช้รายละเอียดของงานในหนึ่งจากสองวิธีดังนี้:

รายละเอียดของงานและการรักษาความปลอดภัย:

งานทุกงานในระบบใช้รายละเอียดของงานในระหว่างการเริ่มงาน สิ่งนี้ควบคุมแอตทริบิวต์ต่างๆ ของงาน พารามิเตอร์ USER ควบคุมชื่อของโปรไฟล์ผู้ใช้ที่กำหนดที่งาน รายละเอียดของงานที่มีชื่อ โปรไฟล์ผู้ใช้ (USER) ที่ระบุควรถูกอนุญาตเฉพาะสำหรับผู้ใช้ที่ระบุเท่านั้น ถ้าไม่ ที่ระดับความปลอดภัย 30 และต่ำกว่า ผู้ใช้อื่นจะสามารถส่ง งานเพื่อรันภายใต้โปรไฟล์ผู้ใช้นั้นได้

ตัวอย่างเช่น ลองพิจารณา

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

ตัวอย่างนี้ มีความเสี่ยงในการรักษาความปลอดภัยเนื่องจากผู้ใช้ใดๆ สามารถส่งงานโดยใช้รายละเอียดของงาน XX ได้ และได้รับอนุญาตในสิ่งที่ JONES ได้รับอนุญาตให้ทำ ถ้า มีการใช้รายละเอียดของงานชนิดนี้บน workstation entry รายละเอียดของงาน จะอนุญาตให้ทุกคนสามารถ sign on เป็นผู้ใช้ได้โดยเพียงแต่กดปุ่ม Enter เพื่อหลีกเลี่ยงการรั่วซึม การรักษาความปลอดภัยอย่างอนุญาตให้รายละเอียดของงานชนิดนี้เป็น *PUBLIC

หมายเหตุ: ที่ระดับความปลอดภัย 40 และ 50 คำสั่ง Submit Job (SBMJOB) กำหนดว่าผู้ส่งงานต้องได้รับอนุญาต (*USE) ในการใช้โปรไฟล์ผู้ใช้ที่ระบุชื่อในรายละเอียดของงาน สิ่งนี้สมมติว่า SBJOB ระบุผู้ใช้ (*JOB) ถึงกระนั้นก็ตาม ให้หลีกเลี่ยงการระบุผู้ใช้ในรายละเอียดของงาน ยกเว้นว่ามีความจำเป็นเนื่องจากเหตุผลเฉพาะบางอย่าง (เช่น งานแบบ autostart) และ คุณควบคุมการเข้าใช้งานอย่างแน่นหนา

พารามิเตอร์ USER และงานแบบ interactive

รายละเอียดของงานที่จะใช้ถูกกำหนดบนคำสั่ง Add Work Station Entry (ADDWSE) ค่าดีฟอลต์คือการใช้รายละเอียดของงานในโปรไฟล์ผู้ใช้ ถ้า USER(*RQD) ถูกระบุในรายละเอียดของงาน ผู้ใช้ต้องป้อนชื่อผู้ใช้ ถ้ามีการระบุ USER(XXXX) (โดยที่ XXXX คือ ชื่อโปรไฟล์ผู้ใช้เฉพาะ) ผู้ใช้จะได้รับอนุญาตให้กดปุ่ม Enter บนจอแสดงผล sign-on และดำเนินการภายใต้ชื่อโปรไฟล์ผู้ใช้ XXXX ยกเว้นว่าระดับความปลอดภัยจะเป็น 40 หรือสูงกว่า

พารามิเตอร์ USER และงานแบ็ตซ์

รายละเอียดของงานที่ใช้สำหรับงานแบ็ตซ์ถูกระบุบนคำสั่ง Submit Job (SBMJOB) หรือ Batch Job (BCHJOB)

ถ้ามีการป้อน input stream ที่มีคำสั่ง BCHJOB ผู้ใช้ที่ป้อนหนึ่งในคำสั่ง Start Reader (STRDBRDR, STRDKTRDR) หรือหนึ่งในคำสั่ง Submit Job (SBMJOB, SBMDKTJOB และอื่นๆ) ต้องมีสิทธิการดำเนินการอ็อบเจกต์ (*OBJOPR) ที่รายละเอียดของงาน ที่ถูกระบุ เมื่อใช้ input stream งานจะดำเนินการภายใต้โปรไฟล์ผู้ใช้ของรายละเอียดของงานเสมอ ไม่ใช่ของผู้ใช้ที่ทำงาน บนคิวงาน ถ้ามีการระบุ USER(*RQD) ในรายละเอียดของงาน จะไม่สามารถใช้รายละเอียดของงานบนคำสั่ง BCHJOB

ถ้าใช้คำสั่ง SBMJOB คำสั่งจะใช้ดีฟอลต์เพื่อให้งานแบ็คชดดำเนินการภายใต้ชื่อโปรไฟล์ผู้ใช้ของผู้ส่งงาน อย่างไรก็ตาม ถ้ามีการระบุ USER(*JOB) บนคำสั่ง SBMJOB งานจะดำเนินการภายใต้ชื่อที่ระบุในพารามิเตอร์ USER ของรายละเอียดของงาน

บ่อยครั้งที่ต้องใช้ชื่อที่ระบุในรายละเอียดของงานเพื่ออนุญาตให้ผู้ใช้ส่งงาน สำหรับโปรไฟล์ผู้ใช้ที่ระบุ ตัวอย่างเช่น รายละเอียดของงาน QBATCH ถูกจัดส่งมา พร้อมด้วย USER(QPGMR) เพื่อให้อนุญาตสิ่งนี้ เพื่อหลีกเลี่ยงความเสี่ยง การรักษาความปลอดภัย อย่านุญาตให้รายละเอียดของงานชนิดนี้เป็น *PUBLIC

Call stacks:

call stack คือรายการที่จัดลำดับของโปรแกรมหรือโพรซีเจอร์ทั้งหมดที่รันอยู่ในปัจจุบันสำหรับงาน โปรแกรมและโพรซีเจอร์สามารถ เริ่มต้นได้โดยตรงด้วยคำสั่ง CALL หรือโดยทางอ้อมจากเหตุการณ์อื่น

call stack พร้อมใช้งานทั้งระดับงานและระดับ thread บนอินเทอร์เฟซแบบอักขระ call stack คือรายการ last-in-first-out (LIFO) ของ call stack entries หนึ่ง entry สำหรับแต่ละโพรซีเจอร์หรือโปรแกรมที่เรียก ใน System i Navigator โดยค่าดีฟอลต์ entry หลังสุดใน stack จะปรากฏอยู่บนสุดของรายการ 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 ถ้าคุณกำลังรันภายใต้โปรไฟล์ผู้ใช้ที่มีสิทธิพิเศษ *SERVICE คุณสามารถดู entries เพิ่มเติมสำหรับ licensed internal code (LIC) และ i5/OS PASE Kernel

งานที่เกี่ยวข้อง

“การดู call stack” ในหน้า 124

คุณสามารถดูข้อมูลเกี่ยวกับงาน หรือ call stack ของ thread ได้โดยใช้ System i Navigator หรืออินเทอร์เฟซแบบอักขระ

คลาสอ็อบเจ็กต์:

คลาสอ็อบเจ็กต์มีแอตทริบิวต์การรันที่ควบคุมสถานะแวดล้อมรันไทม์ ของงาน คลาสอ็อบเจ็กต์ที่ได้มาจาก IBM, หรือคลาส, ใช้ได้กับทั้ง แอ็พพลิเคชันแบบโต้ตอบและแอ็พพลิเคชันแบบแบ็คชดตามปกติ คลาสที่นำมาพร้อมกักระบบ (ตามชื่อ) มีดังนี้:

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

แอตทริบิวต์รันไทม์

ข้อมูลต่อไปนี้เป็นรายการของแอตทริบิวต์รันไทม์, หรือพารามิเตอร์, บางส่วนที่พบในคลาสอ็อบเจ็กต์ ซึ่งมีความสำคัญต่อการจัดการระบบงาน

ระดับความสำคัญการรัน (RUNPTY)

ตัวเลขที่ระบุถึงระดับความสำคัญที่ได้กำหนดให้แก่งานทั้งหมดที่รันอยู่โดยใช้คลาส ระดับความสำคัญถูกใช้ในการ

กำหนดงาน, จากงานทั้งหมดที่แย่งชิงรีซอร์สของระบบ, ที่จะรันถัดไป ค่าอาจเป็นได้ตั้งแต่ 1 ถึง 99, โดยที่ 1 คือระดับความสำคัญสูงสุด (งานทั้งหมดที่มีระดับความสำคัญ 1 จะรันก่อน) ค่านี้คือระดับความสำคัญการรันสูงสุดที่ใช้ได้สำหรับ thread ใดๆ ภายในงาน แต่ละ thread 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

Time slice (TIMESLICE)

จำนวนสูงสุดของเวลาของโปรเซสเซอร์ (ในหน่วยมิลลิวินาที) ที่ให้แก่แต่ละ thread ในงานที่ใช้คลาสนี้ ก่อนที่ threads อื่นในงานหรืองานอื่นจะได้รับโอกาสให้รัน การแบ่งเวลาเป็นตัวกำหนดจำนวนเวลาที่ต้องการโดย thread ในงานในการทำให้การประมวลผลที่สำคัญเสร็จสมบูรณ์ เมื่อสิ้นสุดการแบ่งเวลา, thread อาจถูกกำหนดให้มีสถานะ inactive เพื่อให้ threads อื่นสามารถแอ็คทีฟในพูลของหน่วยเก็บได้

ช่วงเวลารอดีฟอลต์ (DFTWAIT)

จำนวนดีฟอลต์ของเวลาที่ระบบจะคอยการสมบูรณ์ของ คำสั่งที่ดำเนินการรอ ช่วงเวลารอนี้ใช้กับเวลาที่ คำสั่งกำลังรอการดำเนินการของระบบ, ไม่ได้ใช้กับเวลาที่คำสั่ง กำลังรอการตอบสนองจากผู้ใช้โดยทั่วไป, ช่วงเวลานี้เป็นเวลาที่คุณเต็มใจที่จะรอระบบก่อนที่จะจบคำขอ ถ้าผ่านพ้น ช่วงเวลารอแล้ว, ข้อความแสดงความผิดพลาดจะถูกส่งผ่านไปทำงาน จะใช้ช่วงเวลารอดีฟอลต์ นี้ถ้าไม่มีการระบุช่วงเวลารอค่าอื่นสำหรับสถานการณ์ที่กำหนดให้

ช่วงเวลารอที่ใช้สำหรับการจัดสรรรีซอร์สของไฟล์ถูกระบุอยู่ใน file description และสามารถแทนที่ค่าเดิมด้วยคำสั่งแทนที่ค่าเดิม คำสั่งนี้ระบุว่าจะให้ใช้ช่วงเวลารอที่ระบุในคลาสอ็อบเจกต์ ถ้ารีซอร์สของไฟล์ไม่มีพร้อมใช้งาน เมื่อเปิดไฟล์, ระบบจะรอจนกว่าช่วงเวลารอจะ สิ้นสุดลง

หมายเหตุ: คลาสแอ็คทีทริบิวต์ใช้กับขั้นตอนการเรดแต่ละชั้น ของงาน งานส่วนใหญ่มีขั้นตอนการเรดเพียงขั้นตอนเดียว แต่ถ้างานถูกรีเรด (เนื่องจากบางสิ่ง เช่น คำสั่ง Reroute Job (RRTJOB) หรือ Transfer Job (TFTJOB) คลาสแอ็คทีทริบิวต์จะถูกรีเซต

เวลา CPU สูงสุด (CPUTIME)

จำนวนสูงสุดของเวลาของโปรเซสเซอร์ที่อนุญาตสำหรับการประมวลผล ขั้นตอนการเรดของงานให้เสร็จสมบูรณ์ ถ้าขั้นตอนการเรดของงานไม่เสร็จสิ้นภายในเวลานี้, ขั้นตอนจะจบลง, และมีการบันทึกข้อความที่บันทึกการใช้งาน

หน่วยเก็บชั่วคราวสูงสุด (MAXTMPSTG)

จำนวนสูงสุดของหน่วยเก็บชั่วคราวที่ขั้นตอนการเรดของงาน สามารถใช้ได้ หน่วยเก็บชั่วคราวนี้ใช้สำหรับโปรแกรมที่รันอยู่ในงาน, สำหรับอ็อบเจกต์ระบบที่ใช้เพื่อสนับสนุนงาน, และสำหรับอ็อบเจกต์ชั่วคราว ที่งานสร้างขึ้น

Thread สูงสุด (MAXTHD)

จำนวนสูงสุดของ thread ซึ่งงานในคลาสนี้สามารถรันได้ในทุกเมื่อ ถ้าหลาย thread ถูกเริ่มต้นพร้อมกัน, ค่านี้อาจมากเกินได้ Thread ที่มากเกินไปจะได้รับอนุญาตให้รันจนเสร็จสมบูรณ์ตามปกติ การเริ่มต้น thread เพิ่มเติมจะถูกห้ามจนกว่าจำนวนสูงสุดของ thread ในงานจะลดลงต่ำกว่าค่าสูงสุดนี้

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

งานที่เกี่ยวข้อง

“การสร้างคลาสอ็อบเจกต์” ในหน้า 144

คุณสามารถสร้างคลาสอ็อบเจกต์โดยใช้อินเตอร์เฟซแบบอักษระ คลาสกำหนดแอ็คทีทริบิวต์การประมวลผลสำหรับงานที่ใช้

คลาส คลาสที่งานใช้ ถูกระบุอยู่ใน subsystem description routing entry ที่ใช้ในการเริ่มงาน ถ้างานมีขั้นตอนการเรอต์หลาย ขั้นตอน คลาสที่ใช้โดย ขั้นตอนการเรอต์แต่ละขั้นตอนจะถูกระบุอยู่ใน routing entry ที่ใช้ในการเริ่มต้นขั้นตอนการเรอต์

“การเปลี่ยนคลาสอ็อบเจกต์” ในหน้า 145

คุณสามารถเปลี่ยนแอ็ททริบิวต์ของคลาสอ็อบเจกต์ได้โดยใช้อินเตอร์เฟสแบบอักขระ สามารถเปลี่ยนแอ็ททริบิวต์ต่างๆ ยกเว้นแอ็ททริบิวต์สิทธิ์พัลลิก อ้างอิงคำสั่ง Revoke Object Authority (RVKOBJAUT) และคำสั่ง Grant Object Authority (GRTOBJAUT) สำหรับข้อมูลเพิ่มเติมเกี่ยวกับการเปลี่ยนการให้สิทธิ์อ็อบเจกต์

Job user identity:

job user identity (JUID) คือชื่อของโปรไฟล์ผู้ใช้ซึ่งทำงานนี้เป็นที่รู้จักของงานอื่น ชื่อนี้ใช้สำหรับการตรวจสอบการให้สิทธิ์ เมื่องานอื่นพยายามดำเนินการกับงานนี้

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

ในสถานการณ์ที่งานสลับค่าโปรไฟล์ผู้ใช้โปรไฟล์ผู้ใช้ปัจจุบันระบุ โปรไฟล์ซึ่ง initial thread กำลังรันอยู่ภายใต้ แทนที่จะเป็น JUID

ไม่ใช่ JUID ในการตรวจสอบการให้สิทธิ์จากภายในงาน การให้สิทธิ์เพื่อปฏิบัติฟังก์ชันจะใช้โปรไฟล์ผู้ใช้ปัจจุบันของ thread ซึ่งฟังก์ชันถูกเรียกเสมอ

เมื่องานอยู่บนคิวงานหรือเอาต์พุตคิว JUID จะเหมือนกับชื่อผู้ใช้ของงานเสมอ และไม่สามารถเปลี่ยนได้

เมื่องานเริ่มขึ้น และที่จุดเริ่มต้นของขั้นตอนการเรอต์ที่ตามมาใดๆ, JUID คือชื่อที่เหมือนกับชื่อของโปรไฟล์ผู้ใช้ปัจจุบันของงาน ในขณะที่งานแอ็คทีฟ สามารถเปลี่ยน JUID ในวิธีดังต่อไปนี้

- JUID สามารถถูกเซ็ทโดยตรงโดยแอ็พพลิเคชันโดยใช้ Set Job User Identify (QWTSJUID) application program interface (API) หรือฟังก์ชัน QwtSetJuid() JUID ถูกเซ็ทด้วยชื่อของโปรไฟล์ผู้ใช้ที่ thread ซึ่งเรียก API หรือฟังก์ชันกำลังรันอยู่ภายใต้โปรไฟล์นั้น
- JUID สามารถถูกลบโดยตรงโดยแอ็พพลิเคชันโดยใช้ QWTSJUID API หรือฟังก์ชัน QwtClearJuid() งานต้องกำลังรันเป็นงานแบบ single threaded ในขณะนั้น เมื่อลบ JUID จะถูกเซ็ทโดยทางอ้อมโดยระบบเป็นชื่อของโปรไฟล์ผู้ใช้ที่ single thread ของงานกำลังรันอยู่ภายใต้ ณ จุดนั้น
- ถ้างานกำลังรันเป็นงานแบบ single threaded และ JUID ไม่ได้ถูกเซ็ทโดยตรงโดยแอ็พพลิเคชัน ผลคือ ในแต่ละครั้งที่งานใช้ Set Profile (QWTSETP) API เพื่อรันภายใต้โปรไฟล์ผู้ใช้อื่น JUID จะถูกเซ็ทโดยทางอ้อม โดยระบบเป็นชื่อของโปรไฟล์ผู้ใช้ที่ถูกเซ็ทโดย QWTSETP
- เมื่องานแบบ single threaded เริ่ม secondary thread และ JUID ยังไม่ได้ถูก เซ็ทโดยตรงโดยแอ็พพลิเคชัน ผลคือ ระบบจะเซ็ท JUID โดยทางอ้อมด้วยชื่อของโปรไฟล์ผู้ใช้ที่ single thread ของงาน กำลังรันอยู่ภายใต้ ณ จุดที่งานเริ่ม secondary thread

เมื่องาน กลับไปเป็น single thread, ระบบจะเซ็ท JUID โดยทางอ้อมเป็นชื่อของ โปรไฟล์ผู้ใช้ที่ single thread ของงานกำลังรันอยู่ภายใต้ ณ จุดนั้น

หลักการที่เกี่ยวข้อง

สิทธิในการใช้งานที่เหมาะสม

ในการเปลี่ยนแปลงแอตทริบิวต์ของงานส่วนใหญ่ คุณต้องมีสิทธิพิเศษในการควบคุมงาน (*JOBCTL) หรือโปรไฟล์ผู้ใช้ของคุณที่ต้องตรงกับ job user identity ของงานที่จะเปลี่ยน

ตัวอย่าง Job user identity:

ตัวอย่างเหล่านี้แสดงถึงวิธีการกำหนด job user identity (JUID) ในสถานการณ์ต่างๆ

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

ในสถานการณ์นี้ ค่า Set By สำหรับ JUID คือ *DEFAULT เนื่องจากงาน กำลังรันแบบ single-threaded ดังนั้น job user identity คือโปรไฟล์ผู้ใช้ปัจจุบัน ซึ่ง initial thread ของงานกำลังรันอยู่ภายใต้ (ยกเว้นว่า job user identity จะถูกเซตโดยตรงโดยแอสพลีเคชัน) สำหรับงานคิวงาน และงานที่เสร็จสมบูรณ์ job user identity คือชื่อผู้ใช้จากชื่องานที่ต้องการ

- 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 เนื่องจากงานนี้เป็นงานแอคทีฟซึ่ง กำลังรันในปัจจุบันเป็นงาน multi-threaded ดังนั้น job user identity จึงถูกเซต โดยทางอ้อมโดยระบบ job user identity จะถูกเซตเป็นชื่อของโปรไฟล์ผู้ใช้ ซึ่งงานจะรันอยู่ภายใต้มัน เมื่องานกลายเป็น multi-threaded เมื่องานกลับ ไปรัน single-threaded, job user identity จะถูกรีเซตเป็นค่า *DEFAULT

- ถ้าเซิร์ฟเวอร์ที่รันภายใต้โปรไฟล์ผู้ใช้ที่มีชื่อว่า SERVER เรียก QWTSJUID API, JUID จะถูกเซตเป็น SERVER จากนั้นถ้าเซิร์ฟเวอร์เรียก Set Profile (QWTSETP) API เพื่อเซตโปรไฟล์ผู้ใช้ปัจจุบันเป็น CLIENT ในขณะที่ประมวลผลงาน ในนามของโคลเอนต์นั้น JUID ยังคงเป็น SERVER ในลักษณะเดียวกัน ถ้าเซิร์ฟเวอร์ เริ่ม secondary threads ที่แต่ละ thread เรียก QWTSETP เพื่อรันภายใต้ โปรไฟล์ผู้ใช้ที่หลากหลายโปรไฟล์ JUID จะยังคงเป็น SERVER

ในสถานการณ์นี้ ค่า Set By สำหรับ JUID คือ *APPLICATION Job user identity ถูกเซตโดยตรงโดย แอสพลีเคชัน โดยใช้ API ค่านี้ใช้กับงานทั้งแบบ single-threaded และ multi-threaded

thread:

คำว่า *thread* ย่อมาจาก "thread of control" thread คือพาทที่โปรแกรมใช้ในขณะรัน, ขั้นตอนปฏิบัติ และลำดับ การปฏิบัติตามขั้นตอน thread รันได้จากตำแหน่งเริ่มต้น ในลำดับที่จัดลำดับ ที่กำหนดไว้ล่วงหน้าสำหรับชุดของอินพุตที่กำหนดให้

การใช้ thread ภายในงานช่วยให้สามารถทำสิ่งต่างๆ ได้ในเวลาเดียวกัน ตัวอย่างเช่น ขณะที่งานกำลังประมวลผล thread อาจเรียกข้อมูลออกมาดูและคำนวณข้อมูลที่งานต้องการเพื่อให้การประมวลผลเสร็จสิ้น

งานแอคทีฟแต่ละตัวมี thread อย่างน้อยหนึ่งตัว, ซึ่งเรียกว่า initial thread initial thread จะถูกสร้างขึ้นเป็นส่วนหนึ่งของการเริ่มต้นของงาน ใน thread บน System i Navigator โดยค่าดีฟอลต์ คุณจะเห็น **Initial** เป็นชนิดของ thread แรกในรายการ initial thread คือ thread แรกที่ถูกสร้างขึ้นภายในงานเมื่อเริ่มทำงาน

ชนิด thread

ชนิด thread กำหนดวิธีการสร้าง thread บนระบบ

ผู้ใช้ thread สามารถถูกสร้างขึ้นโดยแอ็พพลิเคชันลูกค้ำ initial thread ในงานจะเป็น thread ผู้ใช้เสมอ ฟิลด์ Allow multiple threads ต้องถูกเซ็ทเป็น yes เพื่อให้สามารถใช้ thread ผู้ใช้หลายตัวได้

ระบบ thread จะถูกสร้างขึ้นโดยระบบในนามของผู้ใช้ ฟังก์ชันระบบบางอย่างใช้ thread ระบบเพื่อเสร็จสิ้นการประมวลผล หากแอ็พพลิเคชันของลูกค้ำใช้ฟังก์ชันระบบที่ใช้ thread, thread ระบบจะถูกใช้งาน

งานที่เกี่ยวข้อง

“การดูคุณสมบัติของ thread” ในหน้า 147

thread จะช่วยให้งานสามารถทำงานหลายๆ อย่างได้ในเวลาเดียวกัน หาก thread หยุดการประมวลผล thread จะหยุดงานไม่ให้เห็น

“การดู thread ที่กำลังรันภายใต้งานเฉพาะ” ในหน้า 145

งานแอ็คทีฟทุกงานที่กำลังรันบนระบบมี thread ที่กำลังรันอยู่ภายใต้งานอย่างน้อยหนึ่ง thread thread คือหน่วยงานอิสระที่รันอยู่ภายในงานที่ใช้รีซอร์สเดียวกันกับงาน เนื่องจากงานขึ้นอยู่กับงานที่ทำ โดย thread จึงมีความสำคัญที่ต้องทราบวิธีการค้นหา threads ที่กำลังรัน ภายในงานเฉพาะ

“การจบ หรือการลบ thread” ในหน้า 148

initial thread ซึ่งถูกสร้างเมื่องานเริ่มทำงาน ไม่สามารถถูกลบทิ้งหรือจบการทำงานได้ อย่างไรก็ตาม บางครั้งก็เป็นเรื่องจำเป็นที่ต้องจบการทำงานของ thread รองเพื่อให้งานสามารถรันต่อไปได้ ให้ระวัง thread ที่คุณตั้งใจจะจบ เนื่องจากงานที่รันอยู่ภายใน thread อาจไม่สมบูรณ์ถ้าไม่มีงานของ thread นั้น

ข้อมูลที่เกี่ยวข้อง

ตัวอย่าง: จบ thread โดยใช้ Java

API การจัดการ thread

สิทธิในการใช้งาน thread ที่ถูกต้อง:

ต้องมีระดับสิทธิในการใช้งานบางอย่างก่อน คุณจึงจะสามารถทำงานกับ thread ได้

เพื่อดูและเปลี่ยนแปลงแอ็ททริบิวต์ส่วนใหญ่ของ thread คุณต้องมีสิทธิพิเศษ *JOBCTL หรือโปรไฟล์ผู้ใช้ของคุณต้องตรงกับ job user identity ของงานที่มี thread หากต้องการเปลี่ยนระดับความสำคัญการรันของ thread คุณต้องมีสิทธิพิเศษ *JOBCTL Thread Control authority allows you to view some of the attributes of a thread

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

สำหรับแอ็ททริบิวต์ thread ใดๆที่อ้างอิงถึงอ็อบเจกต์ System i เช่น โลบารรีในรายชื่อโลบารรี ผู้ใช้ต้องมีสิทธิที่ถูกต้องที่อ็อบเจกต์

สำหรับรายละเอียดเพิ่มเติมเกี่ยวกับ i5/OS สิทธิ ให้ดูที่ สิทธิที่จำเป็นสำหรับอ็อบเจกต์ที่ใช้โดยคำสั่งในกลุ่มหัวข้อ Security reference

หมายเหตุ: สิทธิ thread Control ช่วยให้คุณสามารถดึงข้อมูลเกี่ยวกับ thread ของงานอื่นได้ Thread Control สามารถให้ และเรียกคืนสำหรับผู้ใช้แต่ละรายได้โดยใช้ System i Navigator Application Administration support หรือโดยการใช้ Change Function Usage Information (QSYCHFUI) API ด้วย function ID ของ QIBM_SERVICE_THREAD สำหรับ ข้อมูลรายละเอียดเพิ่มเติมเกี่ยวกับการจัดการแอ็พพลิเคชัน ให้ดู Information Center หัวข้อ Application Administration

สถานะ thread:

สถานะปัจจุบันของ thread สามารถดูได้จากหน้า General ในหน้าต่าง Thread Properties ภายใต้สถานะ Detailed

ตัวอย่างของสถานะโดยละเอียดคือ:

กำลังรอถอนจากคิว

thread ของงานกำลังรอการดำเนินการถอนจากคิวที่เสร็จสมบูรณ์ การถอนจากคิวคือการดำเนินการเพื่อย้ายข้อความจากคิว ข้อความคือการสื่อสารที่ถูกส่งจากบุคคลหนึ่งหรือโปรแกรมหนึ่งไปยังอีกบุคคลหนึ่งหรือโปรแกรมหนึ่ง โดยทั่วไป ข้อความจะถูกวางคิว (วาง) ไว้บนอ็อบเจกต์ระบบคิวโดย thread หนึ่งตัวและถูกถอนออกจากคิว (ลบออก) โดย thread อีกตัวหนึ่ง

หมายเหตุ: เมื่อการรอการถอนจากคิวแสดงอยู่ในหน้าคุณสมบัติ ข้อมูลเพิ่มเติมที่ระบุคิวที่กำลังรออยู่จะปรากฏขึ้นมา ถ้างาน หรือ thread กำลังรอการดำเนินการถอนออกจากคิวให้เสร็จสิ้นสำหรับอ็อบเจกต์ i5/OS คุณจะเห็นชื่ออ็อบเจกต์ 10 อักขระ, โลบารีย์ของอ็อบเจกต์ แลชชนิดอ็อบเจกต์ ถ้างานหรือ thread กำลังรอการดำเนินการถอนออกจากคิวให้เสร็จสิ้นสำหรับอ็อบเจกต์ภายใน คุณจะเห็นชื่ออ็อบเจกต์ที่ยาว 30 อักขระ สำหรับอ็อบเจกต์ภายใน คุณต้องมีสิทธิ์พิเศษ job control (*JOBCTL) เพื่อดูชื่อที่ยาว 30 อักขระ

สถานะโดยละเอียดสามารถแสดงค่าสถานะที่เชื่อมโยง ซึ่งให้รายละเอียดเพิ่มเติม เกี่ยวกับสถานะปัจจุบันของ thread ตัวอย่างของสถานะโดยละเอียดบวกค่าสถานะเชื่อมโยงคือ:

ถูกพัก (n)

thread แต่ละอันจะถูกพักไว้ซึ่งต่างจากงาน 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 ให้ดูที่ System i Navigator คำอธิบายทางออนไลน์

อ็อบเจกต์ที่ถูกล็อก:

งานและ threads ใช้อ็อบเจกต์ในการประมวลผลงาน

เนื่องจากมีงานมากกว่าหนึ่งชิ้นที่ประมวลผลในหนึ่งครั้ง จึงมีการล็อกอ็อบเจกต์เพื่อรักษา data integrity ไว้ **อ็อบเจกต์ที่ถูกล็อก** คืออ็อบเจกต์ระบบที่ถูกใช้โดยงานและ thread เพื่อประมวลผลงาน หลังจากทำงานหรือ thread รันเสร็จสิ้นแล้ว อ็อบเจกต์จะถูกปลดล็อกและพร้อมที่จะใช้ ประมวลผลงานเพิ่มเติม การล็อกอ็อบเจกต์จะอนุญาตให้ผู้ใช้เพียงรายเดียวใช้งานอ็อบเจกต์ในหนึ่งครั้ง ทั้งนี้ขึ้นอยู่กับประเภทค่าขอการล็อกที่ใช้ ตัวอย่างเช่น ถ้าผู้ใช้สองราย หรือมากกว่าพยายามเปลี่ยนอ็อบเจกต์ในเวลาเดียวกัน การเปลี่ยนแปลงที่อ็อบเจกต์โดยผู้ใช้ที่สองจะถูกล็อก จนกว่าผู้ใช้แรกจะอัปเดตอ็อบเจกต์ เสร็จสิ้น ด้วยการ ใช้ lock holders, ผู้ใช้สามารถดูได้ว่าสิ่งใดมีการล็อก อยู่ในปัจจุบันหรือกำลังรอการล็อกอ็อบเจกต์อยู่ในปัจจุบัน

ขอบเขต ระบุว่าการล็อกเกี่ยวข้องกับงาน, thread หรือ lock space ขอบเขตมักจะกำหนดว่าจะมีการล็อกอยู่นานเท่าไรและชนิด คำขอล็อกและกฎความขัดแย้งใดที่อ็อบเจกต์มี

ชนิดคำขอล็อก คือ ระดับการเข้าถึงอ็อบเจกต์ที่จะถูกล็อกของงาน, thread หรือ lock space ตัวอย่างเช่น lock exclusive จะไม่มีการใช้ read lock type ถ้าอ็อบเจกต์กำลังถูกเปลี่ยนหรือ ลบบนระบบ คำขอล็อกชนิดนี้ไม่อนุญาตให้บุคคลใดใช้อ็อบเจกต์ และไม่อนุญาตให้บุคคลใดอ่านอ็อบเจกต์

ชนิดคำขอล็อกอื่นๆ คือ:

Exclusive – No read

อ็อบเจกต์จะถูกสงวนไว้เพื่อการใช้งานเฉพาะ อย่างไรก็ตาม หากอ็อบเจกต์ถูกล็อกโดยคำขอล็อกชนิดใดก็ตาม คุณไม่สามารถได้สิทธิเฉพาะในการใช้งานอ็อบเจกต์นั้น สถานะล็อก นี้มีความเหมาะสม เมื่อผู้ใช้ไม่ต้องการให้ผู้อื่นเข้าใช้งานอ็อบเจกต์จนกว่าฟังก์ชันที่กำลังปฏิบัติการจะเสร็จสมบูรณ์

Exclusive-Read

สามารถใช้งานอ็อบเจกต์ร่วมกับชนิดคำขอล็อกแบบ shared-read การล็อกนี้เหมาะสมเมื่อผู้ใช้ต้องการป้องกันไม่ให้ผู้ใช้รายอื่นปฏิบัติการดำเนินการใดๆ นอกเหนือจากการอ่าน

Shared-Update

สามารถใช้งานอ็อบเจกต์ร่วมกับชนิดคำขอล็อกแบบ shared-read หรือ shared-update นั่นคือ ผู้ใช้รายอื่นสามารถร้องขอสถานะการล็อกแบบ shared-read หรือ shared-update ให้กับอ็อบเจกต์เดียวกันได้ สถานะล็อกนี้เหมาะสม เมื่อผู้ใช้ตั้งใจที่จะเปลี่ยนอ็อบเจกต์ แต่ต้องการอนุญาตให้ผู้ใช้รายอื่นสามารถอ่านหรือเปลี่ยนอ็อบเจกต์เดียวกันได้

Shared-No update

สามารถใช้งานอ็อบเจกต์ร่วมกับชนิดคำขอล็อกแบบ share - no update และ shared-read เท่านั้น สถานะล็อกนี้เหมาะสม เมื่อผู้ใช้ไม่ต้องการ เปลี่ยนอ็อบเจกต์ แต่ต้องการทำให้มั่นใจว่าไม่มีผู้ใช้รายอื่นเปลี่ยนอ็อบเจกต์

Shared-Read

สามารถใช้งานอ็อบเจกต์ร่วมกับชนิดคำขอล็อกทั้งหมดนอกเหนือจากแบบ exclusive - no read นั่นคือ ผู้ใช้อื่นสามารถร้องขอชนิดล็อกแบบ exclusive-read, shared-update, shared-read หรือ shared-no update

สถานะการล็อกเป็นตัวบอกสถานะของคำขอล็อก 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 ที่กำลังถือครองล็อกอยู่หรือกำลังรอการล็อกบนอ็อบเจกต์ที่ถูกล็อกที่ระบุไว้

ชนิดของงาน

ระบบของคุณประมวลงานได้หลายชนิด ข้อมูลนี้อธิบายงานเหล่านั้น และวิธีที่งานถูกใช้

งานแบบ autostart:

งานแบบ autostart คืองานแบ็คที่ทำงานซ้ำ, งาน การกำหนดค่าเริ่มต้นหนึ่งครั้งที่เชื่อมโยงกับระบบย่อย, initializes ฟังก์ชันสำหรับแอปพลิเคชัน, หรือนำเสนอฟังก์ชันการให้บริการศูนย์กลาง สำหรับงานอื่นในระบบย่อยเดียวกัน งานแบบ autostart ในระบบย่อยการควบคุม สามารถใช้ในการเริ่มระบบย่อยอื่น (เช่นระบบย่อยการควบคุม ที่ได้มาจาก IBM) งานแบบ autostart ที่เชื่อมโยงกับระบบย่อยจะถูกเริ่มต้นโดยอัตโนมัติ ในทุกครั้งที่ระบบย่อยถูกเริ่มขึ้น

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

คำอธิบายงานที่ใช้กับ autostart job ระบุไว้ด้วยการใช้คำสั่ง Add Autostart Job Entry (ADDAJE) เมื่อระบบย่อย เริ่มขึ้น, งาน จะดำเนินการภายใต้ชื่อโปรไฟล์ผู้ใช้ในรายละเอียดของงาน ที่ระบุ คุณไม่สามารถระบุรายละเอียดของงานที่มี USER(*RQD) เนื่องจากงานแบบ autostart ดำเนินการภายใต้โปรไฟล์ผู้ใช้ ที่ถูกระบุโดยรายละเอียดของงาน, คุณจึงต้องควบคุมบุคคลที่สามารถ จะเปลี่ยนรายละเอียดของงานได้

ถ้ามีการระบุงานแบบ autostart มากกว่าหนึ่งงานสำหรับระบบย่อยหนึ่ง, งานแบบ autostart ทั้งหมดจะถูกเริ่มในทันทีแทนที่จะเริ่มทีละงาน ถ้าจำนวนของงานมาก เกินจำนวนสูงสุดของงานของระบบย่อย, จะไม่มีงานอื่นใดสามารถเริ่มขึ้นได้ใน ระบบย่อย จนกว่าจะมีงานแบบ autostart เสร็จสมบูรณ์มากเพียงพอ เพื่อให้จำนวน ของงานที่รันอยู่ต่ำกว่า activity level สูงสุด

งานแบดซ์:

งานแบดซ์คือกลุ่มของการประมวลผลที่กำหนดไว้ล่วงหน้า ซึ่งถูกส่ง ไปที่ระบบเพื่อปฏิบัติการ โดยมีการโต้ตอบระหว่างผู้ใช้ และระบบน้อยมากหรือ ไม่มีการโต้ตอบเลย งานที่ไม่ต้องการการโต้ตอบกับผู้ใช้ในการรันสามารถถูก ประมวลผลเป็นงานแบดซ์ โดยปกติ งานแบดซ์เป็นงานที่มีระดับความสำคัญต่ำและอาจ ต้องการสภาวะแวดล้อมระบบพิเศษในการรัน

งานแบดซ์จะทำงานอยู่ในส่วนแบดคราวด์ของระบบ เพื่อให้ผู้ใช้งานที่ส่งงานเข้ามาสามารถทำงานอื่นได้ งานแบดซ์สามารถแอสคที่ฟพร้อมกันได้หลายงาน

รายการต่อไปนี้จะแสดงชนิดต่างๆ ของงานแบดซ์:

งานแบดซ์แบบธรรมดา

งานแบดซ์แบบธรรมดาคืองานที่ถูกส่งไปที่คิวงาน งานชนิดนี้ จะรออยู่ในแถวพร้อมกับงานแบดซ์ชนิดอื่น และถูกประมวลผลตามระดับความสำคัญ และหมายเลขลำดับ

งานแบดซ์แบบ immediate

งานแบดซ์แบบ immediate เป็นงานที่ถูกเริ่มขึ้นพร้อมกับแอตทริบิวต์ต่างๆ ของงานตัวแม่ งานจะรันอยู่ในระบบย่อยเดียวกันกับงานแม่ (งานนี้สำเร็จได้โดยใช้ spawn() API) เนื่องจากงาน จะก๊อปปี้แอตทริบิวต์มาจากงานแม่และไม่ผ่านคิวงาน งานนี้จึงสามารถเริ่มได้รวดเร็วกว่างานที่ส่งไปที่คิวงาน

งานแบดซ์แบบ MRT

งานแบบแบดซ์ MRT คืองานแบบ multiple requester terminal (MRT) งานแบบ MRT เป็นงานของ S/36 Environment ที่ทำงานเหมือนเป็นเซิร์ฟเวอร์ ที่จะอนุญาตให้งาน S/36 Environment ตัวอื่นสามารถผนวกติดตามด้วย เพื่อที่จะรันโปรแกรม MRT ด้วยกัน

งานแบดซ์แบบพิมพ์

งานแบดซ์แบบพิมพ์จะติดตามดูโพล์พริเตอร์เอาต์พุต (หรือเรียกอีกอย่างว่าสพูลไฟล์) ที่ถูกสร้างขึ้นโดยงานที่มีโปรไฟล์ผู้ใช้งานปัจจุบันแตกต่างไปจากโปรไฟล์ผู้ใช้ในตอนที่ยานเริ่มต้น

งานแบดซ์สามารถเริ่มได้เมื่อผู้ใช้:

- ทำให้งานอยู่ในคิวงาน
- ออกคำขอการเริ่มโปรแกรมการสื่อสาร

- เริ่มระบบย่อยพร้อมด้วยงานแบบ prestart
- ใช้ spawn() API

วิธีการเริ่มงานแบ็ตซ์:

เมื่อผู้ใช้ส่งงานแบ็ตซ์ งานจะรวบรวมข้อมูลจากอ็อบเจ็กต์ระบบต่างๆ ก่อนที่งานจะถูกวางบนคิวงาน

1. ผู้ใช้ส่งงาน
2. งานจะค้นหาแอ็ททริบิวต์งาน หากไม่พบแอ็ททริบิวต์งานบนคำสั่ง Submit Job (SBMJOB) งานจะค้นในรายละเอียดของงาน (ที่ระบุบนคำสั่ง SBJOB), โปรไฟล์ผู้ใช้ของผู้ใช้ปัจจุบัน และงานที่แอ็ททิฟในปัจจุบัน (งานที่ออกคำสั่ง SBJOB)

หมายเหตุ: ในลักษณะคล้ายกันกับการเริ่มงานแบบโต้ตอบ คุณสามารถระบุในรายละเอียดของงานให้ใช้โปรไฟล์ผู้ใช้โปรไฟล์ผู้ใช้สามารถระบุให้ใช้ค่ากำหนดของระบบในการค้นหาแอ็ททริบิวต์งาน

3. หลังจากทำงานมีแอ็ททริบิวต์ครบทั้งหมดแล้ว งานจะอยู่บนคิวงาน
4. เมื่อระบบย่อยพร้อมที่จะจัดการงาน ระบบย่อยจะค้นหางานใน คิวงาน (คิวงานที่ระบบย่อยได้จัดสรร)
5. จากนั้น เช่นเดียวกับขั้นตอนของงานแบบโต้ตอบ ระบบย่อยจะตรวจสอบ รายละเอียดของงานสำหรับข้อมูลการเราต์
6. ระบบย่อยใช้ข้อมูลการเราต์ในการค้นหา routing entry routing entry นำเสนอข้อมูลพูลทำงานจะใช้, โปรแกรมเราต์ที่จะใช้ และคลาสที่งานจะเรียกแอ็ททริบิวต์รันไทม์ของงานมา
7. หลังจากได้ข้อมูลนี้แล้ว โปรแกรมการเราต์จะรัน ถ้าคุณใช้ QCMD, QCMD จะใช้คำสั่ง SBJOB คำสั่งจะรันคำสั่งที่ระบุบนพารามิเตอร์ CMD หรือ QSDTA

งานที่เกี่ยวข้อง

“การส่งงานแบ็ตซ์” ในหน้า 131

เนื่องจากโดยปกติ งานแบ็ตซ์เป็นงานที่มีระดับความสำคัญต่ำที่ต้องการ สภาวะแวดล้อมระบบพิเศษในการรัน (เช่น การรันในเวลากลางคืน) งานแบ็ตซ์จึงถูกวางในคิวงานแบ็ตซ์ในคิวงาน งานแบ็ตซ์จะได้รับตารางเวลารันไทม์ และระดับความสำคัญ ถ้าต้องการส่งงานไปที่คิวงานแบ็ตซ์ให้คุณ ใช้อินเตอร์เฟสแบบอักขระและหนึ่งในสองคำสั่งต่อไปนี้

“การเริ่มงานแบ็ตซ์ที่กำลังรออยู่ในคิวงาน” ในหน้า 133

ในบางโอกาส คุณอาจต้องการบังคับให้งานเริ่มในทันทีในขณะที่การย้ายงานไปที่คิวงานที่ไม่ยุ่งเป็นวิธีการที่มีประสิทธิภาพสูงสุด แต่มีวิธีการอื่นที่คุณสามารถใช้ได้

ข้อมูลที่เกี่ยวข้อง

งาน QPRTJOB

การ Spawn งานแบ็ตซ์:

Spawn คือฟังก์ชันที่สร้างกระบวนการงานใหม่ (child process) ที่ได้รับแอ็ททริบิวต์หลายอย่างของกระบวนการเรียก (parent process) โปรแกรมใหม่จะถูกระบุและเริ่มการรันใน child process เมื่อคุณ spawn งานแบ็ตซ์ นั่นคือคุณกำลังใช้ parent job เพื่อส่งผ่าน อากิวเมนต์และตัวแปรสภาวะแวดล้อมไปที่ child job spawn() API ใช้งานแบ็ตซ์แบบ immediate, งานแบบ prestart หรืองานแบ็ตซ์แบบ prestart

ข้อมูลที่เกี่ยวข้อง

spawn() -- Spawn Process

คำสั่ง SPAWN CL ตัวอย่าง QUSRTOOL

งานสื่อสาร:

งานสื่อสารคืองานแบ็คซึ่ที่ถูกเริ่มขึ้นโดยคำขอการเริ่มโปรแกรมจากระบบรีโมต การประมวลผลงานเกี่ยวข้องกับคำขอการสื่อสาร และข้อกำหนดคุณลักษณะที่เหมาะสม

สำหรับงานแบ็คซึ่การสื่อสารที่จะรันบนระบบ i5 /OS, subsystem description ที่มี work entry สำหรับงานสื่อสารต้องมีอยู่ระบบรายการงานสื่อสาร ระบุให้ระบบย่อยทราบถึงซอร์สสำหรับงานสื่อสารที่ระบบจะประมวลผล การประมวลผลงานเริ่มขึ้นเมื่อระบบย่อยได้รับ คำขอการเริ่มโปรแกรมการสื่อสารจากระบบรีโมตและพบ routing entry ที่เหมาะสมสำหรับคำขอ

ข้อมูลการเรอต์สำหรับงานสื่อสาร

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

หากต้องใช้สภาวะแวดล้อม การประมวลผลพิเศษสำหรับงานสื่อสารบางงาน, คุณสามารถ ใส่เพิ่ม routing entry เพิ่มเติมที่ subsystem description, โดยระบุ ค่าเปรียบเทียบที่มีตำแหน่งเริ่มต้นเป็น 37 ค่าเปรียบเทียบนี้ควรมี ชื่อโปรแกรมสำหรับคำขอการเริ่มโปรแกรม routing entry ต้องมี หมายเลขลำดับต่ำกว่า routing entry ที่ใช้ PGMEVOKE เป็นค่าเปรียบเทียบ วิธีนี้ช่วยให้งานสื่อสารบางงานสามารถรันด้วยข้อกำหนดคุณลักษณะ คลาสหรือพูลที่แตกต่างกันได้

การรักษาความปลอดภัย

ความปลอดภัยบนระบบควบคุมบุคคลที่สามารถใช้อุปกรณ์สื่อสาร และบุคคลที่สามารถเข้าใช้คำสั่งที่ใช้กับ device description คุณควรพิจารณามาตรการการรักษาความปลอดภัยเพิ่มเติมเมื่อ บันทึกลงและรันแอปพลิเคชันโปรแกรมบนทั้งระบบรีโมตและระบบเป้าหมาย.

รายละเอียดของงานสำหรับงานสื่อสาร

รายละเอียดของงานที่ใช้กับงานสื่อสารถูกระบุบนคำสั่ง Add Communications Entry (ADDCMNE) ผู้ใช้ที่ระบุบนรายละเอียดของงานนี้ จะถูกข้ามไป ระบบเรียกชื่อผู้ใช้สำหรับงานสื่อสารมาจาก คำขอการเริ่มโปรแกรม หากคำขอการเริ่มโปรแกรมไม่ได้ระบุชื่อผู้ใช้, ระบบจะใช้ค่าผู้ใช้ที่กำหนดไว้จากการป้อนข้อมูลแบบ communications เพื่อให้มั่นใจว่าระบบมีการรักษาความปลอดภัยที่แน่นหนามากขึ้น, ให้รวมข้อมูลผู้ใช้ในคำขอการเริ่ม โปรแกรม แทนที่จะระบุผู้ใช้ที่กำหนดไว้ในการป้อนข้อมูลงานแบบ communication

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 โพรเซสเซอร์เดียว

Qsyscomm1 (system communications)

งานตัวนี้จะจัดการการสื่อสารบางตัวและ activity อินพุต/เอาต์พุต (I/O).

Q400filsvr (remote file system communication)

งานตัวนี้จะทำการสื่อสารแบบ common programming interface ทั่วไป (APPN หรือ APPC) สำหรับระบบรีโมตไฟล์

งานแบบโต้ตอบ:

งานแบบโต้ตอบคืองานที่เริ่มขึ้นเมื่อผู้ใช้ signs on ที่ จอภาพ และสิ้นสุดลงเมื่อผู้ใช้ signs off สำหรับงานที่จะรัน ระบบย่อย จะค้นหารายละเอียดของงาน ซึ่งสามารถระบุใน workstation entry หรือโปรไฟล์ผู้ใช้

งานแบบโต้ตอบต้องการการสื่อสารสองทางอย่างต่อเนื่องระหว่างผู้ใช้ และระบบเพื่อปฏิบัติการกิจ งานโต้ตอบจะเริ่มต้นเมื่อผู้ใช้ sign on เข้าสู่ระบบ ระบบจะร้องขอข้อมูล sign-on ถ้าระบบยอมรับคำขอ sign-on request ระบบจะสร้างงานแบบโต้ตอบ หลังจากนั้นระบบจะขอให้ผู้ใช้ส่งคำขอ หลังจากนั้นผู้ใช้ต้องใส่ คำขอ และระบบจะตอบรับโดยการประมวลผลคำขอนั้น รูปแบบนี้จะทำซ้ำจนกว่าผู้ใช้จะจบงาน แบบโต้ตอบโดยการ signing off ระบบ หรืองานจบเนื่องจาก application exception หรือ การแก้ไขข้อผิดพลาดอุปกรณ์

หากงานโต้ตอบเป็นส่วนหนึ่งของกลุ่มงาน หรือคู่ของงาน งานแบบโต้ตอบจะมีประเภทของงานอย่างไรอย่างหนึ่งต่อไปนี้:

Interactive – Group

งาน Interactive – Group เป็นส่วนของกลุ่มของงานที่เกี่ยวข้องกับ อุปกรณ์แสดงผลเดียว

Interactive – System request

งาน Interactive – System request เป็นหนึ่งในคู่ของงานซึ่งเกี่ยวข้อง ซึ่งกันและกันโดยฟังก์ชันคำขอระบบ

คุณทราบหรือไม่? คุณสามารถ sign on ที่ระบบได้สองวิธี คุณสามารถเข้าสู่ระบบด้วยตนเองโดยใช้ user id และรหัสผ่าน และคุณยังสามารถสร้างโปรแกรมเพื่อส่ง user id และรหัสผ่านไปที่เซิร์ฟเวอร์โดยข้ามจอภาพ sign-on

วิธีการเริ่มงานแบบโต้ตอบ:

เมื่อผู้ใช้ signs on ที่ระบบ ระบบย่อยจะรวบรวมข้อมูลจาก อีอบเจ็กต์ระบบต่างๆ ก่อนที่งานแบบโต้ตอบจะพร้อมใช้งาน

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

หมายเหตุ: ความยืดหยุ่นนี้ช่วยให้คุณระบุได้ว่า แอ็ดทริบิวต์ของงาน จะอยู่ที่เวิร์กสเตชันหรืออยู่ที่ผู้ใช้แต่ละราย

2. หลังจากที่ระบบย่อยทราบถึงรายละเอียดของงานที่จะใช้แล้ว ระบบย่อยอาจจะไม่ ค้นหาแอ็ดทริบิวต์งานทั้งหมดในรายละเอียดของงาน บางแอ็ดทริบิวต์อาจอยู่ใน โปรไฟล์ผู้ใช้ ถ้าโปรไฟล์ผู้ใช้ไม่มีข้อมูล ระบบย่อย จะดูที่คำกำหนดของระบบ

หมายเหตุ: โปรไฟล์ผู้ใช้มีแอ็ดทริบิวต์งาน ที่ช่วยให้คุณสามารถปรับแต่งบางสิ่งเฉพาะสำหรับผู้ใช้รายนั้น

3. หลังจากที่ระบบย่อยรวบรวมแอ็ดทริบิวต์งานทั้งหมดแล้ว ระบบย่อยจะกำหนดว่า สามารถจะเริ่มงานแบบโต้ตอบใหม่ได้หรือไม่ หรือควรจะแสดงข้อความแสดงความผิดพลาดบนจอภาพ sign-on หรือไม่ ระบบย่อยจะตรวจสอบว่ามีการใช้งานถึงจำนวนสูงสุดของงานที่ อนุญาตโดยระบบย่อยหรือโดย workstation entry แล้วหรือไม่ จากนั้น ระบบย่อยจะตรวจสอบ

ว่ามีการป้อนชื่อโปรไฟล์ผู้ใช้ที่ถูกต้อง ชื่อโปรไฟล์ผู้ใช้เป็น โปรไฟล์ผู้ใช้ที่ใช้ได้ และตรวจสอบว่ารหัสผ่านที่ใช้ (ถ้าต้องการ) ถูกต้องหรือไม่ ถัดไป จะตรวจสอบว่าผู้ใช้มีสิทธิที่ถูกต้อง ที่รายละเอียดของงาน, subsystem description, workstation device description และเอาต์พุตคิวและโลบารรีหรือไม่ ท้ายสุด ระบบย่อยจะตรวจสอบว่าผู้ใช้ได้ใช้งานถึงขีดจำกัดสำหรับ sign-on ที่อนุญาตสำหรับโปรไฟล์ใช้นั้นหรือไม่ ถ้าพบข้อผิดพลาดในการตรวจสอบใดๆ จอภาพ sign-on จะแสดงขึ้นพร้อมกับข้อความที่เหมาะสม มิฉะนั้น กระบวนการของการเริ่มงานแบบโต้ตอบ จะดำเนินต่อไป

4. หลังจากทีระบบย่อยตรวจสอบว่างานแบบโต้ตอบสามารถเริ่มได้แล้ว ระบบย่อย จะตรวจสอบรายละเอียดของงานสำหรับข้อมูลการเราต์ ระบบย่อยใช้ข้อมูลการเราต์ ในการค้นหา routing entry ใน subsystem description routing entry นำเสนอข้อมูลพูลที่งานจะใช้, โปรแกรมเราต์ที่จะใช้ และคลาสที่งานจะเรียกแอ็ดทริบิวต์รันใหม่ของงานมา
5. เมื่อได้ข้อมูลเหล่านี้ทั้งหมดแล้ว โปรแกรมการเราต์จะรัน IBM จัดเตรียมโปรแกรมการเราต์ที่มีชื่อว่า QCMD ซึ่งคุณสามารถใช้สำหรับงานได้ทุกชนิด QCMD ทราบว่างานเป็นงานแบบโต้ตอบหรือไม่ และตรวจสอบโปรไฟล์ผู้ใช้สำหรับ initial program ที่จะรัน ถ้า initial program รันเสร็จสิ้นแล้ว QCMD จะแสดง เมนูเริ่มต้น

งานที่เกี่ยวข้อง

“การหลีกเลี่ยงฟังก์ชันที่ใช้เวลารันนานจากเวิร์กสเตชัน” ในหน้า 137

เพื่อหลีกเลี่ยงฟังก์ชันที่ใช้เวลารันนาน (เช่น บันทึก/คินสภาพ) จาก เวิร์กสเตชันโดยไม่ทำให้ระบบชะงัก ผู้ควบคุมระบบสามารถส่งงานไปที่ คิวงาน

การตัดการเชื่อมต่องานแบบโต้ตอบ:

เมื่อใช้คำสั่ง Disconnect Job (DSCJOB) งานจะถูกตัดการเชื่อมต่อ และจอแสดงผล sign-on จะแสดงขึ้นอีกครั้งหนึ่ง ถ้าต้องการ เชื่อมต่อกับงานอีกครั้ง ให้ sign on ที่อุปกรณ์เดิมซึ่งคุณตัดการเชื่อมต่องานแบบโต้ตอบอื่นอาจถูกเริ่มบนอุปกรณ์ภายใต้ชื่อผู้ใช้ที่แตกต่างอื่น

- อีอพชันบนเมนู System Request ช่วยให้คุณสามารถตัดการเชื่อมต่องานแบบโต้ตอบ ซึ่งทำให้จอแสดงผล sign-on ปรากฏขึ้น อีอพชันจะเรียกคำสั่ง DSCJOB
- เมื่อเชื่อมต่อกับงานอีกครั้ง ค่าที่ระบุบนจอ sign-on สำหรับโปรแกรม, เมนู และโลบารรีปัจจุบันจะถูกข้ามไป
- งานซึ่งมีฟังก์ชัน PC organizer หรือ PC text assist แอ็คทีฟอยู่ ไม่สามารถตัด การเชื่อมต่อได้
- งาน TCP/IP TELNET สามารถตัดการเชื่อมต่อได้เฉพาะถ้าเซสชันกำลังใช้ device description ที่ผู้ใช้ระบุชื่อเท่านั้น คุณสามารถสร้าง device description ที่ผู้ใช้ระบุชื่อโดยใช้วิธีใดวิธีหนึ่งดังนี้:
 - การใช้ Network Stations พร้อมกับพารามิเตอร์ DISPLAY NAME
 - การใช้ System i Access PC 5250 Client Access support พร้อมฟังก์ชัน workstation ID
 - การใช้ TCP/IP TELNET Device Initialization exit point ในการระบุ ชื่อเวิร์กสเตชัน
 - Telnet client (STRTCPTELN) พร้อมกับพารามิเตอร์อุปกรณ์รีโมต

หมายเหตุ: ชื่ออุปกรณ์ที่ระบุระบบ เช่น QPADEV* ไม่นุญาตให้ตัดการเชื่อมต่องาน เนื่องจากเป็นไปไม่ได้ที่ผู้ใช้รายเดียวกันจะ sign on ที่อุปกรณ์เดียวกัน

- งานทั้งหมดจะถูกตัดการเชื่อมต่อสำหรับงานกลุ่ม เมื่องานถูกเชื่อมต่ออีกครั้ง คุณจะกลับไปที่มีการตัดการเชื่อมต่อ ถ้างานกลุ่มหลังสุดที่ แอ็คทีฟจบลงก่อนที่คุณจะเชื่อมต่ออีกครั้ง คุณจะกลับไปทำงานกลุ่มถัดไป
- ถ้าไม่สามารถตัดการเชื่อมต่องานเนื่องจากเหตุผลใดๆ, งานจะถูกจบแทน
- งานที่ตัดการเชื่อมต่อทั้งหมดในระบบย่อยจะจบลง เมื่อระบบย่อยสิ้นสุดลง ถ้าระบบย่อยกำลังสิ้นสุด จะไม่สามารถใช้คำสั่ง DSCJOB ในงานใดๆในระบบย่อย

- สามารถใช้ค่ากำหนดของระบบ Disconnect Job Interval (QDSCJOBITV) เพื่อระบุช่วงเวลาที่สามารถตัดการเชื่อมต่องานได้ ถ้าถึงช่วงเวลานั้น งานที่ตัดการเชื่อมต่อจะจบลง
- งานที่ตัดการเชื่อมต่อที่ไม่เกินกว่าค่า QDSCJOBITV จะจบลงเมื่อระบบย่อยจบลง หรือเมื่อเกิด IPL

หลักการที่เกี่ยวข้อง

“ข้อควรพิจารณาการตัดการเชื่อมต่องาน” ในหน้า 136
มีหลายปัจจัยที่คุณต้องพิจารณาในทุกครั้งที่คุณตัดการเชื่อมต่องาน

งานที่เกี่ยวข้อง

“การจบงานแบบโต้ตอบ” ในหน้า 135

คุณสามารถใช้วิธีต่างๆหลายแบบในการจบงานแบบโต้ตอบ

“การตัดการเชื่อมต่องานทั้งหมดจากอุปกรณ์” ในหน้า 136

คำสั่ง Disconnect Job (DSCJOB) อนุญาตให้ผู้ใช้แบบโต้ตอบตัดการเชื่อมต่อกับงานแบบโต้ตอบทั้งหมดที่เวิร์กสเตชันและกลับไปที่จะแสดงผล sign-on โลงนที่สลับจะถูกยกเลิกเฉพาะถ้ามีการระบุใน workstation device description ของเวิร์กสเตชันนี้ และถ้า ไม่มีเวิร์กสเตชันอื่นบนโลงนนี้แอ็คทีฟ ถ้างานถูกตัดการเชื่อมต่อเมื่อใช้งานถึง ช่วงตัดการเชื่อมต่อในค่ากำหนดของระบบ Time-out interval for disconnected jobs (QDSCJOBITV) งานจะสิ้นสุดลงและไม่มีการรวมบันทึกการใช้งาน กับสพูลเอาต์พุตของงาน

ข้อผิดพลาด I/O สำหรับอุปกรณ์ที่ร้องขอของงาน:

อุปกรณ์ที่ร้องขอคือ เวิร์กสเตชันซึ่งผู้ใช้สามารถล็อกออน ที่โดเมนและใช้เน็ตเวิร์กหรือรีส แอ็คทีฟิวิตงาน Device Recovery Action (DEVRCYACN) ระบุ action ที่จะใช้เมื่อเกิดข้อผิดพลาด I/O สำหรับอุปกรณ์ที่ร้องขอของงาน

แอ็คทีฟิวิต DEVRCYACN มีอ็อปชันดังต่อไปนี้:

*SYSVAL

นี่เป็นค่าดีฟอลต์ ค่านี้จะชี้ไปที่ Action ที่จะใช้เมื่อเกิดข้อผิดพลาดอุปกรณ์ บนค่ากำหนดของระบบ workstation (QDEVRCYACN) ค่ากำหนดของระบบจะสนับสนุนค่าทั้งหมดที่แอ็คทีฟิวิตงานสนับสนุน (ยกเว้น *SYSVAL)

*MSG ส่งสัญญาณข้อความแสดงความผิดพลาด I/O และอนุญาตให้แอ็พพลิเคชัน โปรแกรมทำการแก้ไขข้อผิดพลาด นี้ไม่ใช้การตั้งค่าที่แนะนำ

*DSCMSG

ตัดการเชื่อมต่องาน นี่เป็นค่าดีฟอลต์ที่จัดส่งมาให้ ในทันทีที่เชื่อมต่ออีกครั้ง ข้อความแสดงความผิดพลาดใหม่จะส่งสัญญาณให้แอ็พพลิเคชันโปรแกรมของผู้ใช้ ทราบว่าอุปกรณ์หายไปแล้วและถูกกู้คืนนับตั้งแต่ I/O และต้องแสดงเนื้อหาของ จอแสดงผลอีกครั้ง

*DSCENDRQS

ตัดการเชื่อมต่องาน ในทันทีที่เชื่อมต่ออีกครั้ง จะมีการปฏิบัติฟังก์ชันคำขอการจบ เพื่อให้การควบคุมของงานกลับไปเป็นระดับคำขอหลังสุด

*ENDJOB

จบงาน อาจมีการจัดทำบันทึกการใช้งานสำหรับงาน ข้อความจะถูกส่งไปที่ บันทึกการใช้งานและไปที่ไฟล์บันทึก QHST โดยระบุว่างานจบลงเนื่องจาก ข้อผิดพลาดอุปกรณ์

*ENDJOBNO LIST

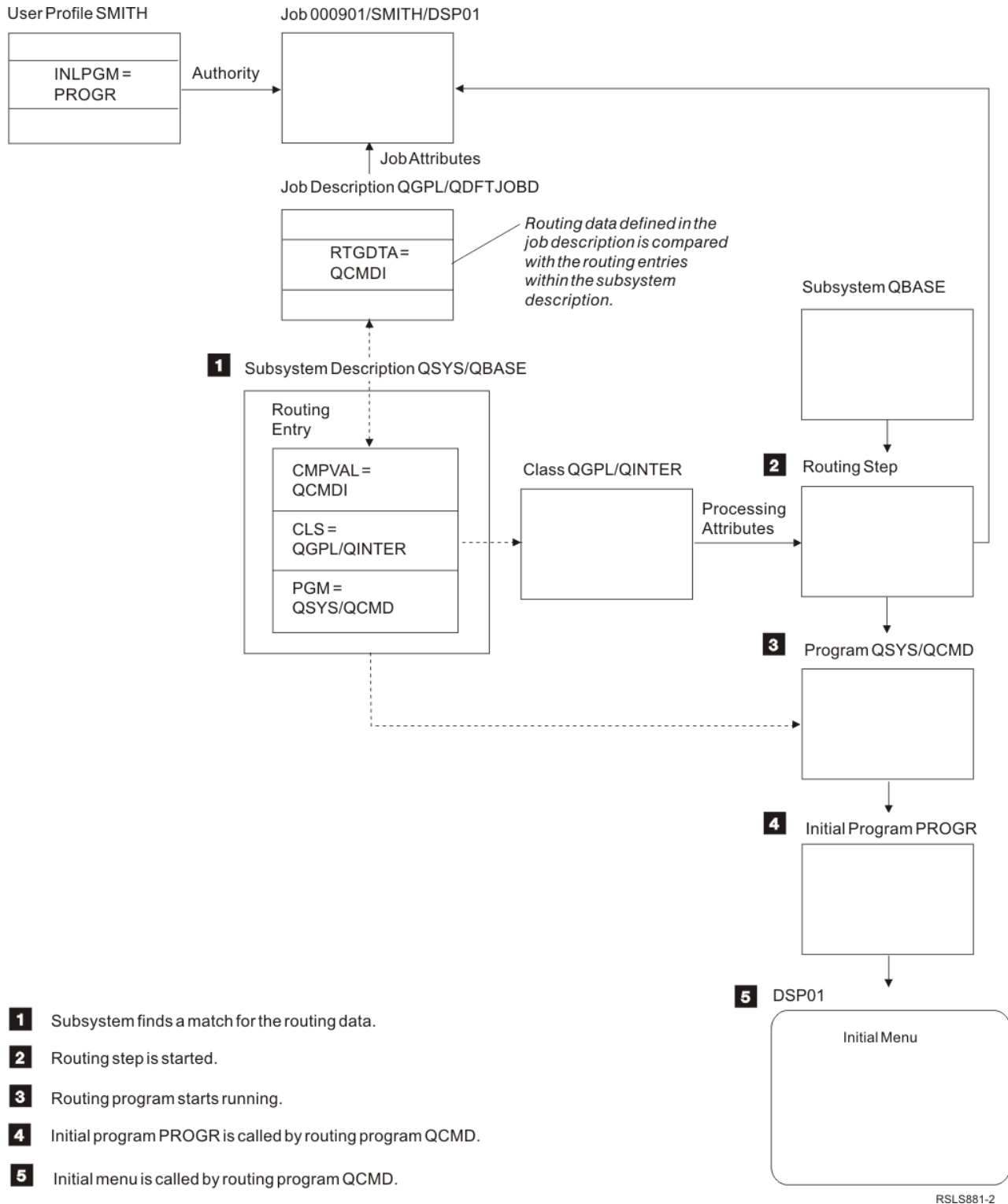
จบงาน ไม่มีการจัดทำบันทึกการใช้งาน ข้อความจะถูกส่งไปที่ ไฟล์บันทึก QHST เพื่อระบุว่างานจบลงเนื่องจาก ข้อผิดพลาดอุปกรณ์

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

งานแบบโต้ตอบและขั้นตอนการเรอต์:

ก่อนการเรียกเมนูเริ่มต้น ข้อมูลการเรอต์จะถูกเปรียบเทียบ กับ routing entries ใน subsystem description เมื่อจับคู่ได้ โปรแกรมที่ระบุใน routing entry จะถูกเรียกและขั้นตอนการเรอต์ถูก เริ่มต้น

ข้อมูลต่อไปนี้แสดง activity ก่อนหน้าที่จะนำไปสู่การเริ่มขั้นตอนการเรอต์ และการแสดงเมนูเริ่มต้นสำหรับโปรไฟล์ผู้ใช้ที่ระบุ initial program



- 1** Subsystem finds a match for the routing data.
- 2** Routing step is started.
- 3** Routing program starts running.
- 4** Initial program PROGR is called by routing program QCMD.
- 5** Initial menu is called by routing program QCMD.

รูปที่ 1. Subsystem Activity

Interactive Job Approaches

คุณสามารถจัดการงานแบบโต้ตอบได้หลายวิธี วิธีเหล่านี้ขึ้นอยู่กับว่าคุณควบคุมขั้นตอนการเราต์อย่างไรในอันดับแรก คุณควรกำหนดสิ่งดังต่อไปนี้:

- 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
- ผู้ใช้จะถูกวางในสภาวะแวดล้อม System/36 เมื่อมีการระบุในโปรไฟล์ผู้ใช้

ยิ่งไปกว่านั้น, คำดีฟอลต์ที่ใช้ QCMD จะนำคุณไปที่ Main Menu ที่ซึ่ง คุณสามารถป้อนคำสั่งได้โดยตรง รวมถึงคำสั่ง CALL ซึ่งถูกใช้ในการเรียก ฟังก์ชันที่ผู้ใช้บันทึก มีการนำเสนอเมนูอ็อปชันที่มีคำอธิบายทางออนไลน์ เพื่อให้เข้าใช้งานฟังก์ชันระบบได้ง่าย และมีการนำเสนอเมนูการเลือกคำสั่ง การเข้าใช้งานการค้นหาค้นหาอย่างรวดเร็ว 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 คุณต้องกำหนดว่าจะกำหนดการเรดตามเวิร์กสเตชันซึ่งงานถูกเริ่มต้น หรือตามผู้ใช้ (โปรไฟล์ผู้ใช้) ที่ signed on

การเรดตามเวิร์กสเตชันสามารถทำได้โดยใช้ข้อมูลการเรด ที่ระบุในรายละเอียดของงานที่เชื่อมโยงกับ workstation entry หรือ โปรไฟล์สำหรับอุปกรณ์ การเรดตามผู้ใช้สามารถทำได้โดยการใช้ initial program ที่ระบุในโปรไฟล์ผู้ใช้หรือรายละเอียดของงานในโปรไฟล์ผู้ใช้ ที่แม้กับ routing entry ไม่ใช่ QCMD

การใช้ initial program

Initial program อาจโต้ตอบกับ เวิร์กสเตชันเพื่อเรียกค่าอินพุตจากผู้ใช้เวิร์กสเตชัน เมื่อเรียก initial program โปรแกรมไม่สามารถรับค่าพารามิเตอร์ สามารถใช้ initial program ในหนึ่งจากสองวิธีดังนี้:

- เพื่อสร้างสภาวะแวดล้อมแรกเริ่มสำหรับผู้ใช้อุปกรณ์คำสั่ง ตัวอย่างเช่น สามารถเปลี่ยนรายชื่อไลบรารีหรือสามารถแทนที่ค่าเดิมของ ไฟล์พิมพ์และไฟล์ข้อความ เมื่อ initial program ทำฟังก์ชันของโปรแกรมเสร็จสมบูรณ์ และกลับไป QSYS/QCMD เมนูเริ่มต้นจะถูกแสดงขึ้น
- เป็นโปรแกรมควบคุมสำหรับงาน ถ้า initial program ไม่ได้กลับไป QSYS/QCMD โปรแกรมจะกลายเป็นโปรแกรมควบคุมสำหรับขั้นตอนการเรอต์ เมนูเริ่มต้นจะไม่แสดงขึ้น ผู้ใช้สามารถร้องขอได้เฉพาะฟังก์ชันที่ พร้อมใช้งานผ่าน initial program เท่านั้น
ตัวอย่างเช่น สามารถแสดงเมนู พร้อมด้วยแอ็พพลิเคชันออฟชั่นเฉพาะ ผู้ใช้สามารถปฏิบัติได้เฉพาะฟังก์ชันบน เมนูเท่านั้น ตัวอย่างหนึ่งของออฟชั่นนั้นคือ sign off ถ้ารันคำสั่ง SIGNOFF งานจะจบและไม่มีการแสดง Main Menu ถ้าคุณใช้วิธีนี้ให้พิจารณาการใช้ออฟชั่นโปรไฟล์ผู้ใช้ INLMNU เพื่อให้มั่นใจว่าไม่มีการแสดงเมนูใด

initial program สามารถถูกบันทึกได้เพื่อให้เมื่อส่งคืน โปรแกรม จะหรือจะไม่กลับไป QSYS/QCMD อย่างใดอย่างหนึ่ง ถ้า initial program กลับไปที่ QSYS/QCMD เมนูเริ่มต้นจะถูกแสดงขึ้น

เมื่องานจบลงพร้อมกัน:

ในบางครั้ง งานหลายงานอาจจบลงในเวลาเดียวกัน ตัวอย่างเช่น เมื่อเกิดข้อผิดพลาด เน็ตเวิร์ก และแอ็ททริบิวต์งานถูกเซ็ทเป็น *ENDJOB หรือ *ENDJOBNO LIST นอกเหนือจาก การจบของงานแล้ว อาจเกิดการแก้ไขอุปกรณ์ดังต่อไปนี้

- ระดับความสำคัญของงานถูกลดลง สิ่งนี้จะเกิดขึ้นเมื่องานไม่ได้มีระดับความสำคัญเหมือน กับงานแบบโต้ตอบที่แอ็ททิฟอื่นอีกต่อไป
- การแบ่งเวลาของงานถูกเซ็ทเป็น 100 มิลลิวินาที สิ่งนี้เกิดขึ้นเพื่อให้งานที่มีระดับความสำคัญ สูงกว่ามีโอกาสที่ดีกว่าในการใช้รีซอร์สในการประมวลผล

บันทึกการใช้งานสำหรับงานที่มีแอ็ททริบิวต์งานถูกเซ็ทเป็น *ENDJOB หรือ *ENDJOBNO LIST อยู่ใน บันทึกการใช้งานที่คงค้าง เพื่อสร้างพรินเตอร์เอาต์พุตจากบันทึกการใช้งานที่อยู่ในบันทึกการใช้งานที่คงค้าง ให้ใช้คำสั่ง Display Job Log (DSPJOBLOG)

เมื่องานจบ คุณสามารถควบคุมวิธีการบันทึกบันทึกการใช้งานไปที่สพูลไฟล์ได้ สิ่งนี้สามารถทำโดยตัวงานเองในขณะที่กำลังจบ ทำโดยงานเซิร์ฟเวอร์ แบ็คกราวด์ หรือไม่ทำเลย ค่าที่คุณระบุส่งผลกระทบต่อเวลา การแก้ไขโดยรวม ถ้างานหลายงานจบลงพร้อมกัน สำหรับข้อมูล เพิ่มเติม ให้ดูหลักการ บันทึกการใช้งานที่คงค้าง ที่เกี่ยวข้อง

หลักการที่เกี่ยวข้อง

“การคงค้างบันทึกการใช้งาน” ในหน้า 86

สภาพการคงค้างบันทึกการใช้งานมีใช้กันเป็นเวลาหลายปีแล้ว เมื่อแอ็ททริบิวต์บันทึกการใช้งานของงานเป็น *PND จะไม่มีการสร้างบันทึกการใช้งาน คุณสามารถควบคุมวิธีการ และสถานการณ์ที่มีการสร้างบันทึกการใช้งานสำหรับงานเฉพาะได้

งานแบบ prestart:

งานแบบ prestart คืองานแบ็คซ์ที่เริ่มต้นก่อนที่จะได้รับคำขอ งานแบบ prestart ถูก เริ่มต้นก่อนงานชนิดอื่นๆ ในระบบย่อย งานแบบ prestart แตกต่างจากงานอื่นๆ เนื่องจาก การใช้ prestart job entry (ส่วนของ subsystem description) ในการกำหนดโปรแกรม, คลาส และพูลของหน่วยความจำที่จะใช้เมื่องานถูกเริ่มต้น

ภายใน prestart job entry คุณต้องระบุแอตทริบิวต์ที่ระบบย่อย จะใช้ในการสร้างและจัดการพูลของงานแบบ prestart ใช้งานแบบ prestart เพื่อลดจำนวนเวลาที่ต้องใช้ในการจัดการคำขอ งานแบบ prestart มีอยู่ 2 ชนิด แต่ละชนิดจะจัดการคำขอต่างชนิดกัน ก่อนที่งานจะรอสำหรับคำขอแรกของงานนั้น งานจะถูกแสดงว่าเป็น "Prestart" เท่านั้น เพราะวาระบบจะยังไม่รู้ว่า คำขอชนิดใดที่งานนั้นจะจัดการ

การสื่อสารแบบ prestart

งานคืองานแบ็คการสื่อสารที่เริ่มต้นก่อนที่ระบบริโมต จะส่งคำขอการเริ่มโปรแกรม

แบ็คแบบ prestart

งานคืองานแบ็คที่เริ่มต้นก่อนที่จะได้รับคำขอของงาน

งานแบบ prestart จะเริ่มขึ้นก่อนที่จะได้รับคำขอการทำงาน ไม่ว่าจะเป็น ตอนที่ระบบย่อยเริ่มขึ้น หรือที่เป็นผลมาจากคำสั่ง Start Prestart Jobs (STRPJ) อย่างไรก็ตาม งานแบบ prestart จะเริ่มทำงานจาก prestart job entry (PJE) ใน subsystem description prestart job entry ระบุแอตทริบิวต์ต่างๆ โปรแกรมที่จะรันใน งานแบบ prestart, โปรไฟล์ผู้ใช้ที่งานแบบ prestart จะเริ่มต้นรันภายใต้ นั้น รายละเอียดของงาน, คลาสที่ใช้ในการระบุแอตทริบิวต์รันใหม่ของ งาน และพูลหน่วยความจำ ซึ่งงานแบบ prestart จะรันในนั้น

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

หมายเหตุ: ค่าที่ระบุสำหรับจำนวนสูงสุดของงานในระบบย่อย อาจทำให้งานแบบ prestart ไม่สามารถเริ่มขึ้นได้ หากจำนวนงานสูงสุดในระบบย่อยมีจำนวนมากเกิน งานแบบ prestart จะไม่สามารถเริ่มทำงานได้ แต่เมื่อมีจำนวนหนึ่งได้เสร็จสิ้นลงจนทำให้จำนวนงานที่รันอยู่ ต่ำกว่าจำนวนงานสูงสุดในระบบย่อยแล้ว งานแบบ prestart ในระบบย่อย จึงจะสามารถทำงานได้

คำขอเริ่มโปรแกรม

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

นอกจากนี้ยังสามารถใช้งานแบบ prestart สำหรับเซิร์ฟเวอร์ TCP/IP ที่ได้มาจาก IBM, และไฮสปีดเซิร์ฟเวอร์ส่วนใหญ่ งานแบบ prestart เหล่านี้รับงานผ่านทางอินเทอร์เน็ตเฟสภายในและไม่ได้ใช้ PSR อย่างไรก็ตาม งานแบบ prestart ที่กำลังรอกงานอยู่แม้ว่าจะไม่ได้ใช้ PSR ยังคงแสดงสถานะเป็น PSRW

หลักการที่เกี่ยวข้อง

“งานแบบ prestart และ job accounting” ในหน้า 101

ถ้าระบบของคุณใช้ job accounting โปรแกรมงานแบบ prestart ควรจะรันคำสั่ง Change Prestart Job (CHGPJ) ด้วยคำขอร้องการเริ่มโปรแกรมสำหรับพารามิเตอร์ accounting code parameter (CHGPJ ACGCDE(*PGMSTRRQS)) โดยทันทีหลังจากที่คำขอการเริ่มโปรแกรมแนบกับงานแบบ prestart

งานที่เกี่ยวข้อง

“การเริ่มงานแบบ prestart” ในหน้า 138

โดยปกติ งานแบบ prestart จะเริ่มในเวลาเดียวกับที่ระบบย่อยเริ่มขึ้น คุณเริ่มงานแบบ prestart ด้วยตนเอง เมื่องานแบบ

prestart ทั้งหมดได้ถูกจบ โดยระบบเนื่องจากข้อผิดพลาดหรืองานไม่ถูกเริ่มขึ้นในระหว่างที่ระบบย่อย เริ่มขึ้นเนื่องจาก STRJOBS (*NO) บน prestart job entry ถ้าต้องการเริ่ม งานแบบ prestart ให้ใช้อินเตอร์เฟสแบบอักษร

“การจบงานแบบ prestart job” ในหน้า 144

คุณสามารถใช้อินเตอร์เฟสแบบอักษร เพื่อจบงานแบบ prestart ในระบบย่อยที่แฉีกที่พอยู่

ข้อมูลที่เกี่ยวข้อง

รายงานจากประสบการณ์: การปรับ prestart job entries

ชื่องานแบบ prestart:

ชื่อสามส่วนที่ถูกต้องทั้งหมดของงานแบบ prestart ไม่เคยเปลี่ยนแปลง หลังจากทำงานแบบ prestart เริ่มขึ้น ส่วนชื่อผู้ใช้ของชื่อ งานสามส่วนที่ถูกต้องทั้งหมดคือ โปรไฟล์ผู้ใช้ซึ่งงานแบบ prestart ถูกเริ่มต้นภายใต้มันเสมอ

ถ้าสพูลไฟล์ถูกเปิดก่อนงานแบบ prestart จะจัดการคำขอของงานใดๆ สพูลไฟล์จะถูกเชื่อมโยงกับโปรไฟล์ผู้ใช้ของ prestart job entry มิฉะนั้น สพูลไฟล์จะถูกเชื่อมโยงกับโปรไฟล์ผู้ใช้ปัจจุบันของงาน

ถ้าโปรไฟล์ prestart job entry และโปรไฟล์ผู้ใช้ปัจจุบันแตกต่างกัน สพูลไฟล์จะถูกสพูลภายใต้งานที่มีชื่องานเป็น QPRTJOB และมีชื่อผู้ใช้เป็นโปรไฟล์ผู้ใช้ปัจจุบัน (และใช้กับ prestart job entries สำหรับงานเซิร์ฟเวอร์ด้วย)

พารามิเตอร์คลาส (CLS) บน prestart job entry นำเสนอวิธีการควบคุม ลักษณะประสิทธิภาพของสองคลาสของงานแบบ prestart ต่อ prestart job entry

วิธีการทำงานของงานแบบ prestart:

งานแบบ prestart คืองานที่ถูกเริ่มก่อนที่งานจะมาถึง การเริ่มก่อนนี้ช่วยให้ระบบสามารถจัดการคำขอสำหรับงานได้โดยไม่ต้อง เสียเวลากับ การเริ่มต้นงานใหม่

งานแบบ prestart เป็นงานแบ็คชิ่งชนิดหนึ่ง นี่หมายความว่างานมี ชนิดของงานเป็น 'B' และชนิดย่อยของงานเป็น 'J' ชนิดงานที่ พัฒนา กำหนดงานเป็นงานแบบ prestart (1610), งานแบ็คชิ่งแบบ prestart (1620), หรืองานสื่อสารแบบ prestart (1630) ชนิดงานที่พัฒนาอธิบายวิธีการที่งานแบบ prestart ยอมรับคำขอของงาน ถ้าโปรแกรมที่รันอยู่ในงานแบบ prestart ใช้ อินเตอร์เฟส การสื่อสารสำหรับการยอมรับงาน, งานจะเป็นงานสื่อสารแบบ prestart ถ้าโปรแกรมที่รันอยู่ในงานแบบ prestart ยอมรับงาน ผ่านอินเตอร์เฟสงานแบ็คชิ่ง งานจะเป็นงานแบ็คชิ่งแบบ prestart ถ้าโปรแกรมยังไม่ถึงจุด ของการยอมรับงาน งานจะเป็นเพียง งานแบบ prestart งานแบ็คชิ่งแบบ Prestart มักถูกอ้างอิงเป็นงานเซิร์ฟเวอร์ เนื่องจากงานนำเสนอการให้บริการ สำหรับคำขอ งาน

คำขอของการสื่อสารถูกจัดการโดยระบบย่อยที่มีการจัดสรร อุปกรณ์การสื่อสารที่จำเป็น โดยปกติคำขอของงานแบ็คชิ่ง ถูกจัดการ โดยระบบย่อยระดับต้นระบบใดระบบหนึ่งที่ถูกจัดส่งมาพร้อมกับระบบ: QSYSWRK, QUSRWRK หรือ QSERVER

งานแบบ prestart ถูกเริ่มตามข้อมูลที่มีอยู่ใน prestart job entries พารามิเตอร์ Start jobs (STRJOBS) ของคำสั่ง Add Prestart Job Entry (ADDPJE) และ Change Prestart Job Entry (CHGPJE) สามารถระบุว่าการแบบ prestart ถูกเริ่มขึ้นเมื่อระบบย่อย เริ่มต้น หรือเมื่อป้อนคำสั่ง Start Prestart Jobs (STRPJ) พารามิเตอร์ Initial number of jobs (INLJOBS) กำหนดจำนวนของ งานแบบ prestart ที่เริ่มต้นในครั้งแรกสำหรับโปรแกรม

เมื่อคำขอของงานมาถึง อาจต้องการงานแบบ prestart มากขึ้น พารามิเตอร์ Threshold (THRESHOLD) ของคำสั่ง Add Prestart Job Entry (ADDPJE) และ Change Prestart Job Entry (CHGPJE) บอกให้ทราบว่าจะเริ่มงานมากขึ้นเมื่อใด เมื่อจำนวนของ

งานแบบ prestart ที่พร้อมใช้สำหรับการจัดการคำขอลดลงต่ำกว่าค่าที่ระบุโดยพารามิเตอร์ THRESHOLD งานเพิ่มเติมจะถูกเริ่มขึ้น พารามิเตอร์ Additional number of jobs (ADLJOBS) บอกรับว่าจะเริ่มงานเพิ่มขึ้นจำนวนเท่าใด

งานแบบ prestart บางงานจัดการคำขอแล้วจึงพร้อมที่จะจัดการ คำขออื่น พารามิเตอร์ Maximum number of uses (MAXUSE) ช่วยให้คุณสามารถระบุจำนวนของคำขอที่งานแบบ prestart เหล่านี้จัดการได้ งานแบบ prestart บางงานจัดการคำขอเพียงคำขอเดียวแล้วจึงจบ โดยข้ามค่า MAXUSE งานแบบ prestart จะจัดการคำขอหลายคำขอ หรือจัดการคำขอเพียงคำขอเดียวขึ้นอยู่กับโปรแกรม ที่รันอยู่ในงานแบบ prestart

เมื่องานแบบ prestart จบลงหลังจากที่จัดการคำขออย่างน้อยหนึ่งคำขอ ระบบย่อยจะเปรียบเทียบจำนวนของงานที่ยังคงรันอยู่กับจำนวนที่ระบุในพารามิเตอร์ INLJOBS ถ้าจำนวนของงานที่เหลืออยู่น้อยกว่า INLJOBS ระบบย่อยจะเริ่มงานอื่น

ถ้างานแบบ prestart จบลงโดยไม่ได้จัดการขออน้อยหนึ่งคำขอ และงานไม่จบลงโดยคำสั่ง End Job (ENDJOB) ถือว่าโปรแกรมงานแบบ prestart มีข้อผิดพลาด ระบบย่อยจบ prestart job entry ในลักษณะที่ควบคุม สิ่งนี้ช่วยให้งานที่ใช้ใน คำขอ งานสามารถทำคำขอนั้นให้เสร็จสิ้นได้, แต่ป้องกันไม่ให้ระบบย่อยเริ่ม งานเพิ่มเติม

ระบบย่อยจะตรวจสอบจำนวนของงานแบบ prestart เป็นครั้งคราวเพื่อกำหนดว่า มีงานแบบ prestart ที่พร้อมใช้มากเกินไปหรือไม่ งานแบบ prestart จะพร้อมใช้งาน เมื่องานกำลังรอสำหรับคำขอ

ข้อมูลที่เกี่ยวข้อง

รายงานจากประสบการณ์: การปรับ prestart job entry

รายงานจากประสบการณ์: การคอนฟิกูเรชันระบบย่อย

Prestart job entry:

คุณกำหนดงานแบบ prestart โดยใช้ prestart job entry prestart job entry ไม่มีผลต่อการจัดสรรอุปกรณ์หรือการกำหนดคำขอ การเริ่มโปรแกรม

แอตทริบิวต์งานของงานแบบ prestart จะไม่ถูกเปลี่ยนแปลงโดยระบบย่อย เมื่อ คำขอการเริ่มโปรแกรมแนบกับงานแบบ prestart อย่างไรก็ตาม โดยทั่วไป งานเซิร์ฟเวอร์ จะเปลี่ยนแอตทริบิวต์งานเป็นแอตทริบิวต์ของโปรไฟล์ผู้ใช้ที่สลับค่า

คำสั่ง Change Prestart Job (CHGPJ) อนุญาตให้งานแบบ prestart เปลี่ยนแอตทริบิวต์งานบางอย่างเป็นแอตทริบิวต์ของรายละเอียดของงาน (ที่ระบุในรายละเอียดของงานที่เชื่อมโยงกับโปรไฟล์ของผู้ใช้ของคำขอเริ่มโปรแกรม หรือในรายละเอียดของงานที่ระบุใน prestart job entry)

หลักการที่เกี่ยวข้อง

“การสืบสวนงานแบบ prestart” ในหน้า 244

หัวข้อนี้นำเสนอขั้นตอนที่ช่วยให้คุณในการตอบคำถาม “ฉันจะ ค้นหาผู้ใช้จริงของงานแบบ prestart และกำหนดรีซอร์สที่ใช้ โดย งานแบบ prestart นั้นได้อย่างไร?”

งานที่เกี่ยวข้อง

“การใส่เพิ่ม prestart job entries” ในหน้า 183

Prestart job entries ระบุงานแบบ prestart ที่สามารถเริ่มต้น เมื่อระบบย่อยเริ่มต้น หรือเมื่อป้อนคำสั่ง Start Prestart Jobs (STRPJ) คุณสามารถใส่เพิ่ม prestart job entries ที่ subsystem description โดยใช้อินเตอร์เฟซแบบอักษร

“การเปลี่ยน prestart entry” ในหน้า 188

คุณสามารถเปลี่ยน prestart job entry ใน subsystem description ที่ระบุได้ ระบบย่อยอาจเฝ้าที่พอย์เมื่อ prestart job entry

ถูก เปลี่ยน การทำการเปลี่ยนแปลงที่ entry เมื่อระบบย่อยแอ็ดคทีฟจะมีผลต่อการดำเนิน การในอนาคต งานแบบ prestart ใหม่ใดๆ ที่เริ่มขึ้นหลังจากใช้คำสั่ง จะใช้ ค่าที่เกี่ยวข้องกับงานใหม่ คำสั่งนี้ระบุงานแบบ prestart ที่เริ่มขึ้นเมื่อระบบย่อยเริ่มต้น หรือเมื่อมีการใช้คำสั่ง Start Prestart Jobs (STRPJ)

“การลบ prestart job entry” ในหน้า 192

คุณสามารถลบ prestart job entries ออกจาก subsystem description โดยใช้อินเตอร์เฟสแบบอักขระ จะไม่สามารถลบ prestart job entry ถ้างานแอ็ดคทีฟในปัจจุบันใดๆ ถูกเริ่มโดยใช้ entry

ข้อมูลที่เกี่ยวข้อง

รายงานจากประสบการณ์: การปรับ prestart job entry

คำขอการเริ่มโปรแกรมการจัดการงานแบบ prestart:

เมื่องานแบบ prestart เริ่มต้น งานจะรันภายใต้โปรไฟล์ผู้ใช้ของ งานแบบ prestart เมื่อคำขอการเริ่มโปรแกรมแนบกับงานแบบ prestart โปรไฟล์ผู้ใช้ของงานแบบ prestart จะถูกแทนที่โดยโปรไฟล์ผู้ใช้ของคำขอการเริ่มโปรแกรม เมื่องานแบบ prestart จัดการกับคำขอการเริ่มโปรแกรมเสร็จสิ้นแล้ว โปรไฟล์ผู้ใช้ของ คำขอการเริ่มโปรแกรมจะถูกแทนที่ด้วยโปรไฟล์ผู้ใช้ของงานแบบ prestart หากมีโปรไฟล์กลุ่มที่เชื่อมโยงกับโปรไฟล์ผู้ใช้โปรไฟล์กลุ่มจะถูกเปลี่ยนด้วย

โปรไฟล์ผู้ใช้ที่ถูกแลกเปลี่ยนใช้สำหรับการตรวจสอบสิทธิเท่านั้น ไม่มีการแลกเปลี่ยนแอ็ดทริบิวต์อื่นที่เชื่อมโยงกับโปรไฟล์ผู้ใช้สำหรับไลบรารีต่างๆ บนรายชื่อไลบรารีซึ่งโปรไฟล์ผู้ใช้ของ prestart job entry ได้รับสิทธิในการใช้งานแบบ prestart ยังคงได้รับสิทธิในไลบรารีนั้นต่อไปเมื่อโปรไฟล์ผู้ใช้ของคำขอ การเริ่มโปรแกรมแทนที่โปรไฟล์ผู้ใช้ของ prestart job entry อย่างไรก็ตาม รายชื่อไลบรารี อาจถูกเปลี่ยนแปลงโดยคำสั่ง Change Library List (CHGLIBL)

การให้สิทธิอ็อบเจ็กต์งานแบบ prestart สำหรับคำขอการเริ่มโปรแกรม

เมื่องานแบบ prestart เริ่มขึ้น จะมีการตรวจสอบสิทธิในโปรไฟล์ผู้ใช้ของ prestart job entry บนทุกอ็อบเจ็กต์ที่จำเป็นสำหรับการเริ่มงาน ก่อนที่คำขอการเริ่ม โปรแกรมจะได้รับอนุญาตให้แนบงานแบบ prestart จะมีการตรวจสอบเฉพาะโปรไฟล์ผู้ใช้ /รหัสผ่านของคำขอการเริ่มโปรแกรมและสิทธิในการใช้งานอุปกรณ์สื่อสารและ ไลบรารี/โปรแกรมเท่านั้น

เพื่อหลีกเลี่ยงกรณีที่โปรไฟล์ผู้ใช้ของคำขอการเริ่ม โปรแกรมไม่ได้รับสิทธิในอ็อบเจ็กต์ที่โปรไฟล์ผู้ใช้ของ prestart job entry ได้รับสิทธิ คุณต้องตรวจสอบให้แน่ใจว่าโปรไฟล์ผู้ใช้ของ คำขอการเริ่มโปรแกรมได้รับสิทธิในอ็อบเจ็กต์เท่ากับสิทธิของโปรไฟล์ผู้ใช้ของ prestart job entry เป็นอย่างน้อย เพื่อให้เป็นเช่นนี้ สามารถสร้างโปรแกรมงานแบบ prestart โดยผู้ใช้ prestart job entry ที่มี USRPRF(*OWNER) ระบุอยู่บน คำสั่ง CRTxxxPGM (โดยที่ xxx คือภาษาโปรแกรม) สิทธิเจ้าของโปรแกรม จะถูกถ่ายโอนโดยอัตโนมัติไปที่โปรแกรมใดๆ ที่ถูกเรียกโดย โปรแกรมงานแบบ prestart มิฉะนั้น คุณอาจเลือกที่จะตรวจสอบการให้สิทธิอ็อบเจ็กต์โดยตรง (CHKOBJ) ก่อนการอ้างอิงถึงอ็อบเจ็กต์ใดๆ

ไฟล์และอ็อบเจ็กต์ ที่โปรไฟล์ผู้ใช้ของงานแบบ prestart ไม่ได้รับสิทธิควรจะถูกปิดและ จัดสรรคืน ก่อนทำการจบ transaction บนอุปกรณ์ requestor ถ้าไฟล์ฐานข้อมูลถูกเปิดทิ้งไว้ในงานแบบ prestart เพื่อให้มั่นใจถึง ความปลอดภัยของฐานข้อมูล โปรแกรมงานแบบ prestart ต้องตรวจสอบสิทธิในการ ใช้ไฟล์เปิดของโปรไฟล์ผู้ใช้ของคำขอการเริ่มโปรแกรม

งานแบบ prestart สำหรับแอ็พพลิเคชันแบ็ตช์:

งานแบบ prestart และเซิร์ฟเวอร์ที่ใช้งานแบบ prestart แสดงถึงสถานการณ์พิเศษสำหรับ job accounting หากงานแบบ prestart เดียวให้บริการผู้ใช้ที่ต่างกัน คุณอาจต้องการให้ผู้ใช้เหล่านี้แต่ละคนสำหรับริชอร์สที่ใช่ของพวกเขา ในสถานการณ์ที่ต้องการอัปเดต accounting code ก่อน และล้างการขอบริการแต่ละครั้ง

สำหรับข้อมูลเพิ่มเติมเกี่ยวกับการเชื่อมโยงระหว่าง job accounting และงานแบบ prestart ให้ดูที่ “งานแบบ prestart และ job accounting” ในหน้า 101.

คำแนะนำเกี่ยวกับประสิทธิภาพสำหรับงานแบบ prestart:

งานแบบ prestart ควรทำงานให้มากที่สุดเท่าที่จะเป็นไปได้ ก่อนที่งานจะ พยายามจัดหาอุปกรณ์โปรแกรม ICF หรือยอมรับ CPI Communications conversation จำนวนที่ทำก่อนยิ่งมาก (การจัดสรรอ็อบเจกต์, การเปิดไฟล์ฐานข้อมูล ฯลฯ) สิ่งที่ต้องทำเมื่อได้รับคำขอเริ่มโปรแกรมจะยิ่งน้อย ดังนั้น จึงตอบสนอง transaction ได้เร็วขึ้น ข้อมูลต่อไปนี้เป็นข้อควรพิจารณาเกี่ยวกับประสิทธิภาพเพิ่มเติมเมื่อใช้งานแบบ 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

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

หมายเหตุ: โดยทั่วไป ไฟล์ฐานข้อมูลที่ปล่อยให้เปิดไว้ใน งานแบบ prestart ต้องคำนึงถึงข้อควรพิจารณาเดียวกันกับของไฟล์ฐานข้อมูลที่ถูกระบุไว้ในงานเดียวกัน

- เนื่องจากบางไลบรารี QTEMP ถูกใช้สำหรับตลอดช่วงชีวิตของงานแบบ prestart, ดังนั้นจึงควรลบอ็อบเจกต์ที่ไม่ต้องการอีกต่อไปออก
- เนื่องจาก Local Data Area (LDA) เดียวกันถูกใช้สำหรับตลอดช่วงชีวิตของงาน แบบ prestart ดังนั้นจึงสามารถเก็บและส่งผ่านข้อมูลไปยัง transaction ถัดไปได้
- เนื่องจากงานแบบ prestart แต่ละงานสามารถจัดการคำขอการเริ่มโปรแกรมได้ หลายคำขอ และมีบันทึกการใช้งานเพียงบันทึกเดียว ดังนั้นคุณจึงอาจต้องการให้ แอปพลิเคชันของคุณส่งข้อความไปที่บันทึกการใช้งานเพื่อระบุ activity ของงานแบบ prestart สิ่งนี้ยังมีประโยชน์เนื่องจากบันทึกการใช้งานของงานแบ้ตซ์แบบ prestart จะถูกลบระหว่างการใช้
- แอตทริบิวต์งานของงานแบบ prestart จะไม่ถูกเปลี่ยนแปลงโดยระบบย่อย เมื่อ คำขอการเริ่มโปรแกรมแนบกับงานแบบ prestart คำสั่ง change Prestart Job (CHGPI) อนุญาตให้งานแบบ prestart เปลี่ยนแอตทริบิวต์งานบางอย่างเป็นแอตทริบิวต์ของรายละเอียดของงาน (ที่ระบุในรายละเอียดของงานที่เชื่อมโยงกับโปรไฟล์ผู้ใช้ของคำขอการเริ่มโปรแกรม หรือในรายละเอียดของงานที่ระบุใน prestart job entry)
- พารามิเตอร์คลาส (CLS) บน prestart job entry นำเสนอวิธีการควบคุม ลักษณะประสิทธิภาพของสองคลาสของงานแบบ prestart ต่อ prestart job entry ตัวอย่างเช่น คุณสามารถใช้ระดับความสำคัญการดำเนินการที่ ต่ำกว่าสำหรับงานที่ได้รับเมื่อระบบยุ่งอยู่แล้ว

สพูลไฟล์และ prestart job entry:

ถ้าสพูลไฟล์ถูกเปิดก่อนที่งานแบบ prestart จะจัดการคำขอเริ่มโปรแกรมใดๆ สพูลไฟล์จะถูกเชื่อมโยงกับโปรไฟล์ผู้ใช้ของ prestart job entry มิฉะนั้น สพูลไฟล์จะถูกเชื่อมโยงกับโปรไฟล์ผู้ใช้ของคำขอการเริ่มโปรแกรมปัจจุบัน

ถ้าโปรไฟล์ prestart job entry และโปรไฟล์ผู้ใช้ของคำขอการเริ่มโปรแกรม ปัจจุบันแตกต่างกัน สพูลไฟล์จะถูกสพูลภายในงานที่มีส่วนแรกของชื่องานสามส่วน เป็น QPRTJOB และส่วนที่สองเป็น ชื่อของโปรไฟล์ผู้ใช้

งาน reader และงาน writer:

งานแบบ reader คืองานอินพุตที่สพูล และงานแบบ writer คืองานเอาต์พุตที่สพูล

Reader งานแบบ reader จะอ่าน streams งานแบ็ตซ์จากไฟล์ฐานข้อมูล, และวางงาน บนคิวงาน งาน reader เป็นส่วนหนึ่งของ การสพูลอินพุต และเป็นโปรแกรมที่ได้มาจาก IBM

Writer งาน writer จะเขียนเร็กคอร์ดจาก printer output files (หรือเรียกว่าสพูลไฟล์) ไปยังพรินเตอร์ งานแบบ writer คือ โปรแกรมที่ได้มาจาก IBM ที่เริ่มต้นใน ระบบย่อยการสพูลที่ซึ่งงานจะเลือกไฟล์จากเอาต์พุตคิวที่จะพิมพ์

งานเซิร์ฟเวอร์:

งานเซิร์ฟเวอร์ คือ งานที่รันอย่างต่อเนื่องในแบ็คกราวด์บนระบบของคุณ

งานอาจมาจากฟังก์ชันเน็ตเวิร์ก, ฟังก์ชันระบบปฏิบัติการ, ในนามของผู้ใช้, ระบบอื่นภายในเน็ตเวิร์ก หรือจากเซอวีร์ระบบทั่วไป เช่น งานเซิร์ฟเวอร์การรวมกลุ่มเพื่อทำงานเสมือนระบบเดียว โดยปกติ งานเซิร์ฟเวอร์จะรันในหนึ่งในสามระบบย่อยระดับต้น ที่ให้มาพร้อมกับระบบ - QSYSWRK, QSERVER หรือ QUSRWRK งานเซิร์ฟเวอร์มีความเชื่อมโยงมากที่สุดกับ ฟังก์ชันเช่น HTTP Lotus Notes และ TCP/IP ระบบของคุณมีแบบจำลองพื้นฐานสำหรับงานเซิร์ฟเวอร์สามแบบ:

Threaded Job Model

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

Prestart Job Model

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

หมายเหตุ: สำหรับงานที่รันโค้ดผู้ใช้โดยปกติงานจะไม่ถูกนำกลับมาใช้ใหม่ (เหมือนกับงานเซิร์ฟเวอร์ส่วนใหญ่) ที่เป็นเช่นนี้เนื่องจากโค้ดผู้ใช้ อาจเปลี่ยนแปลงใดๆ ในงาน (เช่น เซิร์ฟเวอร์คำสั่งรีโมต)

Multiple Listening Job Model

ในแบบจำลองงาน multiple listening มีการเริ่มงานเซิร์ฟเวอร์หลายงาน เมื่อคำขอเข้ามาในระบบ, งานที่ได้รับคำขอ จะจัดการคำขอ งาน ในขณะที่งานเซิร์ฟเวอร์ที่พร้อมใช้งานถัดไปจะคอยคำขอใหม่ที่จะ เข้ามา เมื่องานเซิร์ฟเวอร์ ดำเนินการตามคำขอเสร็จสมบูรณ์งานจะหยุดการเชื่อมต่อและจบลง งานเซิร์ฟเวอร์ใหม่จะเริ่มต้นและดำเนินการใน ลักษณะวัฏจักรนี้ต่อไป

ด้วยแบบจำลองนี้ คุณไม่ต้องกังวลกับ prestart job entries อย่างไรก็ตาม ในบางครั้งอาจไม่สามารถตั้งค่าระบบย่อย เฉพาะสำหรับสภาพแวดล้อมของคุณ เพราะแบบจำลองนี้จะรันที่ระบบย่อยดีฟอลต์ exception หนึ่งคือ File Transfer Protocol (FTP) ด้วย file transfer protocol คุณสามารถตั้งค่าระบบย่อยที่รันเซิร์ฟเวอร์ file transfer protocol ได้ไม่

สามารถมีงาน FTP บางงานรันในระบบย่อยหนึ่งและงานที่เหลือรันในระบบย่อยอื่น นอกจากนี้จากมุมมองประสิทธิภาพการทำงาน ไม่สามารถหลีกเลี่ยงต้นทุนของการเริ่มต้นทำงานและการสิ้นสุดการทำงานได้เพราะเมื่องานถูกรันและสิ้นสุดการทำงาน งานอื่นจะเริ่มต้นงาน อย่างไรก็ตาม เนื่องมาจากงานจะสิ้นสุดการทำงานเมื่อการเชื่อมต่อเสร็จสมบูรณ์ และงานถัดไปได้เริ่มขึ้น, โดยทั่วไปงานใหม่จะถูกเรียกขึ้นมาและทำการรันเมื่อได้รับคำขอถัดไป ดังนั้น ต้นทุนการเริ่มต้นใช้งานและการสิ้นสุดการทำงานไม่ควรมีผลต่อเวลาที่ใช้เชื่อมต่อกับเซิร์ฟเวอร์

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

สำหรับข้อมูลรายละเอียดเกี่ยวกับชื่องานของงาน เซิร์ฟเวอร์ที่รันบนระบบ ให้ดูตารางงานเซิร์ฟเวอร์ ตารางนี้แสดงระบบย่อยและชื่องาน เพื่อให้คุณสามารถค้นหา งานแอดทีฟและบันทึกการใช้งานของงานได้ ตารางจะแสดง job description ที่งานเซิร์ฟเวอร์แต่ละตัวใช้ด้วย ตามค่าดีฟอลต์ งานเซิร์ฟเวอร์ส่วนใหญ่จะไม่สร้างบันทึกการทำงานเมื่องานสิ้นสุดการทำงาน (พารามิเตอร์ LOG ถูกตั้งค่าเป็น 4 0 *NOLIST) ซึ่งหมายความว่าไม่มีการสร้างบันทึกการทำงาน ถ้าคุณต้องการให้มีการสร้างบันทึกการทำงานที่มีข้อความทั้งหมดที่ส่งไปที่ บันทึกการใช้งาน พารามิเตอร์ LOG ต้องระบุ 4 0 *SECLVL

ข้อมูลที่เกี่ยวข้อง

ตารางงานเซิร์ฟเวอร์

งานระบบ:

งานระบบจะถูกสร้างขึ้นโดยระบบปฏิบัติการ เพื่อควบคุมรีซอร์สของระบบและปฏิบัติฟังก์ชันของระบบ งานระบบจะรันเมื่อเซิร์ฟเวอร์เริ่มขึ้น หรือเมื่อดีสก์พูลอิสระ varied on งานเหล่านี้ปฏิบัติการกิจหลายอย่าง ตั้งแต่การเริ่มต้นระบบปฏิบัติการ, การเริ่มต้นและการจบ ระบบย่อยไปจนถึงการจัดตารางเวลางาน

งานการเริ่มทำงาน:

งานการเริ่มทำงาน คืองานระบบที่รันที่ IPL งานนี้จัดการภารกิจการจัดเตรียมสภาวะแวดล้อมระบบปฏิบัติการและทำให้พร้อมใช้งาน ข้างล่างนี้เป็นรายการของงานการเริ่ม ระบบต่างๆ

Scpf (start control program function)

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

Qwbcleanup (job table cleanup)

งานตัวนี้จะถูกใช้ระหว่างการเริ่มต้นระบบเพื่อยืนยันให้แน่ใจว่าโครงสร้างงานพร้อมใช้งานได้โดยปกติ งานจะประมวลผลเสร็จสมบูรณ์ก่อน การจบของการเริ่มระบบ แต่งานสามารถรันต่อไปได้หลังจากที่ระบบเริ่มต้น ถ้ามีโครงสร้างงานที่จะล้างอยู่มาก งานระบบตัวนี้จะจบลงเมื่อเสร็จสิ้นการประมวลผล

Qlpsvr (software agreements acceptance)

งานนี้จะเริ่มทำงานโดยอัตโนมัติระหว่างช่วง IPL หลังจากการตอบรับข้อตกลงการใช้ซอฟต์แวร์ทางออนไลน์ งานจะจบเมื่อข้อตกลงทั้งหมดถูกยอมรับหรือปฏิเสธ อย่างใดอย่างหนึ่ง

System arbiter:

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

system arbiters ที่ระบุโดยชื่องาน QSYSARB และ QSYSARB2 ถึง QSYSARB5 คือ งานศูนย์กลางและมีระดับความสำคัญสูง สุดภายในระบบปฏิบัติการ แต่ละ system arbiter ตอบสนองต่อเหตุการณ์ต่างๆ ที่ทั้งระบบที่ต้องถูกจัดการโดยทันที และ เหตุการณ์ที่สามารถจัดการโดยงานเดียวได้อย่างมีประสิทธิภาพมากกว่าหลายงาน

system arbiter (QSYSARB) ยังรับผิดชอบสำหรับการเริ่ม งาน Logical Unit Services (QLUS) ในระหว่าง IPL ด้วย system arbiter ยังคง แอ็คทีฟจนกว่าระบบจะจบลง

ข้างล่างนี้เป็นรายการของ system arbiters

Qsysarb (system arbiter)

system arbiter จะจัดเตรียมสถานะแวดล้อมสำหรับการรันฟังก์ชันที่มีความสำคัญสูง และยังช่วยจัดการรีซอร์สระบบ และช่วยติดตามสภาพของ ระบบให้อีกด้วย system arbiter จะตอบสนองเหตุการณ์ต่างๆ ที่ทั้ง ระบบที่จะต้องมีการ จัดการในทันที และเหตุการณ์ที่สามารถจัดการได้ อย่างมีประสิทธิภาพมากขึ้นด้วยงานชิ้นเดียว Qsysarb, Qtaparb (tape arbiter) และ Qcmnarbxx (communications arbiters) รับผิดชอบสำหรับการประมวลผลคำขอการสื่อสาร, การล็อกอุปกรณ์, โลင်း, คอนโทรลเลอร์ และ configuration อุปกรณ์ และการจัดการรีซอร์สที่ทั้งระบบอื่น

Qsysarb2 (system arbiter 2)

งานตัวนี้จะรับผิดชอบการจัดการเทปรีซอร์ส, จัดการ command analyzer space สำหรับการประมวลผลคำสั่ง และการ ประมวลผล อื่นๆ ที่ระบบสำหรับระบบปฏิบัติการ

Qsysarb3 (system arbiter 3)

งานตัวนี้จะรับผิดชอบการสร้างและการดูแลรักษาโครงสร้างงานบน ระบบ เมื่อใดก็ตามที่จำเป็นต้องใช้โครงสร้างงาน ชั่วโมงหรือถาวร ในการเริ่มต้นงาน, คำขอดังกล่าวจะถูกประมวลผลโดย Qsysarb3 Qsysarb3 ยังเริ่ม และจบงาน ระบบหลายอย่างด้วย

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, และขีดจำกัดตารางการล็อก โดย ปกติแล้ว เหตุการณ์ ของเครื่องจะถูกจัดการและข้อความ CPF ที่สัมพันธ์กันนั้นจะถูกส่งไปยัง Qsysopr และ Qhst

งานสื่อสารของระบบ:

หัวข้อนี้มีรายการของงานสื่อสารของ ระบบ

Qlus (logical unit services)

Qlus จัดการเหตุการณ์สำหรับอุปกรณ์โลจิคัลยูนิต หรือที่เรียกกันว่า อุปกรณ์สื่อสาร Qlus ยังรับผิดชอบการจัดสรร อุปกรณ์ให้กับระบบย่อย การสื่อสารที่ถูกต้องอีกด้วย

Qcmnarbxx (communication arbiters)

communications arbiters ที่มี Qsysarb (system arbiter) และ Qtaparb (tape arbiter) ประมวลผลงานสำหรับอุปกรณ์ ทุกชนิด ไม่เฉพาะแต่อุปกรณ์ การสื่อสาร งานเหล่านี้รวมถึงการเชื่อมต่อระบบสื่อสาร, การตัดการเชื่อมต่อ, การล็อก

อุปกรณ์ และการประมวลผลวิธีแก้ไขข้อผิดพลาด 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:

ข้อมูลนี้ประกอบด้วยรายการงานฐานข้อมูล

Qdbfstccol (การรวบรวมสถิติไฟล์ฐานข้อมูล)

งานนี้จะรวบรวมข้อมูลสถิติเกี่ยวกับไฟล์ฐานข้อมูล สถิติเหล่านี้มีความสำคัญอย่างยิ่ง ต่อการแก้ไขการเคียวรีฐานข้อมูลให้มีประสิทธิภาพสูงสุด

Qdbsrvxr (การอ้างอิงข้ามฐานข้อมูล) และ Qdbx###xr สำหรับกลุ่มดิสก์พูล อิสระ

งานตัวนี้จะดูแลรักษาไฟล์การอ้างอิงข้ามระบบระดับไฟล์ใน Qsys ไฟล์เหล่านี้จะประกอบด้วยข้อมูลการอ้างอิงข้ามเกี่ยวกับไฟล์ฐานข้อมูลและข้อมูล SQL ข้ามระบบ ไฟล์เหล่านี้จะเริ่มต้นด้วยค่าเสริมหน้า เป็น Qadb ในไลบรารี Qsys ไฟล์หลักที่จะต้องได้รับการดูแล คือ Qadbxref, ซึ่งเป็นไฟล์อ้างอิงข้าม ไฟล์นี้ประกอบด้วย เร็กคอร์ดของฐานข้อมูลพินัล, 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 Qdbs###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) ที่แอ็คเซสพาทตัวใหม่ถูกเรียกคืนมา จากนั้น SMAPP จะระบุได้ว่าแอ็คเซสพาทเหล่านี้ต้องการให้มีการ ปกป้องหรือไม่
- การเตรียมลิสต์ของแอ็คเซสพาทที่จำเป็นต้องสร้างขึ้นใหม่ เนื่องจากแอ็คเซสพาทไม่ได้ถูกเรียกคืน

ในส่วนครั้งแรกของงานเซิร์ฟเวอร์ฐานข้อมูลที่เหลืออยู่นั้น จะประมวลผลค่าขอที่มีระดับความสำคัญสูง, และค่าขอที่มีระดับความสำคัญต่ำจะ ประมวลผลที่ครั้งหลัง (ตัวอย่าง: Qdbsrv02 ถึง Qdbsrv05 มีระดับความสำคัญสูง Qdbsrv06 ถึง Qdbsrv09 มีระดับความสำคัญต่ำ)

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

งานเหล่านี้จะทำเจอร์นัลและ commitment control maintenance ให้กับ ระบบและถูกจัดว่าเป็นงานที่มีการรันรวดเร็ว หรือใช้เวลานสั้น

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

งานเหล่านี้ดูแลรักษาแอ็คเซสพาร์ในไฟล์ข้อมูลของผู้ใช้โดยปกติงานเหล่านี้จะ inactive, แต่ในบางกรณีงานอาจปฏิบัติงานการสร้างแอ็คเซสพาร์ขึ้นใหม่ เหตุผลบางประการที่งานเหล่านี้อาจจะแอ็คทีฟคือ:

- การคืนสภาพไฟล์ฐานข้อมูลที่ไม่ได้ถูกบันทึกพร้อมกับแอ็คเซสพาร์
- 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. ถ้าผู้ใช้เคียวรีไฟล์แบบกระจายงานหลายงานจะถูกเรียกใช้เพื่อให้ได้ผลลัพธ์ที่เร็วขึ้นโดยการแบ่งงานเป็นหลายงานในลักษณะขนาน .

งานระบบอื่น:

ข้อมูลนี้มีรายการงานระบบชนิดอื่น

Qalert (alert manager)

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

Qdcpobjx (คลายอ็อบเจกต์ระบบ)

งานนี้คลายอ็อบเจกต์ระบบปฏิบัติการที่ติดตั้งใหม่เมื่อจำเป็น งานเหล่านี้ต้องการใช้หน่วยความจำเพื่อจะทำงาน ถ้าหน่วยความจำที่พร้อมใช้งาน บนระบบของคุณลดลงน้อยกว่าขีดจำกัด งานเหล่านี้จะสิ้นสุดจำนวนของงานคลายการบีบอัดของอ็อบเจกต์ระบบคือจำนวนของโพเรสเซออร์บวกด้วยหนึ่ง

Qfilesystem1 (ระบบไฟล์)

งานตัวนี้จะสนับสนุนการประมวลผลส่วนหลังของระบบไฟล์รวม งานนี้ทำให้มั่นใจว่าการเปลี่ยนแปลงที่ไฟล์จะถูกบันทึกลงในหน่วยความจำ และปฏิบัติ activities การล้างระบบไฟล์ทั่วไปด้วย

Qjobsd (ตารางเวลางาน)

งานตัวนี้จะควบคุมฟังก์ชันการจัดตารางเวลาของระบบ Qjobsd จะมอนิเตอร์ตัวจับเวลาของ job schedule entry และงานที่จัดตาราง เวลา

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

งานนี้ล้างไลบรารีบนดิสก์พูลอีสระ

Qli###rp สำหรับกลุ่มดิสก์พูลอีสระ ### (การล้างอ็อบเจกต์)

งานนี้ล้างอ็อบเจกต์ที่แทนที่บนไลบรารีดิสก์พูลอีสระ

Qlur (LU 6.2 resynchronization)

Qlur จะจัดการกระบวนการ two-phase commit resynchronization

Qpfradj (การปรับประสิทธิภาพ)

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

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

งานนี้ทำหน้าที่ของฟังก์ชันการสุขภาพของระบบที่ประกอบด้วย:

- การล้างสพูลไฟล์หลังจาก IPL หรือกลุ่มดิสก์พูลอิสระถูก varied on
- ย้าย stranded spooled files ของเอาต์พุตคิวผู้ใช้ที่เสียหายในพูลหน่วยความจำรองระบบย่อย หรือในพูลหน่วยความจำรองผู้ใช้ระดับต้นไปในเอาต์พุตคิว QSPRCLOUTQ ในไลบรารี QRCL
- ลบเมมเบอร์ฐานข้อมูลสพูล ซึ่งมีข้อมูล และแอตทริบิวต์ของสพูลไฟล์ที่ลบบอก
- ลบเมมเบอร์ฐานข้อมูลสพูลที่ไม่ได้ถูกใช้ซ้ำภายในเวลาที่ระบุในค่ากำหนดของระบบ Automatically clean up unused printer output storage (QRCLSPLSTG)

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

งานนี้ปฏิบัติการดำเนินการสพูลไฟล์สำหรับกลุ่มดิสก์พูล อิสระเฉพาะ

Qtaparb (อุปกรณ์เทป)

งานนี้ประมวลผลงานที่เกี่ยวข้องกับอุปกรณ์เทป รวมถึง การลือกอุปกรณ์ และการประมวลผลการแก้ไขข้อผิดพลาด

Qnwharbxx

งานระบบเหล่านี้จัดการเหตุการณ์ที่เกี่ยวข้องกับอุปกรณ์ Network Server Host Adapter (NWSH) จะมีการเริ่มงานเหล่านี้อย่างน้อยหนึ่งงานเสมอในระหว่าง IPL ปัจจุบัน

Qwcpjobs

งานนี้จัดการการล้างแบ็คกราวด์ของโครงสร้างงานถาวร

Qwctjobs

งานนี้จัดการการล้างแบ็คกราวด์ของโครงสร้างงานชั่วคราว

อ็พชันการจัดตารางเวลางาน

ฟังก์ชันการจัดตารางเวลางานช่วยในการจัดตารางเวลางานแบ็คซ์ System i ตามเวลาที่ต้องการได้ คุณสามารถจัดตารางเวลาที่งานจะถูกรีลีจากคิวงาน ณ เวลาที่ต้องการ หรือคุณสามารถใช้ job schedule entry เพื่อส่งงานของคุณไปที่คิวงาน โดยอัตโนมัติ เวลาที่คุณระบุ การจัดตารางเวลางานช่วยให้คุณควบคุม วันที่และเวลาที่งานแบ็คซ์จะถูกส่งหรือกลายเป็นมีลิสต์ที่จะเริ่ม จากคิวงาน ความยืดหยุ่นนี้เป็นสิ่งที่มีประโยชน์เมื่อคุณปรับสมดุลเวิร์กโหลด บนระบบของคุณ

ตัวอย่างเช่น คุณสามารถใช้การจัดตารางเวลางานเพื่อมอบหมายงานที่ซ้ำ อย่าง การส่งจดหมายแจ้งเชิญประชุมซ้ำๆ , ค่าจ้าง หรือรายงานประจำสัปดาห์และ ประจำเดือน จากตารางเวลางานของคุณไปที่ตารางเวลางานของระบบได้ การจัดตารางเวลางานแบ็คซ์สามารถทำได้สี่วิธี

Management Central Scheduler

System i Navigator นำเสนอ integrated scheduler, Management Central scheduler เพื่อจัดตารางเวลาที่คุณต้องการให้งานของคุณประมวลผล คุณมีอ็พชันการเลือกที่จะปฏิบัติ การกิจในทันทีหรือในภายหลัง คุณสามารถใช้ Management Central scheduler เพื่อจัดตารางเวลาการกิจส่วนใหญ่ใน Management Central

หน้าต่าง Management Central Scheduler พร้อมใช้งานได้ตลอดเวลาที่คุณเห็นปุ่ม **Schedule** บนหน้าต่าง System i Navigator

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

งานที่เกี่ยวข้อง

“การจัดตารางเวลางานโดยใช้ Management Central Scheduler” ในหน้า 149

ถ้าคุณไม่ได้ติดตั้ง plug-in Advanced Job Scheduler คุณสามารถใช้ Management Central Scheduler เพื่อจัดตารางเวลางานได้

Advanced Job Scheduler

ตัว IBM® Advanced Job Scheduler for i5/OS (5761-JS1) โลเซ็นส์โปรแกรมเป็นตัวจัดลำดับงานที่ยืดหยุ่นที่ช่วยให้งานแบบ unattended ถูกประมวลผลได้ตลอด 24 ชั่วโมงต่อวัน, 7 วันต่อสัปดาห์ เครื่องมือการจัดตารางเวลานี้แนะนำคุณลักษณะปฏิทินเพิ่มเติม และนำเสนอการควบคุมเหตุการณ์ที่จัดตารางเวลาในระดับที่สูงขึ้นกว่าตัวจัดตารางเวลา Management Central นอกจากนี้ คุณยังสามารถดูประวัติความสมบูรณ์ของงานและบริหารการแจ้งเตือนสถานะของงานได้อีกด้วย

หากคุณต้องการจัดตารางเวลางานบนหลายระบบในเน็ตเวิร์กของคุณ ต้องติดตั้งผลิตภัณฑ์บนแต่ละระบบของคุณ หากคุณต้องการใช้ Advanced Job Scheduler ใน System i Navigator (และใน Management Central) จากนั้น คุณต้องติดตั้ง client plug-in จากระบบที่ติดตั้ง Advanced Job Scheduler ไว้

ทั้งนี้ ไม่จำเป็นต้องติดตั้ง Advanced Job Scheduler โลเซ็นส์โปรแกรมบนระบบจุดปลายทุกระบบในเน็ตเวิร์ก Management Central ของคุณ เมื่อคุณติดตั้ง Advanced Job Scheduler บนระบบศูนย์กลาง งาน หรือภารกิจที่คุณกำหนดบนระบบจุดปลาย จะรวบรวมข้อมูลที่ต้องการจากระบบศูนย์กลาง คุณต้องตั้งค่าข้อมูล job definition ทั้งหมดบนระบบศูนย์กลาง

ถ้าระบบในเน็ตเวิร์กของคุณมี Advanced Job Scheduler ติดตั้งไว้แบบโลคัล คุณสามารถจัดตารางเวลาภารกิจภายนอกเน็ตเวิร์ก Management Central ได้ ภายใต้ My Connections ใน System i Navigator, คุณมีสิทธิเข้าใช้งาน Advanced Job Scheduler บนระบบโลคัลเมื่อคุณขยาย Work Management.

หมายเหตุ: สำหรับข้อมูลในการสั่งซื้อ ให้อ่านที่ Job Scheduler สำหรับ i5/OS  Web site.

Job schedule entries

ถ้าระบบของคุณไม่มี Management Central Scheduler หรือ Advanced Job Scheduler คุณยังคงสามารถจัดตารางเวลางานได้โดยใช้ job schedule entry ซึ่งเข้าใช้งานได้จากอินเทอร์เน็ตเฟสแบบอักขระ ด้วยการใช้อินเทอร์เน็ต คุณจึงสามารถจัดตารางเวลางานในรันเช้าหรือรันเพียงครั้งเดียวได้

เนื่องจาก job schedule entries คือ entries ในอ็อบเจกต์ถาวร entry ชนิดนี้จึงไม่ได้อยู่บนคิวงานเหมือนกับงานที่จัดตารางเวลาดังนั้นจึงไม่สูญหาย เมื่อคิวงานถูกลบ คุณยังสามารถบันทึกและคืนสภาพอ็อบเจกต์ ตารางเวลางานได้ด้วย นี่เป็นวิธีการสำรองข้อมูลตารางเวลางานของคุณ

เมื่อคุณต้องการให้งานประมวลผลในช่วงเวลาสม่ำเสมอ คุณสามารถสร้าง job schedule entry สำหรับงานได้ job schedule entry มีข้อมูลทั้งหมดที่จำเป็น สำหรับการส่งงานและข้อมูลการจัดตารางเวลางาน แต่ละ entry ในอ็อบเจกต์ถูกระบุอย่างไม่ซ้ำกันโดยชื่องานที่คุณให้และหมายเลข entry 6 ตำแหน่งที่ระบบกำหนดให้ ไม่มีสอง entry ใดมีชุดของ ชื่องานและหมายเลข entry เหมือนกัน

นอกจากนี้ job schedule entry ยังมีข้อมูลที่ระบบใช้ในการจัดการ entry ในบางสถานการณ์ด้วย ข้อมูลที่กำหนดงานคล้ายกับ พารามิเตอร์ที่ระบุบนคำสั่ง Submit Job (SBMJOB) ได้แก่ ชื่องาน, รายละเอียดของงาน, คิวงาน, โพรไฟล์ผู้ใช้ และ message queue local data area (LDA) ของงานที่ส่งจาก job schedule entry จะว่างเปล่าเมื่องานเริ่มต้น

job schedule entries ทั้งหมดมีอยู่ในอ็อบเจกต์ตารางเวลางาน อ็อบเจกต์ตารางเวลางาน QDFTJOBSCD อยู่ในไลบรารี QUSRSYS และมีชนิด อ็อบเจกต์เป็น *JOBSCD คุณไม่สามารถสร้าง, ลบ, เปลี่ยนชื่อ หรือทำซ้ำอ็อบเจกต์ตาราง เวลางาน คุณไม่สามารถย้ายอ็อบเจกต์ไปที่ไลบรารีอื่น อ็อบเจกต์ตารางเวลางานถูกจัดส่งมา พร้อมกับสิทธิพัลลิก *CHANGE นี่เป็น สิทธิต่ำสุดที่จำเป็นสำหรับการ ใส่เพิ่ม, เปลี่ยน, พัก, รีลีส, หรือลบ job schedule entry

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

หลักการที่เกี่ยวข้อง

“การทำงานกับ job schedule entries” ในหน้า 175

นอกเหนือจากหน้าต่าง System i Navigator Job Properties – Job Queue คุณยังสามารถเปลี่ยนแปลง job schedule entry ได้โดยตรง โดยใช้อินเตอร์เฟซแบบอักษร ข้างล่างนี้เป็นรายการของภารกิจอินเตอร์เฟซ แบบอักษรทั่วไป ที่คุณสามารถ ใช้เมื่อทำงานกับ job schedule entries

ตัวอย่าง: job schedule entry:

หัวข้อนี้แนะนำตัวอย่างสำหรับการใช้คำสั่ง Add Job Schedule Entry (ADDJOBSCDE)

จัดตารางเวลางานรายเดือน: ตัวอย่างนี้แสดงวิธีการส่งงาน เพื่อรันโปรแกรม INVENTORY เวลา 11:30 p.m. ในวันสุดท้าย ของทุกเดือน 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. งานจะรัน ภายใต้อุปกรณ์ผู้ใช้ SOMEPMGR งานนี้จะไม่ถูกส่ง ถ้าระบบปิดหรืออยู่ในสภาพถูกจำกัดในเวลานั้น

```
ADDJOBSCDE JOB(*JOB)
CMD(CALL DAILYCLEAN)
SCDDAY(*ALL)
SCDTIME('18:00:00')
SCDDATE(*NONE)
USER(SOMEPMGR)
FRQ(*WEEKLY)
RCYACN(*NOSBM)
```

การจัดตารางเวลางานรายสัปดาห์: ตัวอย่าง นี้แสดงวิธีการส่งงานเพื่อรันโปรแกรม PGM1 ทุกสัปดาห์ โดยเริ่มต้นวันที่ 12/17/05 ที่เวลาปัจจุบัน เนื่องจาก 12/17/05 เป็นวันเสาร์ ดังนั้นงานจึงจะถูกส่งทุกวันเสาร์ และรันภายใต้อุปกรณ์ผู้ใช้


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

การจัดตารางเวลางานทุกวันจันทร์และวันพุธที่สาม: ตัวอย่างนี้แสดงวิธีการส่งงานเพื่อรันโปรแกรม PGM2 ในวันจันทร์ที่สามและวันพุธที่สามของเดือน ในเวลา 11:30 p.m. งานนี้จะถูกส่งในวันจันทร์หรือวันพุธที่สามถัดไปในเวลา 11:30 p.m. ขึ้นอยู่กับว่าวันเหล่านั้นของเดือนนี้ได้ผ่านไปแล้วยังหรือไม่ ถ้าเมื่อวานเป็นวันจันทร์ที่สาม วันนี้คือวันอังคารที่สาม และพรุ่งนี้คือวันพุธที่สาม งานจะถูกส่งวันพรุ่งนี้แล้วไม่ส่งอีกจนกว่าจะถึงเดือนถัดไป

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

คำสั่งการส่งงาน

คำสั่งอินเตอร์เฟซแบบอักษรนี้ควบคุมเวลาที่งาน จะถูกรีลีสในคิวงาน คำสั่งนี้เป็นวิธีที่ง่ายในการจัดตารางเวลางานที่ต้องรันเพียงครั้งเดียวเท่านั้น คำสั่งนี้อุญาตให้คุณใช้แอ็ททริบิวต์งานที่กำหนดสำหรับ งานปัจจุบันของคุณได้หลายแอ็ททริบิวต์

เมื่อคุณจัดตารางเวลางานเพื่อที่จะรันเพียงครั้งเดียว (คำสั่งแบบอักษร SBMJOB) งานจะถูกรีลีสจากคิวงานที่เวลาที่จัดตารางเวลาไว้ ข้อมูลต่อไปนี้ เป็นข้อสรุปสำหรับงานระบบที่จะเกิดขึ้นเมื่อคุณใช้ SBMJOB เพื่อจัดตารางเวลางานแบ็ตช์

1. คุณจัดตารางเวลางานโดยใช้ System i Navigator interface (**Basic Operations** → **Jobs** → **อย่างใดอย่างหนึ่งให้คลิกขวาที่งาน** → **Properties** → **Job Queue tab**) หรือการอินเตอร์เฟซแบบอักษร (คำสั่ง SBMJOB ที่มีการระบุพารามิเตอร์ SCDATE และ SCDTIME ที่ระบุ)

2. งานยังคงอยู่บนคิวงานในสถานะถูกจัดตารางเวลา (สถานะ SCD) จนกว่าจะถึง วันที่และเวลาที่ระบุโดยพารามิเตอร์
3. ณ เวลาที่จัดตารางเวลาไว้งานจะถูกรีลีสจากคิวงาน สถานะของงาน จะเปลี่ยนจากถูกจัดตารางเวลา (SCD) เป็นรีลีส (RLS) ยกเว้นว่างานจะถูกพัก (SCDHL) ซึ่งสถานะจะเปลี่ยนจากถูกจัดตารางเวลาเป็นถูกพัก (HLD)
4. งานถูกประมวลผลเช่นเดียวกับงานอื่นๆ บนคิวงาน
5. งานจะเริ่มต้นถ้ามีสภาพปกติ (เช่น คิวงานถูกจัดสรรไปที่ระบบย่อยที่แอคทีฟ และงานสูงสุดยังไม่แอคทีฟ)

หมายเหตุ: วิธีการนี้จะวางแผนบนคิวงานในทันที ดังนั้นถ้าคิวงานถูกลบก่อนวันที่ และเวลาที่จัดตารางเวลา คุณจะเสียงานของคุณ

งานที่เกี่ยวข้อง

“การส่งงานครั้งเดียว” ในหน้า 127

เมื่อคุณต้องการรันงานเพียงครั้งเดียว ไม่ว่าจะรันในทันที หรือรันในวันที่ และเวลาที่จัดตารางไว้ ให้ใช้คำสั่ง Submit Job (SBMJOB) วิธีนี้จะ วางงานบนคิวงานในทันที

“การส่งงานแบ็คซ์” ในหน้า 131

เนื่องจากโดยปกติ งานแบ็คซ์เป็นงานที่มีระดับความสำคัญต่ำที่ต้องการ สภาวะแวดล้อมระบบพิเศษในการรัน (เช่น การรันในเวลากลางคืน) งานแบ็คซ์จึงถูกวางในคิวงานแบ็คซ์ในคิวงาน งานแบ็คซ์จะได้รับตารางเวลารันไทม์ และระดับความสำคัญ ถ้าต้องการส่งงานไปที่คิวงานแบ็คซ์ให้คุณ ใช้อินเตอร์เฟซแบบอักษรและหนึ่งในสองคำสั่งต่อไปนี้

ข้อควรพิจารณาเกี่ยวกับตารางเวลางาน

เมื่อทำการเลือกผลิตภัณฑ์ตารางเวลางาน คุณต้องพิจารณาความหลากหลายของคุณลักษณะพิเศษต่างๆ ต่อไปนี้เป็นรายการของคุณลักษณะพิเศษที่ต้องพิจารณาเมื่อมีการตัดสินใจว่าจะใช้ตารางเวลางานใด:

- **การจัดตารางเวลางานแบบอัตโนมัติ**
 - ความยืดหยุ่นในการจัดตารางเวลางาน
 - การประมวลผลงานที่ไม่ตั้งใจ (หรือ ตั้งใจ) ตลอด 24 ชั่วโมงต่อวัน, 7 วันต่อสัปดาห์ พร้อมด้วยการประยุกต์ทั้งหมดให้เข้ากับตารางเวลางานที่คุณตั้งไว้
 - ส่วนขยายของระบบปฏิบัติการ i5/OS
 - การควบคุมโดยสมบูรณ์ของงานที่ส่งเข้ามาว่า อย่างไร, เมื่อไร และที่ไหน
 - งานส่วนขยายขึ้นอยู่กับ อ็อบเจกต์ (การเกิดขึ้นของไฟล์ หรือเรีกคอร์ดภายในฟิลิสต์ไฟล์), activity หรือ inactivity ของงานอื่น หรือ สถานะของไลน์, คอนโทรลเลอร์ หรือ ระบบย่อย
 - เสริมสร้างฟังก์ชันการจัดการปฏิทิน รวมถึงปฏิทินปีบัญชีและปฏิทินวันหยุด
 - การทำงานหลายอย่างต่อวัน
- **พารามิเตอร์ระบบ และพารามิเตอร์ที่ผู้ใช้กำหนด**
 - วันที่ปัจจุบัน, วันที่ส่งงาน, วันที่ก่อนหน้านี้ และเวลาปัจจุบันสามารถส่งผ่านเข้าไปในแอ็พพลิเคชันโปรแกรมได้
 - ค่าพารามิเตอร์ที่ผู้ใช้กำหนดสามารถสร้าง, เปลี่ยน และส่งผ่านเข้าไปในแอ็พพลิเคชันโปรแกรมได้
- **การพยากรณ์เวิร์กโหลด/ประวัติ**
 - พยากรณ์งานที่จัดตารางเวลาทั้งหมดให้รัน สัปดาห์หน้า, เดือนหน้า หรือวันถัดไป
 - Optimize ข้อกำหนดในการผลิต
 - การแทรกข้อมูลในอดีต และการบันทึก Advanced Job Scheduler activity ทั้งหมด
- **การจัดการเน็ตเวิร์ก**

- สามารถตั้งค่างานบน System i ผลิตภัณฑ์ใดๆในเน็ตเวิร์กให้รันบน System i ผลิตภัณฑ์อื่นใดบนเน็ตเวิร์กได้
- ให้ข้อมูลงานในอดีตที่เสร็จสิ้นของงานบนระบบที่มีการส่งงาน
- เราสามารถส่งงานกลุ่มและงานส่วนเสริมผ่านทางเน็ตเวิร์กได้
- **การแจกจ่ายและการจัดการรายงาน**
 - การเรอต์, การมอนิเตอร์ และการควบคุมรายงานเอาต์พุตทั้งหมดที่สร้างโดย Advanced Job Scheduler หรือ i5 / OS ระบบปฏิบัติการ
 - การแจกจ่ายสพูลไฟล์ไปยังเอาต์พุตคิวหลายอัน หรือไปยังระบบรีโมทพร้อมด้วยตัวเลือกแบนเนอร์เพจ
 - สามารถทำสำเนาสพูลเอาต์พุต หรือส่งให้แก่ผู้ใช้ใดๆบน i5 / OS เน็ตเวิร์ก
- **การรักษาความปลอดภัย**
 - ความปลอดภัย i5 / OS ที่มีอยู่สามารถใช้ภายใน Advanced Job Scheduler ได้
 - ให้ระบุบุคคลในองค์กรของคุณที่มีสิทธิในการตั้งค่าหรือเปลี่ยนแปลงข้อมูลเกี่ยวกับงานที่จัดตารางเวลา
 - เราสามารถระบุสิทธิในการใช้งานสำหรับฟังก์ชันแต่ละฟังก์ชันของ Advanced Job Scheduler หรือสำหรับงานที่ระบุเฉพาะ อย่างไม่อย่างหนึ่งได้
- **Graphical user interface**
 - ความสามารถในการเลือกและคลิก เมื่อจัดตารางเวลางาน
 - การจัดการงาน
 - การคงไว้ของส่วนเสริม
 - การแทร็ก activity ของตัวจัดตารางเวลาและข้อมูลไฟล์บันทึก
- **คุณลักษณะพิเศษของคีย์อื่นๆ**
 - หลายคำสั่งต่องาน
 - Definition สำหรับงาน LDA (Local Data Area)
 - มอนิเตอร์คอนโซลเพื่อรันทานในสภาพที่ถูกจำกัด
 - การตรวจสอบรันไทม์สูงสุดสำหรับแต่ละงาน
 - อินเตอร์เฟสโดยตรงกับระบบการเพจแบบข้อความของบริษัทที่ให้บริการอื่นๆ
 - บทบัญญัติสำหรับเอกสารคู่มือออนไลน์แบบเต็มของแต่ละงาน
 - ปุ่มสัมผัสเคอร์เซอร์ที่ขยายข้อความคำอธิบายบนจอแสดงผลทั้งหมด

การจัดตารางเวลางานและสภาพพร้อมใช้งานระบบ

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

job schedule entries และงานที่จัดตารางเวลาจะถูกประมวลผลในลำดับของ การจัดการตามปกติถ้างานเหล่านั้นสามารถดำเนินการได้. 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 ที่งานแบบแบดช์ที่ส่งไปพักรอ ก่อนที่จะไปทำงาน อยู่ในระบบย่อย งานจะถูกพักอยู่ที่นั่นจนกว่าจะมีสภาพตรงตามเงื่อนไขการทำงานต่างๆ

งานที่อยู่ในคิวงานจะได้รับการประมวลผล ก็ต่อเมื่อมีระบบย่อยที่แฉีกที่ปรับเองงานที่อยู่ในคิวงานไป เมื่อระบบย่อยเริ่มทำงาน ระบบย่อยดังกล่าวจะพยายามจัดสรรคิวงานที่ถูกตั้งไว้ให้รับงานเข้ามา และจะต้องจัดสรรคิวงานเพื่อทำการประมวลผลงานจากคิวงานนั้นให้ประสบความสำเร็จ ดังนั้น ในขณะที่ระบบย่อยหนึ่งสามารถประมวลผลงานจากหลายคิวงานได้ แต่จะมีระบบย่อยเพียงระบบเดียวเท่านั้นที่สามารถประมวลผลงานจากคิวงานหนึ่งในแต่ละครั้ง

ระบบย่อยจะเลือกงานจากคิวงานตามระดับความสำคัญ ภายใต้ข้อจำกัดที่ได้ตั้งค่าไว้สำหรับแต่ละระดับความสำคัญ งานแต่ละชั้นมีค่าระดับความสำคัญคิวงานที่สามารถ จัดการได้เมื่องานนั้นอยู่บนคิวงานผ่านทางคุณสมบัติของงาน ชุดคิวงานพื้นฐานจะมาพร้อมกับระบบของคุณ นอกจากนั้น, คุณยังสามารถสร้างคิวงานเพิ่มเติมตามความต้องการได้อีกด้วย

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

หลักการที่เกี่ยวข้อง

“การจัดการคิวงาน” ในหน้า 207

เมื่อคุณจัดการงานบนระบบ คุณอาจพบว่ามีความจำเป็น ต้องดำเนินการงานที่กำลังรออยู่ในคิวงาน บางที อาจมีบางคนต้องการ รันงานในทันที แต่งานรออยู่ในคิวโดยมีระดับความสำคัญต่ำ หรือบางทีคุณต้องการปฏิบัติการบำรุงรักษาบางอย่างบนระบบย่อย และต้องการย้าย งานทั้งหมดไปที่คิวซึ่งไม่ได้เชื่อมโยงกับระบบย่อยนั้น

งานที่เกี่ยวข้อง

“การลบคิวงาน” ในหน้า 209

เมื่อคุณล้างข้อมูลคิวงาน งานทุกงานบนคิวจะถูกลบออก รวมถึงงานใดๆ ที่อยู่ในสถานะถูกพักด้วย คุณสามารถใช้ System i Navigator หรืออินเตอร์เฟซแบบอักษรเพื่อลบคิวงาน งานที่กำลังรันอยู่จะไม่ได้รับ ผลกระทบ เนื่องจากถือว่างานเหล่านั้นเป็นงานแฉีกที่ไฟและไม่ได้อยู่บนคิว อีกต่อไป

“การสร้างคิวงาน” ในหน้า 209

ถ้าต้องการสร้างคิวงาน ให้ใช้อินเตอร์เฟซแบบอักษร

“การลบคิวงาน” ในหน้า 210

ถ้าต้องการลบคิวงาน ให้ใช้อินเตอร์เฟซแบบอักษร

“การพักคิวงาน” ในหน้า 211

เมื่อคุณพักคิวงาน นั้นหมายถึงคุณป้องกันการประมวลผลงานทั้งหมดที่กำลังรออยู่บนคิวงานในปัจจุบัน การพักคิวงานไม่มีผลต่องานที่กำลังรันอยู่ สามารถวางงานเพิ่มเติม บนคิวงานในขณะที่คิวงานถูกพักได้ แต่งานจะไม่ถูกประมวลผล

“การรีลีสคิวงาน” ในหน้า 211

เมื่อคุณรีลีสคิวงาน งานทั้งหมดที่ถูกพักไว้ก่อนเป็นผลมาจาก การพักคิวงานจะถูกรีลีสด้วย ถ้า งานแต่ละงานถูกพักก่อนที่คิวงานจะถูกพัก งานจะไม่ถูก รีลีส

ข้อมูลที่เกี่ยวข้อง

API การจัดการระบบงาน

รายการที่จัดลำดับ

รายการที่จัดลำดับแสดงถึงลำดับที่งานจะปรากฏขึ้นบนคิวงาน สภาพพร้อมใช้งาน, ระดับความสำคัญ และค่าวันที่และเวลาช่วยกำหนด ลำดับของงานบนคิวงาน

ไม่ใช่หมายเลขงานในการกำหนดลำดับของงานในคิวงาน เช่นเดียวกับการไม่ใช่ผลกระทบเมื่องานถูกรัน

สภาพพร้อมใช้งาน

หมายถึงสถานะของงานบนคิวงาน ค่าที่เป็นไปได้ตามลำดับคือ กำลังรอ, ที่จัดตารางเวลา และถูกพัก

ระดับความสำคัญ

หมายถึงระดับความสำคัญของงานบนคิวงาน ค่าระดับความสำคัญที่เป็นไปได้คือ 0-9 โดยที่ 0 เป็นระดับความสำคัญสูงสุดในกรณีที่งานเป็น งานที่จัดตารางเวลา ระดับความสำคัญจะไม่มีส่วนในการจัดลำดับของ งานบนคิวงาน ตัวอย่างเช่น ถ้างานสองงานถูกจัดตารางเวลาให้รันเวลา 12:00:00 งานจะถูกจัดลำดับตามตำแหน่งของงานในตารางเวลางาน

วันที่และเวลา

หมายถึงวันที่และเวลาของงาน:

- หากงานถูกจัดตารางเวลาไว้วันที่และเวลาหมายถึงเวลาที่งานถูกจัดตารางเวลาให้รัน
- หากงานไม่ได้ถูกจัดตารางเวลาไว้วันที่และเวลาหมายถึงเวลาที่งานเข้าสู่ระบบ

หมายเหตุ: มีโอกาสที่วันที่และเวลาการจบงานจะถูกใช้เป็นที่และเวลาที่เซตด้วยตนเอง เพื่อจัดตำแหน่งงานที่ย้ายไปที่คิวงานอย่างถูกต้อง

วิธีการทำงานของคิวงาน

คิวงานถูกจัดสรรโดยระบบย่อยผ่านทาง job queue entry. สามารถวางงานบนคิวงานได้ แม้ว่ายังไม่ได้เริ่มระบบย่อย. เมื่อระบบย่อยเริ่มขึ้น, ระบบย่อยจะประมวลผลงานบนคิว.

subsystem description ระบุจำนวนสูงสุดของงาน (แบบเบ็ดเสร็จหรือ แบบโต้ตอบ) ที่สามารถแอนด์ที่ไฟได้พร้อมกัน. จำนวนของงานที่สามารถ แอนด์ที่ไฟจากคิวงานใดๆ ถูกระบุอยู่ใน job queue entry.

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

subsystem description มากกว่าหนึ่ง description สามารถอ้างอิงคิวงานเดียวกันได้, แต่ในแต่ละครั้งจะมีระบบย่อยที่แอนด์ที่ไฟเพียงระบบเดียวเท่านั้นที่สามารถใช้ คิวงานเป็นซอร์สของงานเบ็ดเสร็จได้. ดังนั้น, ถ้าระบบย่อยจบลงและงานยังคงอยู่บนคิวงาน, ระบบย่อยอื่นที่อ้างอิงไปที่คิวงานนั้นสามารถถูกเริ่มให้ประมวลผลงานได้. ถ้าระบบย่อยอื่นได้ถูกเริ่มต้นแล้วและกำลังรอคิวงานเดียวกัน, ระบบย่อยจะจัดสรรคิวงานให้โดยอัตโนมัติเมื่อคิวงาน พร้อมใช้งาน.

หลักการที่เกี่ยวข้อง

“ระบบย่อยจัดการกับหลายคิวงานได้อย่างไร” ในหน้า 207

เพื่อสาธิตวิธีการที่ระบบย่อยใช้ในการจัดการกับหลายคิวงาน ให้พิจารณา สถานการณ์จำลองนี้
งานที่เกี่ยวข้อง

“การกำหนดว่าระบบย่อยใดมีคิวงานที่ถูกจัดสรร” ในหน้า 210

คุณสามารถกำหนดว่าระบบย่อยใดมีคิวงานที่ถูกจัดสรรได้ โดยใช้ System i Navigator อินเทอร์เน็ตหรืออินเทอร์เน็ตแบบ
อักษร สิ่งนี้มีประโยชน์เมื่อคุณพบว่าจำเป็นต้อง ลบคิวงานออก เนื่องจากคุณไม่สามารถลบคิวงานซึ่งมี ระบบย่อยแอ็คทีฟ

“การสร้างคิวงาน” ในหน้า 209

ถ้าต้องการสร้างคิวงาน ให้ใช้อินเทอร์เน็ตแบบอักษร

“การกำหนดคิวงานที่ระบบย่อย” ในหน้า 207

ถ้าต้องการกำหนด job queue entry ที่ subsystem description ให้ใช้อินเทอร์เน็ตแบบอักษร

วิธีการนำงานมาจากคิวงาน

มีปัจจัยหลายอย่างที่กำหนดวิธีการเลือกงานจากคิวงาน และเริ่มต้น.

งานแอ็คทีฟสูงสุดสำหรับระบบย่อย

ตัวเลขนี้แสดงจำนวนสูงสุดของงานที่สามารถรันได้ในระบบย่อยหนึ่ง. เมื่อใช้งานถึงจำนวนสูงสุดนี้, จะไม่สามารถเริ่ม
งานเพิ่มเติมในระบบย่อย.

งานแอ็คทีฟสูงสุดสำหรับคิวงาน

ตัวเลขนี้แสดงจำนวนสูงสุดของงานจากคิวงานที่สามารถรันพร้อมกัน ได้ในระบบย่อย. หลังจากใช้งานถึงจำนวนสูง
สุดนี้แล้ว, จะ ไม่สามารถเริ่มงานเพิ่มเติมจากคิวงานนั้น.

ระดับความสำคัญบนคิวงาน

งานที่กำลังรอที่จะรันจะถูกเลือกขึ้นมารันตามระดับความสำคัญใน คิวงาน. ระบบจะพยายามที่จะรันงานที่มีระดับ
ความสำคัญสูงกว่าก่อน (ระดับความสำคัญในคิวงานมีตั้งแต่ 0-9 โดยที่ 0 คือระดับความสำคัญสูงสุด), แต่ถ้าจำนวน
ของงานที่รันจากรดับความสำคัญมีมากถึงค่า Maximum Active Jobs, ระดับความสำคัญถัดไปจะถูกประมวลผลต่อ
ไป. (ถ้างานที่มีระดับความสำคัญเดียวกัน เข้าสู่คิวงาน, งานแรกที่ส่งจะรันก่อน, จากนั้นเป็นงานที่สอง, และต่อไป.)

ลำดับ คุณสามารถระบุลำดับใน job queue entry ของ subsystem description ได้. หมายเลขลำดับกำหนดลำดับซึ่งระบบย่อย
จะประมวลผลคิวงาน. ระบบย่อยจะดึงเอางานที่มีหมายเลขลำดับต่ำสุดออกจากคิวงานก่อน. หากไม่มีงานอื่นบนคิว
งาน, หรือมีการไปถึงค่าสูงสุดค่าใดค่าหนึ่งที่เกี่ยวข้องกับคิวงาน, ระบบย่อยจะประมวลผลคิวงานที่มีหมายเลข
ลำดับสูงสุดถัดไป.

งานที่เกี่ยวข้อง

“การวางแผนบนคิวงาน” ในหน้า 213

งานถูกวางแผนบนคิวงานโดยการย้ายงานที่มีอยู่จากคิวงานหนึ่งไปยังคิวงานอื่น หรือโดยการส่งงานใหม่ อย่างไม่อย่างหนึ่ง ใช้
System i Navigator เพื่อย้ายงานระหว่างคิวต่างๆ ใช้อินเทอร์เน็ตแบบอักษรเพื่อส่งงานใหม่

“การย้ายงานไปที่คิวงานอื่น” ในหน้า 212

มีหลายเหตุผลที่คุณอาจต้องการย้ายงานไปที่คิวอื่น ตัวอย่างเช่น ในบางครั้งงานอาจถูก backlogged ในคิวเนื่องจากเป็น
งานที่ใช้เวลารันนาน บางครั้งรันใหม่ที่จัดตารางเวลาของงานขัดแย้งกับ งานใหม่ที่มีระดับความสำคัญสูงกว่า วิธีหนึ่งใ
การจัดการสถานการณ์นี้คือ การย้ายงานที่รอไปยังคิวอื่นที่ไม่ยุ่ง

“การเปลี่ยนจำนวนของงานที่รันได้พร้อมกันในคิวงาน” ในหน้า 208

ระบบย่อย QBASE ถูกจัดส่งมาพร้อมกับ job queue entry สำหรับคิวงาน QBATCH entry นี้อนุญาตให้มีงานแบ็ตช์ที่จะรันได้เพียงหนึ่งงานเท่านั้นในแต่ละครั้ง ถ้าคุณต้องการให้สามารถรันงานแบ็ตช์ได้พร้อมกันหลายงานจากคิวงานนั้น คุณต้องเปลี่ยน job queue entry

Job queue entry

job queue entry ระบุคิวงานซึ่งงานจะถูกเลือกสำหรับการรันในระบบย่อย พารามิเตอร์ใน job queue entry ที่ควบคุมวิธีการจัดการคิวงานมีอยู่ห้าพารามิเตอร์

Subsystem Description (SBSD)

นี่คือชื่อและโลบรารีของ subsystem description ซึ่ง job queue entry จะถูกใส่เพิ่มไป

Job queue (JOBQ)

ระบุชื่อและโลบรารีของคิวงานที่เป็นต้นทางของงานแบ็ตช์ ซึ่งถูกเริ่มโดยระบบย่อย

Maximum active jobs (MAXACT)

ระบุจำนวนสูงสุดของงานที่สามารถแอ็คทีฟได้พร้อมกัน จากคิวงานนี้

Sequence number (SEQNBR)

ระบุหมายเลขลำดับสำหรับคิวงานนี้ ซึ่งถูกใช้โดยระบบย่อย เพื่อกำหนดลำดับการประมวลผลคิวงาน

Max active priority 1 (through 9) (MAXPTYx)

ระบุจำนวนของงานที่สามารถเริ่มสำหรับระดับความสำคัญงาน ที่ระบุงานที่เกี่ยวข้อง

“การใส่เพิ่ม job queue entries” ในหน้า 183

job queue entry ระบุคิวงานซึ่งงานจะถูกเลือกสำหรับการรันในระบบย่อย งานที่เริ่มต้นจากคิวงานคืองานแบ็ตช์ คุณสามารถใส่เพิ่ม job queue entry โดยใช้อินเตอร์เฟซแบบอักษร

“การเปลี่ยน job queue entry” ในหน้า 188

คุณสามารถเปลี่ยน job queue entry ที่มีอยู่ใน subsystem description ที่ระบุได้ สามารถใช้คำสั่งนี้ในขณะที่ระบบย่อยแอ็คทีฟหรือไม่แอ็คทีฟ ถ้าต้องการเปลี่ยน job queue entry ในระบบย่อย ให้ใช้อินเตอร์เฟซแบบอักษร

“การลบ job queue entry” ในหน้า 191

คุณสามารถลบ job queue entries ออกจาก subsystem description โดยใช้อินเตอร์เฟซแบบอักษร งานบนคิวงานยังคงอยู่บนคิว เมื่อ job queue entry ถูกลบออกจาก subsystem description จะไม่สามารถลบ job queue entry ถ้างานแอ็คทีฟในปัจจุบันใดๆ ถูกเริ่มจาก คิวงาน

“การเปลี่ยนจำนวนของงานที่รันได้พร้อมกันในคิวงาน” ในหน้า 208

ระบบย่อย QBASE ถูกจัดส่งมาพร้อมกับ job queue entry สำหรับคิวงาน QBATCH entry นี้อนุญาตให้มีงานแบ็ตช์ที่จะรันได้เพียงหนึ่งงานเท่านั้นในแต่ละครั้ง ถ้าคุณต้องการให้สามารถรันงานแบ็ตช์ได้พร้อมกันหลายงานจากคิวงานนั้น คุณต้องเปลี่ยน job queue entry

วิธีการจัดสรรคิวงานที่ระบบย่อย

สามารถเชื่อมโยงคิวงานกับหลายระบบย่อยได้ แต่คิวงานสามารถถูกจัดสรรให้กับระบบย่อยเพียงหนึ่งระบบเท่านั้นในแต่ละครั้ง. เมื่อระบบย่อยเริ่มต้น, การมอนิเตอร์ระบบย่อยจะพยายามจัดสรรคิวงานแต่ละคิวที่กำหนดใน subsystem job queue entries.

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

หากคิวงานไม่มีอยู่เมื่อเริ่มระบบย่อย, คิวงานจะถูกจัดสรร ที่ระบบย่อยเมื่อเกิดอย่างใดอย่างหนึ่งดังนี้:

- คิวงานถูกสร้างขึ้น.
- คิวงานถูกเปลี่ยนชื่อเป็นชื่อที่กำหนดที่ระบบย่อย.
- คิวงานถูกย้ายไปยังโบบารี่อื่นและชื่อที่ถูกต้องใหม่ ตรงกับชื่อใน subsystem description.
- โบบารี่ที่มีคิวงานถูกเปลี่ยนชื่อ และชื่อที่ถูกต้องใหม่ ตรงกับชื่อใน subsystem description.

หลายคิวงาน

ในหลายกรณี การใช้ QBATCH เพียงหนึ่งคิวงานพร้อมกับค่าดีฟอลต์ งานแอนด์ที่ฟหนึ่งงานจะเพียงพอสำหรับความต้องการของคุณ หากไม่เพียงพอ คุณอาจต้องการมีหลายคิวงานเพื่อให้บางคิวงานแอนด์ที่ฟในระหว่างชั่วโมงทำงานปกติ บางคิวงานสำหรับวัตถุประสงค์พิเศษ และบางคิวงาน แอนด์ที่ฟหลังชั่วโมงทำงานปกติ

ตัวอย่างเช่น คุณสามารถกำหนดคิวงานที่แตกต่างสำหรับ:

งานที่รันนานเพื่อให้คุณสามารถควบคุมจำนวนงานที่แอนด์ที่ฟได้พร้อมกัน

คุณยังอาจต้องการให้งานเหล่านี้ใช้ระดับความสำคัญที่ต่ำกว่างานแบ็คชอื่น

งานข้ามคืนที่ไม่สะดวกที่จะรันในระหว่างชั่วโมงการทำงานปกติ

ตัวอย่างเช่น การรันคำสั่ง Reorganize Physical File Member (RGZPFM) บนไฟล์ฐานข้อมูลขนาดใหญ่ต้องใช้การลออกเฉพาะบนไฟล์ นี้หมายความว่า ผู้ใช้อื่นจะไม่สามารถเข้าใช้งานไฟล์ในขณะที่ทำการดำเนินการนี้ นอกจากนี้ การดำเนินการนี้อาจใช้เวลานาน เป็นการดีกว่าที่จะวางงานนี้ บนคิวงานสำหรับงานซึ่งรันในระหว่างนอกชั่วโมงทำงาน

งานที่มีระดับความสำคัญสูง

คุณอาจต้องการมีคิวงานที่จะส่งงานที่มีระดับความสำคัญสูงทั้งหมดไปที่นั่น จากนั้น คุณสามารถแน่ใจได้ว่างานจะเสร็จสมบูรณ์อย่างรวดเร็วและไม่ถูกหน่วง โดยงานที่มีระดับความสำคัญต่ำ

งานที่ถูกส่งไปยังข้อกำหนดรีซอร์สเฉพาะ เช่น ดิสเก็ต หรือเทป

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

ตัวอย่างเช่น ถ้าใช้เทปสำหรับงานหลายงาน งานทั้งหมดที่ใช้เทปจะถูกวางบนคิวงานเดียว จากนั้น ในแต่ละครั้งจะมีการเลือกงานหนึ่งงานจากคิวงาน สิ่งนี้ทำให้มั่นใจได้ว่า ไม่มีงานสองงานแย่งชิง อุปกรณ์เดียวกันในเวลาเดียวกัน ถ้าเกิดสภาพเช่นนี้งานใดงานหนึ่งจะจบลง ด้วยข้อผิดพลาดการจัดสรร

หมายเหตุ: เทปเอาต์พุตไม่สามารถสพูลได้

งานโปรแกรมเมอร์

คุณอาจต้องการมีคิวงานหนึ่งคิวที่จัดการกับงานโปรแกรมเมอร์หรือชนิด ของงานที่สามารถพักได้ในขณะที่กำลังรันงานการผลิต

การรันแบบเป็นลำดับของชุดงาน

คุณอาจมีแอฟพลิเคชันซึ่งงานหนึ่งงานขึ้นอยู่กับความสำเร็จ ของงานอื่น ถ้าคุณวางงานเหล่านี้บนคิวงานที่เลือก และรันงานหนึ่งงาน ในแต่ละครั้ง ลักษณะเช่นนี้ทำให้มั่นใจได้ว่าการรันงานเหล่านี้จะเป็นไปตามลำดับ

ถ้างาน ต้องการการควบคุมเฉพาะของไฟล์ คุณอาจต้องการวางแผนงานบนคิวงาน เมื่อคิวเป็นเพียงคิวเดียวที่แอสคทีฟบนเซิร์ฟเวอร์ เช่นในระหว่างช่วง กลางคืนหรือวันหยุด

ถ้าคุณใช้คิวงานหลายคิว คุณจะพบว่า การควบคุมคิวงานหลากหลายคิว เป็นสิ่งที่ต้องพิจารณาเป็นอันดับแรก โดยปกติ คุณจะ ต้องควบคุม:

- จำนวนคิวงานที่มีอยู่
- จำนวนคิวงานที่แอสคทีฟในระบบย่อยในเวลาเดียวกัน
- จำนวนงานแอสคทีฟที่สามารถเลือกจากคิวงานในเวลาหนึ่ง
- จำนวนงานที่สามารถแอสคทีฟในระบบย่อยในเวลาหนึ่ง

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) นอกจากนี้ คุณยังสามารถควบคุมจำนวนงานของแต่ละระดับความสำคัญที่สามารถแอสคทีฟได้โดยใช้พารามิเตอร์ Maximum active priority MAXACTx ตัวอย่างเช่น ถ้า MAXACT=10, MAXACT5=2 และมีงานบนคิวงานที่มีระดับความสำคัญ 5 อยู่สามงาน จะมีเพียงสองงานเท่านั้นที่สามารถแอสคทีฟได้ในขณะใดๆ

ระบบย่อยจะประมวลผลงานจากคิวงานที่มีหมายเลขลำดับน้อยที่สุดก่อน เมื่องานทั้งหมดที่อยู่บนคิวงานได้ถูกประมวลผลแล้ว หรือเมื่อใช้งานถึงจำนวนสูงสุดของงานจากคิว ระบบย่อยจะประมวลผลงานจากคิวที่มีหมายเลขลำดับสูงกว่าถัดไป

ลำดับดำเนินต่อไปจนกว่าระบบย่อยได้ประมวลผล job queue entries ที่พร้อมใช้งานทั้งหมด หรือจนกว่าระบบย่อยทำงานถึงขีดจำกัดของงานที่สามารถรันหรือรอในระบบย่อย จำนวนของงานที่สามารถรัน หรือรอได้ถูกกำหนดโดยพารามิเตอร์ Maximum active jobs (MAXACT) parameter ใน subsystem description ในบางกรณี ลำดับอาจถูกอินเตอร์รัปต์ เมื่องานจบหรือถูกถ่ายโอน การสร้าง, การพัก และการรีลีสคิวงาน สามารถเปลี่ยนลำดับของคิวงานที่ประมวลผลได้ด้วย

งานที่เกี่ยวข้อง

“การวางแผนงานบนคิวงาน” ในหน้า 213

งานถูกวางแผนบนคิวงานโดยการย้ายงานที่มีอยู่จากคิวงานหนึ่งไปยังคิวงานอื่น หรือโดยการส่งงานใหม่ อย่างเป็นอิสระหนึ่ง ใช้ System i Navigator เพื่อย้ายงานระหว่างคิวต่างๆ ใช้อินเตอร์เฟซแบบอักษรเพื่อส่งงานใหม่

“การย้ายงานไปที่คิวงานอื่น” ในหน้า 212

มีหลายเหตุผลที่คุณอาจต้องการย้ายงานไปที่คิวอื่น ตัวอย่างเช่น ในบางครั้งงานอาจถูก backlogged ในคิวเนื่องจากเป็นงานที่ใช้เวลารันนาน บางครั้งรันใหม่ที่จัดตารางเวลาของงานขัดแย้งกับงานใหม่ที่มีระดับความสำคัญสูงกว่า วิธีหนึ่งในการจัดการสถานการณ์นี้คือ การย้ายงานที่รอไปยังคิวอื่นที่ไม่ยุ่ง

“การเปลี่ยนจำนวนของงานที่รันได้พร้อมกันในคิวงาน” ในหน้า 208

ระบบย่อย QBASE ถูกจัดส่งมาพร้อมกับ job queue entry สำหรับคิวงาน QBATCH entry นี้อนุญาตให้มีงานแบ็คชั้ที่จะรันได้เพียงหนึ่งงานเท่านั้นในแต่ละครั้ง ถ้าคุณต้องการให้สามารถรันงานแบ็คชั้ได้พร้อมกันหลายงานจากคิวงานนั้น คุณต้องเปลี่ยน job queue entry

การรักษาความปลอดภัยคิวงาน

คุณสามารถรักษาระดับของการรักษาความปลอดภัยที่คิวงานของคุณโดยอนุญาตเฉพาะบางคน (โปรไฟล์ผู้ใช้) ให้สามารถเข้าใช้คิวงานนั้น โดยทั่วไป มีสามวิธี ที่ผู้ใช้สามารถได้สิทธิในการควบคุมคิวงาน (เช่น พักหรือรีสตาร์ทคิวงาน)

- ผู้ใช้ได้รับการกำหนดสิทธิควบคุมการสพูล (SPCAUT(*SPLCTL)) ในโปรไฟล์ผู้ใช้
- ผู้ใช้ได้รับการกำหนดสิทธิในการควบคุมงาน (SPCAUT(*JOBCTL)) ในโปรไฟล์ผู้ใช้ และคิวงานสามารถถูกควบคุมโดยผู้ควบคุมเครื่อง (OPRCTL(*YES))
- ผู้ใช้มีสิทธิอ็อบเจกต์ที่ต้องการที่คิวงาน สิทธิอ็อบเจกต์ที่ต้องการถูกระบุโดยพารามิเตอร์ AUTCHK บนคำสั่ง CRTJOBQ ค่าของ *OWNER บ่งชี้ว่า เฉพาะเจ้าของของคิวงานเท่านั้นที่ได้รับสิทธิผ่านทางสิทธิอ็อบเจกต์สำหรับ คิวงาน ค่าของ *DTAAUT บ่งชี้ว่าผู้ใช้ที่มีสิทธิ *CHANGE สำหรับคิวงานจะได้รับสิทธิในการควบคุมคิวงาน

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

วิธีการให้สิทธิสามวิธีเหล่านี้ใช้เฉพาะกับคิวงาน ไม่ได้ใช้กับ งานบนคิวงาน กฎสิทธิปกติสำหรับการควบคุมงาน ใช้กับงานที่อยู่บนคิวงานและงานที่กำลังรันอยู่ในปัจจุบัน

เอาต์พุตคิว

เอาต์พุตคิวคือพื้นที่ที่ printer output files (หรือเรียกว่าสพูลไฟล์) รอกการถูกประมวลผลและถูกส่งไปยังพริ้นเตอร์ พริ้นเตอร์เอาต์พุตจะถูกสร้างขึ้นทั้งโดยระบบหรือโดยผู้ใช้ที่ใส่ไฟล์พิมพ์ อย่างไรก็ตาม

ไฟล์พิมพ์คล้ายกับเท็มเพลต หรือต้นแบบที่มีการเซตค่าดีฟอลต์สำหรับแอตทริบิวต์ของพริ้นเตอร์เอาต์พุต ไว้ให้แล้ว และเป็นจุดเริ่มต้นของช่วงชีวิตพริ้นเตอร์เอาต์พุต

ไฟล์พิมพ์มีแอตทริบิวต์เอาต์พุตคิว (OUTQ) และอุปกรณ์การพิมพ์ (DEV) ซึ่งบ่งชี้ว่าพริ้นเตอร์เอาต์พุตจะถูกกำหนดทิศทางอย่างไร โดยปกติ การตั้งค่า ดีฟอลต์คือ *JOB ซึ่งหมายความว่าแอตทริบิวต์งานของเอาต์พุตคิว และอุปกรณ์การพิมพ์เป็นตัวกำหนดทิศทางของพริ้นเตอร์เอาต์พุต แอตทริบิวต์งานของเอาต์พุตคิวและค่าอุปกรณ์การพิมพ์มาจากข้อมูลที่รับจากงานที่ถูกสร้างขึ้น ข้อมูลนี้มาจากข้อมูลที่ได้มาจาก โปรไฟล์ผู้ใช้ที่ทำงานกำลังรันอยู่ภายใต้รายละเอียดของงาน, workstation device description และค่ากำหนดของระบบ Printer device description (QPRTEDEV)

เมื่อพริ้นเตอร์เอาต์พุตพร้อมที่จะถูกสร้าง ระบบจะตรวจสอบไฟล์พิมพ์และแอตทริบิวต์งาน (ในลำดับนี้) เพื่อดูว่าเอาต์พุตคิวใดจะประมวลผลพริ้นเตอร์เอาต์พุตและอุปกรณ์การพิมพ์ใดที่ระบบจะใช้งาน คุณสามารถเปลี่ยนพารามิเตอร์ของเอาต์พุตคิว (OUTQ) และอุปกรณ์การพิมพ์ (DEV) ขณะทำงานถูกส่งหรือขณะรันงานเพื่อข้ามการประมวลผลแบบขยาย ยกตัวอย่าง ผู้ใช้สามารถตั้งค่าเอาต์พุตคิวของไฟล์พิมพ์ให้เป็นคิวเฉพาะได้ และตั้งค่าอุปกรณ์การพิมพ์เป็นเครื่องพิมพ์ที่ต้องการในไฟล์พิมพ์เมื่อเริ่มต้นทำงานเพื่อให้การเปลี่ยนแปลงมีผลในทันที ในการทำเช่นนี้ พริ้นเตอร์เอาต์พุตไม่ต้องเข้าไปยังแอตทริบิวต์งานเพื่อค้นหาเอาต์พุตคิวและ อุปกรณ์การพิมพ์ที่จะใช้งาน หากไม่พบเอาต์พุตคิวที่ระบุ พริ้นเตอร์เอาต์พุตจะถูกส่งไปที่ QGPL/QPRINT สำหรับข้อมูลเพิ่มเติม เกี่ยวกับวิธีการสร้างพริ้นเตอร์เอาต์พุต ให้ดูบทที่ 1 ของคู่มือ Printer Device Programming

Printer output files คือไฟล์ที่มีข้อมูลซึ่งกำลังรอที่จะพิมพ์หรือถูกประมวลผล printer output file มีแอตทริบิวต์ที่สำคัญซึ่งกำหนดตำแหน่งของพริ้นเตอร์เอาต์พุตบนคิวซึ่งสัมพันธ์กับพริ้นเตอร์เอาต์พุตอื่น ตำแหน่งจะถูกกำหนดโดยแอตทริบิวต์ระดับความสำคัญ, สถานะ และตารางเวลา

เอาต์พุตคิว

เอาต์พุตคิว คืออ็อบเจกต์ที่ประกอบด้วยรายการของ printer output file ซึ่งจะถูกบันทึกไปที่อุปกรณ์เอาต์พุต เอาต์พุตคิวมีแอ็ททริบิวต์ที่สำคัญซึ่งกำหนดลำดับที่พรินเตอร์เอาต์พุตจะถูกประมวลผลและสิทธิที่ต้องใช้ในการทำการเปลี่ยนแปลงที่ printer output file

ระดับความสำคัญ

พรินเตอร์เอาต์พุตที่กำลังรอการประมวลผลจะถูกย้ายไปที่เอาต์พุตคิวตามระดับความสำคัญของเอาต์พุต (ตั้งแต่ 1-9 โดย 1 คือระดับความสำคัญสูงสุด)

สถานะ สถานะปัจจุบันของพรินเตอร์เอาต์พุต คุณสามารถดูสถานะนี้จากหน้า General ของหน้าต่าง Output properties

ตารางเวลา

แอ็ททริบิวต์ตารางเวลาจะบอกว่าเมื่อใดที่ไฟล์ควรเริ่มการพิมพ์แบบฟิสิกส์ข้อมูลเอาต์พุต

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 .

หลักการที่เกี่ยวข้อง

“การจัดการเอาต์พุตคิว” ในหน้า 214

เอาต์พุตคิวจะช่วยคุณจัดการพรินเตอร์เอาต์พุตที่สร้างขึ้นเมื่อสิ้นสุดการทำงาน เป็นสิ่งสำคัญที่ต้องทำความเข้าใจวิธีการรักษาเอาต์พุตคิวของคุณอย่างมีประสิทธิภาพ เพื่อให้เอาต์พุตที่พิมพ์ของคุณดำเนินไปอย่างราบรื่น

ข้อมูลที่เกี่ยวข้อง

รายงานจากประสบการณ์: ข้อควรพิจารณาสำหรับประสิทธิภาพของสพูล

การพิมพ์ระดับต้น

แอ็ททริบิวต์ของเอาต์พุตคิว

เอาต์พุตคิวควบคุมวิธีการประมวลผลไฟล์พรินเตอร์เอาต์พุต (เรียกอีกอย่างว่าสพูลไฟล์) และผู้มีสิทธิปฏิบัติการบนเอาต์พุตคิวและพรินเตอร์เอาต์พุตที่เกี่ยวข้อง

เนื่องจากข้อมูลส่วนใหญ่ที่คุณพิมพ์บนระบบจะถูกสร้างเป็นพรินเตอร์เอาต์พุต การรักษาความปลอดภัยจึงเป็นสิ่งจำเป็นเพื่อป้องกันไม่ให้ผู้ใช้ที่ไม่ได้รับอนุญาตเข้าถึงเอกสารที่เป็นความลับ หรือสำคัญ สิทธิการตรวจสอบ, การให้สิทธิในข้อมูล, การควบคุมของผู้ควบคุมเครื่อง, การควบคุมสพูล หรือการเป็นเจ้าของ ช่วยให้คุณสามารถเข้าใช้งานและทำการเปลี่ยนแปลงเอาต์พุตคิวหรือ printer output file คุณต้องใช้สิทธิอย่างใดอย่างหนึ่งต่อไปนี้เพื่อปฏิบัติ action ใดๆ บนเอาต์พุตคิว หรือพรินเตอร์เอาต์พุต:

สิทธิการตรวจสอบ

คุณต้องเป็นเจ้าของคิวหรือมีสิทธิในข้อมูล

การแสดงข้อมูล

เมื่อสิทธิ์นี้ถูกเซตเป็น *YES สิทธิจะทำให้คุณสามารถดำเนินการต่างๆ เช่น การดู, การย้าย, การส่งเอาต์พุตไปยังระบบอื่น และการทำสำเนาพรินเตอร์เอาต์พุต

การควบคุมของผู้ควบคุมเครื่อง

หากแอตทริบิวต์นี้ถูกเซตเป็น *YES ผู้ใช้ที่มีสิทธิ์พิเศษ *JOBCTL จะได้รับสิทธิให้ดำเนินการต่างๆ เช่น การพัก, การรีลีส และการลบพรินเตอร์เอาต์พุตจากเอาต์พุตคิว การดำเนินการอื่นบนพรินเตอร์เอาต์พุต, เอาต์พุตคิว และ writer ด้วย

การควบคุมสพูล

อนุญาตให้ผู้ใช้ดำเนินการต่างๆ บนพรินเตอร์เอาต์พุต ผู้ใช้ต้องมีสิทธิ์ใช้งาน *EXECUTE กับไลบรารีที่มีเอาต์พุตคิวอยู่เพื่อดำเนินการต่างๆ บนเอาต์พุตคิว

เจ้าของ

อนุญาตให้ผู้ใช้ที่เป็นเจ้าของเอาต์พุตคิวเปลี่ยนหรือลบพรินเตอร์เอาต์พุต

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

สำหรับข้อมูลเพิ่มเติมเกี่ยวกับ i5/OS สิทธิในการใช้งาน ให้อูที่ สิทธิที่จำเป็นสำหรับอ็อบเจกต์ที่ใช้โดยคำสั่งในกลุ่มหัวข้อ Security reference

ลำดับของไฟล์

แอตทริบิวต์ Order of files on the queue (SEQ) กำหนดว่าพรินเตอร์เอาต์พุตจะถูกเอาต์พุตคิวประมวลผลอย่างไร

แอตทริบิวต์นี้มีสองค่า:

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

หมายเหตุ: คุณสามารถเปลี่ยนได้เฉพาะลำดับเอาต์พุตคิวของไฟล์แอตทริบิวต์เมื่อไม่มี printer output files อยู่บนคิว

สพูลไฟล์

การสพูลคือฟังก์ชันระบบที่บันทึกข้อมูลสำหรับการประมวลผลหรือ การพิมพ์ในภายหลัง ข้อมูลนี้ถูกจัดเก็บในสพูลไฟล์ สพูลไฟล์ทำงานใน ลักษณะคล้ายกับเทปไฟล์หรือไฟล์อุปกรณ์อื่น สพูลไฟล์ช่วยให้คุณจัดการข้อมูลของคุณซึ่งถูกกำหนดเป้าหมายสำหรับอุปกรณ์ที่ต่อพ่วงภายนอก เช่น พรินเตอร์

ฟังก์ชันการสพูลช่วยผู้ใช้เซิร์ฟเวอร์ให้สามารถจัดการการดำเนินการ อินพุตและเอาต์พุตได้อย่างมีประสิทธิภาพมากขึ้น เซิร์ฟเวอร์สนับสนุนการสพูลสองชนิด, การสพูลเอาต์พุตและการสพูลอินพุต การสพูลเอาต์พุตสามารถใช้สำหรับอุปกรณ์การพิมพ์ การสพูลอินพุตใช้กับอินพุตไฟล์ฐานข้อมูล

ข้อมูลที่เกี่ยวข้อง

สพูลไฟล์และเอาต์พุตคิว

การสพูลเอาต์พุต:

สามารถใช้การสพูลเอาต์พุตสำหรับทั้งพริ้นเตอร์และอุปกรณ์ดิสเก็ต การสพูลเอาต์พุตจะส่งเอาต์พุตงานไปที่หน่วยเก็บของดิสก์ แทนที่จะส่งตรงไปที่พริ้นเตอร์หรืออุปกรณ์ดิสเก็ตเอาต์พุต การสพูลเอาต์พุตช่วยให้งานที่สร้างเอาต์พุต สามารถประมวลผลต่อไปได้โดยไม่ต้องพิจารณาถึงความเร็ว หรือสภาพพร้อมใช้งานของอุปกรณ์เอาต์พุต

นอกจากนี้ การสพูลเอาต์พุตยังช่วยให้เซิร์ฟเวอร์สามารถสร้างเอาต์พุตบนหลาย อุปกรณ์เอาต์พุต เช่น พริ้นเตอร์และอุปกรณ์ดิสเก็ตได้อย่างมีประสิทธิภาพ 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 ของรูปแบบเอาต์พุต และรายการของแอ็ททริบิวต์ที่อธิบายว่าเซิร์ฟเวอร์ควรจะประมวลผลสพูลไฟล์อย่างไร

ฟังก์ชันการสพูลเอาต์พุตถูกดำเนินการโดยเซิร์ฟเวอร์โดยไม่ต้อง การดำเนินการพิเศษใดๆ โดยโปรแกรมที่สร้างเอาต์พุต เมื่อไฟล์อุปกรณ์ถูกเปิด โดยโปรแกรม, ระบบปฏิบัติการจะกำหนดว่าเอาต์พุตจะถูกสพูล หรือไม่ เมื่อไฟล์พริ้นเตอร์ที่ระบุการสพูลถูกเปิด สพูลไฟล์ที่มีเอาต์พุตของโปรแกรมจะถูกวางบนเอาต์พุตคิวที่เหมาะสมในเซิร์ฟเวอร์

สพูลไฟล์สามารถพร้อมใช้งานสำหรับการพิมพ์เมื่อไฟล์พริ้นเตอร์ ถูกเปิด เมื่อไฟล์พริ้นเตอร์ถูกปิด หรือเมื่อจบงาน เครื่องเขียนพริ้นเตอร์ ถูกเริ่มต้นในระบบย่อยการสพูลเพื่อส่งเร็คคอร์ดไปที่พริ้นเตอร์ สพูลไฟล์จะถูกเลือกจากเอาต์พุตคิว

การสพูล device description

Device description ต้องถูกสร้างขึ้นสำหรับแต่ละพริ้นเตอร์และอุปกรณ์ดิสเก็ต เพื่อกำหนดอุปกรณ์นั้น ไปที่เซิร์ฟเวอร์ Printer device description ถูกสร้างขึ้นโดยใช้คำสั่ง Create Device Description for Printer (CRTDEVPR); diskette device description ถูกสร้างขึ้นโดยใช้คำสั่ง Create Device Description for Diskette (CRTDEVDKT)

การเปลี่ยนทิศทางไฟล์ของสพูลไฟล์

การเปลี่ยนทิศทางไฟล์ จะเกิดขึ้นเมื่อสพูลไฟล์ถูกส่งไปที่อุปกรณ์เอาต์พุตที่ไม่ใช่อุปกรณ์เอาต์พุตที่ตั้งใจไว้ตั้งแต่แรก การเปลี่ยนทิศทางของไฟล์อาจเกี่ยวข้องกับอุปกรณ์ที่ประมวลผลสื่อบันทึกต่างๆ (เช่น พริ้นเตอร์เอาต์พุตที่ส่งไปที่อุปกรณ์ดิสเก็ต) หรืออุปกรณ์ที่ประมวลผลสื่อบันทึกชนิดเดียวกัน แต่มีชนิดของอุปกรณ์ต่างกัน (เช่น พริ้นเตอร์เอาต์พุต 5219 ถูกส่งไปยังพริ้นเตอร์ 4224)

ขึ้นอยู่กับอุปกรณ์เอาต์พุต ใหม่สำหรับสพูลไฟล์ ไฟล์อาจสามารถถูกประมวลผล เหมือนกับที่เคยใช้บนอุปกรณ์ที่ระบุในครั้งแรก อย่างไรก็ตาม อุปกรณ์ที่แตกต่างกันมักทำให้เอาต์พุตถูกจัดรูปแบบแตกต่างกัน ในกรณีนี้ เซิร์ฟเวอร์จะส่งข้อความสอบ

ถามไปที่ message queue ของ writer เพื่อแจ้งให้คุณทราบถึงสถานการณ์และคุณสามารถระบุได้ว่า คุณต้องการพิมพ์ต่อไปหรือไม่

เอาต์พุตคิวและสพูลไฟล์:

การประมวลผลงานแบตช์และงานแบบโต้ตอบอาจทำให้เกิด spooled output records ที่จะถูกประมวลผลบนอุปกรณ์เอาต์พุต เช่น พรินเตอร์หรือ ดิสเก็ตไดรฟ์ เอาต์พุตเรกคอร์ดเหล่านี้ถูกจัดเก็บในสพูลไฟล์จนกว่าจะสามารถประมวลผลได้งานเดียว สามารถมีหลายสพูลไฟล์ได้

เมื่อสพูลไฟล์ถูกสร้างขึ้น ไฟล์จะถูกวางบนเอาต์พุตคิว แต่ละเอาต์พุตคิวมีรายการที่จัดลำดับของสพูลไฟล์ งานอาจมีสพูลไฟล์บนหนึ่งหรือหลายเอาต์พุตคิว สพูลไฟล์ทั้งหมดบนเอาต์พุตคิวหนึ่ง ควรมีชุดของแอตทริบิวต์เอาต์พุตที่ใช้ร่วมกัน เช่น อุปกรณ์, form type และจำนวนบรรทัดต่อนี้ การใช้แอตทริบิวต์ร่วมกันบนเอาต์พุตคิว ช่วยลดจำนวนของ intervention ที่ต้องการและเพิ่มปริมาณงานของอุปกรณ์

ข้อมูลต่อไปนี้เป็นรายการของพารามิเตอร์รับคำสั่ง Create Output Queue (CRTOUTQ) และสิ่งที่พารามิเตอร์ระบุ:

- MAXPAGES: ระบุขนาดสพูลไฟล์สูงสุดในหน้าที่อนุญาตให้พิมพ์ได้ระหว่างเวลาเริ่มต้น และเวลาสิ้นสุดของวัน
- AUTOSTRWTR: ระบุจำนวน writer ที่จะถูกเริ่มโดยอัตโนมัติที่เอาต์พุตคิวนี้
- DSPDTA: การมีหรือไม่มีสิทธิพิเศษใดๆของผู้ใช้ แต่จะมีสิทธิ *USE ที่เอาต์พุตคิวสามารถแสดง, ก๊อปปี้ หรือส่งเนื้อหาของสพูลไฟล์ที่ไม่ใช่ของตนได้ ถ้ามีการระบุ *OWNER สำหรับ DSPDTA นั้นหมายความว่าเฉพาะเจ้าของของไฟล์หรือผู้ใช้ที่มีสิทธิพิเศษ *SPLCTL เท่านั้นที่สามารถแสดง, ก๊อปปี้ หรือส่งไฟล์ได้
- JOBSEP: จำนวนหน้าต่างตัวแบ่งงาน ถ้ามี ที่จะพิมพ์ระหว่างเอาต์พุตของแต่ละงาน เมื่อพิมพ์เอาต์พุต
- DTAQ: คิวข้อมูลที่เชื่อมโยงกับเอาต์พุตคิวนี้ ถ้าระบุ entry จะถูกส่งไปที่คิวข้อมูลเมื่อใดก็ตามที่สพูลไฟล์ที่ ready status บนคิว
- OPRCTL: การมีหรือไม่มีสิทธิในการควบคุมงานของผู้ใช้เป็นตัวควบคุมเอาต์พุตคิว (ตัวอย่างเช่น ผู้ใช้สามารถพักเอาต์พุตคิวได้หรือไม่)
- SEQ: ควบคุมลำดับซึ่งสพูลไฟล์จะถูกเรียงลำดับบนเอาต์พุตคิว
- AUTCHK: ระบุชนิดของสิทธิที่เอาต์พุตคิวที่จะเปิดทางให้ผู้ใช้สามารถควบคุมสพูลไฟล์บนเอาต์พุตคิว (ตัวอย่างเช่น เปิดทางให้ผู้ใช้พักสพูลไฟล์บนเอาต์พุตคิว)
- AUT: สิทธิพับลิค ระบุว่าผู้ใช้สามารถควบคุมอะไร บนตัวเอาต์พุตคิวเอง
- TEXT: รายละเอียดข้อความ ข้อความที่ยาวได้ถึง 50 อักขระ ซึ่งอธิบายเอาต์พุตคิว

ดีฟอลต์เอาต์พุตคิวของระบบ:

ดีฟอลต์บนคำสั่ง CL ใช้ดีฟอลต์เอาต์พุตคิวสำหรับพรินเตอร์ระบบเป็นดีฟอลต์เอาต์พุตคิวสำหรับเอาต์พุตที่สพูลทั้งหมด พรินเตอร์ระบบถูกกำหนดโดยค่าเซิร์ฟเวอร์ QPRTDEV

เมื่อสพูลไฟล์ถูกสร้างขึ้นโดยการเปิดไฟล์อุปกรณ์ และไม่พบเอาต์พุตคิวที่ระบุสำหรับไฟล์ ระบบจะพยายามระบุสพูลไฟล์บนเอาต์พุตคิว QPRINT ในไลบรารี QGPL ถ้ามีบางเหตุผลที่ทำให้ไม่สามารถวางสพูลไฟล์บนเอาต์พุตคิว QPRINT จะมีการส่งข้อความแสดงความผิดพลาด และเอาต์พุตจะไม่ถูกสพูล

มีการจัดเอาต์พุตคิวต่อไปนี้ให้

- 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 คือ i5/OS โปรแกรมที่จะนำสพูลไฟล์จากเอาต์พุตคิว และจัดทำสพูลไฟล์บนอุปกรณ์เอาต์พุต สพูลไฟล์ที่ถูกวางบนเอาต์พุตคิวแล้ว จะยังคงถูกเก็บในระบบจนกว่า writer จะถูกเริ่มที่เอาต์พุตคิว

writer นำสพูลไฟล์มาครั้งละหนึ่งไฟล์จากเอาต์พุตคิวตาม ระดับความสำคัญของสพูลไฟล์ writer จะประมวลผลสพูลไฟล์เฉพาะถ้า entry ของสพูลไฟล์บนเอาต์พุตคิวระบุว่า สพูลไฟล์มีสถานะพร้อมใช้งาน (RDY) คุณสามารถแสดงสถานะของสพูลไฟล์ได้โดยใช้คำสั่ง Work with Output Queue (WRKOUTQ)

ถ้าสพูลไฟล์มีสถานะพร้อมใช้งาน writer จะนำ entry มาจาก เอาต์พุตคิวและพิมพ์งานที่ระบุ หรือตัวแบ่งไฟล์ หรือทั้งสองอย่างตามด้วยข้อมูลเอาต์พุตในไฟล์ ถ้าสพูลไฟล์ไม่ได้มีสถานะพร้อมใช้งาน writer จะปล่อย entry ไว้บนเอาต์พุตคิวและไปที่ entry ถัดไปในกรณีส่วนใหญ่ writer จะประมวลผลสพูลไฟล์ต่อไป (มีงาน และตัวแบ่งไฟล์ก่อน) จนกว่าไฟล์ทั้งหมดที่มีสถานะพร้อมใช้งานได้ถูกนำออกจากเอาต์พุตคิว

พารามิเตอร์ AUTOEND บนคำสั่งเริ่ม writer เป็นตัวกำหนดว่า writer จะยังคงรอสพูลไฟล์ใหม่พร้อมถูกบันทึก จะจบลงหลังการประมวลผลหนึ่งไฟล์ หรือจบลงหลังจากที่สพูลไฟล์ทั้งหมดที่มีสถานะพร้อมใช้งานได้ถูกนำออกจากเอาต์พุตคิวแล้ว

คำสั่ง spooling writer:

ต่อไปนี้เป็นคำสั่งต่างๆ ที่คุณสามารถใช้ในการควบคุม spooling writer

- **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 ที่จะส่งสพูลไฟล์จากเอาต์พุตคิวไปที่ระบบรีโมต
- **Change Writer (CHGWTR):** เปลี่ยนแอตทริบิวต์ 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 ซึ่งเป็นส่วนที่เลือกติดตั้งได้ของ i5/OS

ข้อมูลที่เกี่ยวข้อง

คำสั่ง Start Printer Writer (STRPRTWTR)

คำสั่ง Start Remote Writer (STRRMTWTR)

คำสั่ง Change Writer (CHGWTR)

คำสั่ง Hold Writer (HLDWTR)

คำสั่ง Release Writer (RLSWTR)

คำสั่ง End Writer (ENDWTR)

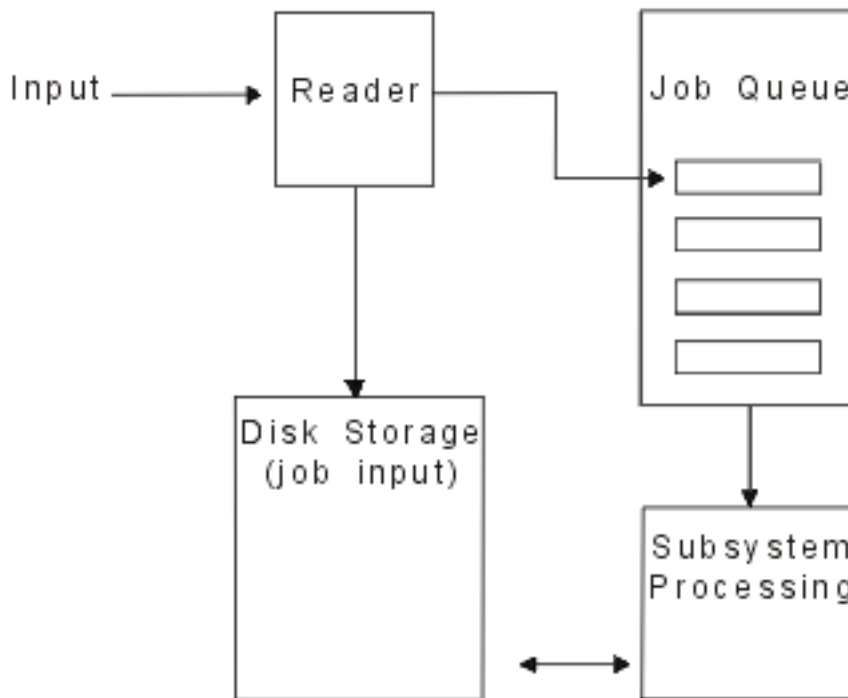
Input spooling:

Input spooling นำข้อมูลมาจากอุปกรณ์อินพุต, จัดเตรียม งานสำหรับการจัดตารางเวลา และวาง entry ในคิวงาน ด้วยการใช้ input spooling โดยปกติสามารถช่วยให้รันไทม์ของงานลดลง, เพิ่มจำนวนของงานที่สามารถ รันในลำดับถัดไปได้ และทำให้ ปริมาณงานของอุปกรณ์มากขึ้น

องค์ประกอบหลักของ input spooling มีดังนี้:

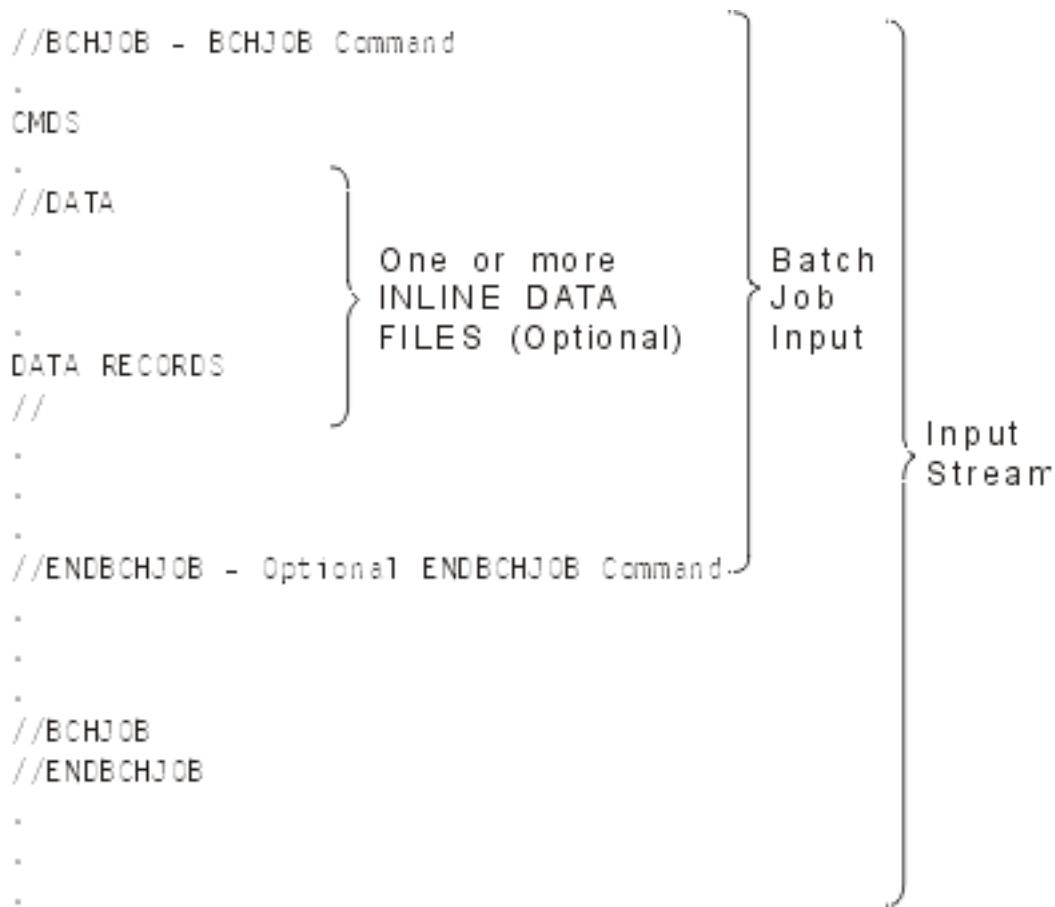
- คิวงาน: รายการที่จัดลำดับของงานแบ็ตช์ที่ส่งไปที่ระบบสำหรับการรัน และเป็นที่ยังงานแบ็ตช์ถูกเลือกมาเพื่อรัน
- เครื่องอ่าน: ฟังก์ชันที่นำงานมาจากอุปกรณ์อินพุตหรือไฟล์ฐานข้อมูล และวางงานบนคิวงาน

เมื่อเครื่องอ่านอ่านงานแบ็ตช์จากอินพุตซอร์ส, คำสั่งใน input stream จะถูกจัดเก็บในระบบเป็นคำขอสำหรับงาน, ข้อมูล อินไลน์ถูกสพูลเป็นไฟล์ข้อมูลอินไลน์ และ entry สำหรับงานถูกวางบนคิวงาน ข้อมูลงานยังคงถูกจัดเก็บอยู่ในระบบที่วาง เครื่องอ่านไว้นานกว่า job entry จะถูกเลือกจากคิวงานสำหรับการประมวลโดยระบบย่อย



รูปที่ 2. ความสัมพันธ์ขององค์ประกอบ input spooling

คุณสามารถใช้ฟังก์ชันเครื่องอ่านเพื่ออ่าน input stream จากดิสเก็ต หรือไฟล์ฐานข้อมูล



รูปที่ 3. การจัดระบบตามปกติของ input stream

คิวงานที่งานวางอยู่ถูกระบุอยู่บนพารามิเตอร์ JOBQ บนคำสั่ง Batch Job BCHJOB หรือ Start Database Reader STRDBRDR command หรือในรายละเอียดของงาน คำต่างๆของพารามิเตอร์ JOBQ สำหรับคำสั่ง BCHJOB เป็นดังนี้:

- *RDR: คิวงานจะถูกเลือกจากพารามิเตอร์ JOBQ บนคำสั่ง STRDBRDR
- *JOBQ: คิวงานจะถูกเลือกจากพารามิเตอร์ JOBQ ในรายละเอียดของงาน
- คิวงานเฉพาะ: จะใช้คิวที่ระบุ

สำหรับงานที่มี input streams เล็ก คุณอาจพัฒนาประสิทธิภาพของระบบได้โดยการไม่ใช้ input spooling คำสั่ง Submit Job (SBMJOB) จะอ่าน input stream และวางงานบนคิวงานในระบบย่อยที่เหมาะสม โดยการข้ามระบบย่อยการสพูล และการดำเนินการของเครื่องอ่าน

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

คำสั่งของการอินพุตงาน:

คุณสามารถใช้คำสั่งเหล่านี้ในการส่งงานไปยังระบบ สามารถใช้คำสั่ง start reader สำหรับการสพูลอินพุตงาน; คำสั่ง submit job จะไม่ใช้การสพูล

- Batch Job (BCHJOB): ทำเครื่องหมายจุดเริ่มต้นองงานใน batch input stream และกำหนดลักษณะการดำเนินการของงาน
- Data (DATA): ทำเครื่องหมายจุดเริ่มต้นของไฟล์ข้อมูลอินไลน์

- End Batch Job (ENDBCHJOB): ทำเครื่องหมายจุดสิ้นสุดของ batch input stream
- End Input (ENDINP): ทำเครื่องหมายจุดสิ้นสุดของ batch input stream
- Submit Database Jobs (SBMDBJOB): อ่าน input stream จากไฟล์ฐานข้อมูล และวางแผนงานใน input stream บนคิวงานที่เหมาะสม
- Submit Diskette Jobs (SBMDKTJOB): อ่าน input stream จากดิสเก็ตต์ และวางแผนงานใน input stream บนคิวงานที่เหมาะสม
- Start Database Reader (STRDBRDR): เริ่มต้นเครื่องอ่านเพื่ออ่าน input stream จากไฟล์ฐานข้อมูล และวางแผนงานใน input stream บนคิวงานที่เหมาะสม
- Start Diskette Reader (STRDKTRDR): เริ่มต้นเครื่องอ่านเพื่ออ่าน input stream จากดิสเก็ตต์ และวางแผนงานใน input stream บนคิวงานที่เหมาะสม

ข้อมูลที่เกี่ยวข้อง

ตัวค้นหาคำสั่ง CL

คำสั่ง Batch Job (BCHJOB)

คำสั่ง Data (DATA)

คำสั่ง End Batch Job (ENDBCHJOB)

คำสั่ง End Input (ENDINP)

คำสั่ง Submit Data Base Jobs (SBMDBJOB)

คำสั่ง Start Data Base Reader (STRDBRDR)

ไฟล์ข้อมูลออนไลน์:

ไฟล์ข้อมูลออนไลน์คือไฟล์ข้อมูลทั้งหมดอยู่เป็นส่วนหนึ่งของงานแบ็ตช์ เมื่องานถูกอ่านโดยเครื่องอ่านหรือคำสั่งการส่งงาน คุณใช้ SBMDBJOB หรือ STRDBRDR เพื่อจัดคิว CL batch stream (stream ของคำสั่ง CL ที่จะรัน) CL batch stream นั้นสามารถรวมข้อมูลที่จะวางเข้าไฟล์ออนไลน์ (ไฟล์ชั่วคราว) เมื่องานสิ้นสุดลงไฟล์ข้อมูลออนไลน์จะถูกลบออก

ไฟล์ข้อมูลออนไลน์ถูกค้นในงานด้วยคำสั่ง //DATA ที่ตอนต้นของไฟล์ และถูกค้นโดย end-of-data delimiter ที่ตอนท้ายของไฟล์

end-of-data delimiter อาจเป็นสตริงอักขระที่ผู้ใช้กำหนด หรือค่าดีฟอลต์ของ // // ต้องปรากฏในตำแหน่ง 1 และ 2 ถ้าข้อมูลของคุณมี // ในตำแหน่ง 1 และ 2 คุณควรจะใช้เซตของอักขระที่ไม่ซ้ำกัน เช่น // *** END OF DATA เพื่อระบุว่าสิ่งนี้เป็น end-of-data delimiter ที่ไม่ซ้ำกัน พารามิเตอร์ ENDCHAR บนคำสั่ง //DATA ควรเป็นโค้ด เช่น:

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

หมายเหตุ: ไฟล์ข้อมูลออนไลน์สามารถเข้าถึงได้เฉพาะในระหว่างขั้นตอนการเรอต์แรกของงานแบ็ตช์เท่านั้น ถ้างานแบ็ตช์มีคำสั่ง Transfer Job (TFRJOB), Reroute Job (RRTJOB) หรือ Transfer Batch Job (TFRBCHJOB) ไฟล์ข้อมูลออนไลน์จะไม่สามารถเข้าถึงได้ในขั้นตอนการเรอต์ใหม่.

ไฟล์ข้อมูลออนไลน์สามารถเป็นแบบระบุชื่อหรือแบบไม่ระบุชื่อ ได้อย่างหนึ่ง สำหรับไฟล์ข้อมูลออนไลน์แบบไม่ระบุชื่อ อาจมีการระบุ QINLINE เป็นชื่อไฟล์ในคำสั่ง //DATA หรือไม่มีการระบุชื่ออย่างใดอย่างหนึ่ง สำหรับไฟล์ข้อมูลออนไลน์ที่ระบุชื่อ จะมีการระบุชื่อไฟล์

ตัว ไฟล์ข้อมูลออนไลน์แบบระบุชื่อ มีลักษณะดังต่อไปนี้:

- มีชื่อที่ไม่ซ้ำกันในงาน ไฟล์ข้อมูลออนไลน์ต้องมีชื่อที่ไม่ซ้ำกัน
- สามารถใช้ได้มากกว่าหนึ่งครั้งในงานหนึ่ง
- ในแต่ละครั้งที่เปิดไฟล์ ไฟล์จะถูกจัดตำแหน่งอยู่ที่เร็กคอร์ดแรก

ในการใช้ไฟล์ข้อมูลออนไลน์แบบระบุชื่อ คุณต้องระบุชื่อไฟล์ในโปรแกรม หรือใช้คำสั่งแทนที่เพื่อเปลี่ยนชื่อไฟล์ที่ระบุ ในโปรแกรมเป็นชื่อของไฟล์ข้อมูลออนไลน์ อย่างไรก็ตาม ไฟล์ต้องถูกเปิด สำหรับการอินพุตเท่านั้น

ตัว **ไฟล์ข้อมูลออนไลน์แบบไม่ระบุชื่อ** มีลักษณะดังต่อไปนี้:

- ชื่อของไฟล์คือ QINLINE (ในงานแบ็ตซ์ไฟล์ข้อมูลออนไลน์แบบไม่ระบุชื่อทั้งหมดจะ ถูกตั้งชื่อเหมือนกัน)
- สามารถใช้ได้ครั้งเดียวเท่านั้นในงานหนึ่ง
- เมื่อมีไฟล์ข้อมูลออนไลน์แบบไม่ระบุชื่ออยู่มากกว่าหนึ่งไฟล์ในงาน ไฟล์ต้องอยู่ใน input stream ในลำดับเดียวกับลำดับเมื่อไฟล์ถูกเปิด

ถ้าต้องการใช้ไฟล์ข้อมูลออนไลน์แบบไม่ระบุชื่อ ให้ทำอย่างใดอย่างหนึ่งดังนี้:

- ระบุ QINLINE ในโปรแกรม
- ใช้คำสั่งการแทนที่เพื่อเปลี่ยนชื่อที่ระบุในโปรแกรมเป็น QINLINE

ถ้าภาษาชั้นสูงของคุณต้องการชื่อไฟล์ที่ไม่ซ้ำกันในหนึ่งโปรแกรม, คุณสามารถใช้ QINLINE เป็นชื่อไฟล์ได้เพียงครั้งเดียวเท่านั้น ถ้าคุณต้องการใช้ไฟล์ข้อมูลออนไลน์แบบไม่ระบุชื่อมากกว่าหนึ่งไฟล์ คุณสามารถใช้คำสั่งการแทนที่ไฟล์ในโปรแกรมเพื่อระบุ QINLINE สำหรับไฟล์ข้อมูลออนไลน์แบบไม่ระบุชื่อเพิ่มเติม

หมายเหตุ: ถ้าคุณรันคำสั่งแบบมีเงื่อนไขและประมวลผลไฟล์ข้อมูลออนไลน์แบบไม่ระบุชื่อ มากกว่าหนึ่งไฟล์ จะไม่สามารถคาดการณ์ผลลัพธ์ได้ถ้าใช้ไฟล์ข้อมูลออนไลน์แบบไม่ ระบุชื่อที่ไม่ถูกต้อง

หลักการที่เกี่ยวข้อง

“ข้อควรพิจารณาในการเปิดไฟล์ข้อมูลออนไลน์”

คุณต้องพิจารณาองค์ประกอบเหล่านี้เมื่อคุณเปิดไฟล์ข้อมูลออนไลน์

ข้อควรพิจารณาในการเปิดไฟล์ข้อมูลออนไลน์:

คุณต้องพิจารณาองค์ประกอบเหล่านี้เมื่อคุณเปิดไฟล์ข้อมูลออนไลน์

- ความยาวเร็กคอร์ดระบุความยาวของอินพุตเร็กคอร์ด (ความยาวเร็กคอร์ดเป็นสิ่งที่เลือกได้) เมื่อความยาวเร็กคอร์ดเกินกว่าความยาวของข้อมูล จะมีการส่ง ข้อความไปที่โปรแกรมของคุณ ข้อมูลจะถูกเสริมด้วยพื้นที่เปล่า เมื่อความยาวเร็กคอร์ด น้อยกว่าความยาวของข้อมูล เร็กคอร์ดจะถูกตัดปลาย
- เมื่อมีการระบุไฟล์ในโปรแกรม ระบบจะค้นหาไฟล์ตามไฟล์ข้อมูลออนไลน์ที่ระบุชื่อ ก่อนที่จะค้นหาไฟล์ในไลบรารี ดังนั้น ถ้าไฟล์ข้อมูลออนไลน์ที่ระบุชื่อมีชื่อเหมือนกับไฟล์ที่ไม่ใช่ไฟล์ข้อมูลออนไลน์ ไฟล์ข้อมูลออนไลน์จะถูกใช้เสมอ แม้ว่าชื่อไฟล์ จะถูกต้องตามชื่อไลบรารี
- ไฟล์ข้อมูลออนไลน์ที่ระบุชื่อสามารถถูกแบ่งใช้ระหว่างโปรแกรมต่างๆ ในงานเดียวกันได้โดยการระบุ SHARE(*YES) บน คำสั่งการสร้างไฟล์หรือการแทนที่ไฟล์ ตัวอย่างเช่น ถ้าคำสั่งการแทนที่ไฟล์ระบุว่าไฟล์ที่มีชื่อว่า INPUT และ SHARE (*YES) อยู่ในงานแบ็ตซ์พร้อมกับไฟล์ข้อมูลออนไลน์ที่มีชื่อว่า INPUT โปรแกรมใดๆที่รันในงานที่ระบุชื่อไฟล์ INPUT จะแบ่งใช้ไฟล์ข้อมูลออนไลน์ที่มีชื่อเหมือนกัน ไฟล์ข้อมูลออนไลน์ที่ไม่ได้ระบุชื่อไม่ สามารถถูกแบ่งใช้ระหว่างโปรแกรมต่างๆ ในงานเดียวกัน

- เมื่อคุณใช้ไฟล์ข้อมูลออนไลน์ คุณควรตรวจสอบให้แน่ใจว่าได้ระบุชนิดของไฟล์ที่ต้องการบนคำสั่ง //DATA ตัวอย่างเช่น ถ้าไฟล์จะถูกใช้เป็น ไฟล์ต้นฉบับ ชนิดของไฟล์บนคำสั่ง //DATA ต้องเป็นต้นฉบับ
- ไฟล์ข้อมูลออนไลน์ต้องถูกเปิดสำหรับอินพุตเท่านั้น

งานที่เกี่ยวข้อง

“ไฟล์ข้อมูลออนไลน์” ในหน้า 82

ไฟล์ข้อมูลออนไลน์คือไฟล์ข้อมูลที่อยู่เป็นส่วนหนึ่งของงานแบ็ตช์ เมื่องานถูกอ่านโดยเครื่องอ่านหรือคำสั่งการส่งงาน คุณใช้ SBMDBJOB หรือ STRDBRDR เพื่อจัดคิว CL batch stream (stream ของคำสั่ง CL ที่จะรัน) CL batch stream นั้นสามารถรวมข้อมูลที่จะวางเข้าไฟล์ออนไลน์ (ไฟล์ชั่วคราว) เมื่องานสิ้นสุดลง ไฟล์ข้อมูลออนไลน์จะถูกลบออก

บันทึกการใช้งาน

บันทึกการใช้งานมีข้อมูลที่เกี่ยวข้องกับคำขอที่ป้อนสำหรับ งาน บันทึกการใช้งานมีสองรูปแบบ รูปแบบการคงค้างและรูปแบบสพูล

ในรูปแบบการคงค้าง บันทึกการใช้งานสำหรับงานที่เสร็จสมบูรณ์สามารถเปลี่ยน เมื่องานอื่น (ระบบย่อย, ผู้ควบคุมระบบ และอื่นๆ) โต้ตอบกับงานที่เสร็จสมบูรณ์ในรูปแบบสพูล บันทึกการใช้งาน คือ 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 เพื่อให้พิมพ์บันทึกได้อย่างไรก็ได้ การจัดทำบันทึกการใช้งานไม่จำเป็นต้องหมายถึงการพิมพ์บันทึกการใช้งานหรือการสร้างสพูลไฟล์ (ตัวอย่างเช่น สามารถใช้ Control Job Log QMHCTLJL API เพื่อระบุให้บันทึกการใช้งานถูกบันทึกเป็น outfile เมื่อจบงาน)

คุณสามารถลดจำนวนของบันทึกการใช้งานที่จัดทำและลด contention สำหรับรีซอร์ส (เช่นเอาต์พุตคิว) ได้ สิ่งนี้จะช่วยลดการใช้รีซอร์ส ที่เกิดจากการจัดทำบันทึกการใช้งาน

หลักการที่เกี่ยวข้อง

“การจัดการบันทึกการใช้งาน” ในหน้า 217

งานส่วนใหญ่ในระบบของคุณมีบันทึกการใช้งานที่เชื่อมโยง บันทึกการใช้งานบอกให้คุณ ทราบถึงสิ่งต่างๆ มากมาย เช่น งานจะเริ่มเมื่อไร, งานจะจบเมื่อไร, คำสั่งอะไรกำลังรันอยู่, หมายเหตุความล้มเหลวและข้อความแสดงความผิดพลาด ข้อมูลนี้ช่วยให้คุณเห็นภาพว่าวัฏจักรการรันงานเป็นอย่างไร

“การจัดการเซิร์ฟเวอร์บันทึกการใช้งาน” ในหน้า 217

ระบบย่อย QSYSWRK ควบคุมเซิร์ฟเวอร์บันทึกการใช้งาน อย่างไรก็ดี มีบางภารกิจที่คุณสามารถปฏิบัติเพื่อปรับหรือจัดการเซิร์ฟเวอร์บันทึกการใช้งาน

งานที่เกี่ยวข้อง

“การลบไฟล์เอาต์พุตบันทึกการใช้งาน” ในหน้า 226

บันทึกการใช้งานจะถูกลบออกจากระบบ เมื่องานเสร็จสมบูรณ์ตามปกติตามปกติ หรือเมื่อใช้คำสั่ง Remove Pending Job

Log (QWTRMVJL) API หรือ End Job (ENDJOB) อย่างใดอย่างหนึ่ง นอกจากนั้น ถ้ามีการระบุ "clear incomplete job logs" บน IPL งานทั้งหมดในบันทึกการใช้งานที่ค้างจะถูกลบออกจากระบบในระหว่างการ IPL สามารถดูไฟล์เอาต์พุต บันทึกการใช้งานที่เหลือได้ข้างใต้ **Basic Operations → Printer Output.**

“การควบคุมข้อมูลบันทึกการใช้งานแบ็ตซ์” ในหน้า 225

สำหรับแบ็ตซ์แอ็พพลิเคชัน คุณอาจต้องการเปลี่ยนจำนวนของข้อมูลที่บันทึก ระดับไฟล์บันทึก (LOG(40 *NOLIST)) ที่ระบุในรายละเอียดของงาน สำหรับระบบย่อยที่ได้มาจาก IBM QBATCH นับเป็นไฟล์บันทึกที่สมบูรณ์ ถ้างานจบลงอย่างผิดปกติ ถ้างานจบลงอย่างปกติ จะไม่มีการจัดทำบันทึกการใช้งาน

“การเปลี่ยนระดับไฟล์บันทึกของงาน” ในหน้า 225

ระดับไฟล์บันทึกของงานคือระดับตัวเลขที่กำหนดให้กับ ชนิดข้อความที่จะบันทึก คุณสามารถเปลี่ยนระดับไฟล์บันทึก ในรายละเอียดของงานได้โดยการใช้อินเตอร์เฟซแบบอักษร อย่างไรก็ตาม ถ้าคุณต้องการเปลี่ยนระดับไฟล์บันทึกของงานเฉพาะ ให้ใช้หน้าต่าง **Job Properties – Job Log** ใน System i Navigator.

ข้อมูลที่เกี่ยวข้อง

รายงานจากประสบการณ์: ข้อควรพิจารณาเกี่ยวกับประสิทธิภาพของสพูล

วิธีการสร้างบันทึกการใช้งาน

บันทึกการใช้งานจะ พร้อมให้ใช้งานได้เมื่อต้องการ และไม่มีการสร้างบันทึกการใช้งาน ที่ไม่ต้องการ

พารามิเตอร์ LOG มีองค์ประกอบอยู่สามอย่างดังนี้: ระดับข้อความ (หรือการบันทึก), ความรุนแรงของข้อความ และระดับของเนื้อความ แต่ละองค์ประกอบ เหล่านี้มีค่าเฉพาะซึ่งเมื่อรวมกันแล้วกลายเป็นตัวกำหนดจำนวน และชนิดของข้อมูลที่งานจะส่งไปที่บันทึกการใช้งาน

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

คุณสามารถควบคุมสิ่งที่จะสร้างบันทึกการใช้งานได้โดยใช้พารามิเตอร์ LOGOUTPUT เมื่องานเสร็จสมบูรณ์ หนึ่งในสาม actions ที่ส่งผลกระทบต่อวิธีการสร้างบันทึกการใช้งาน จะเกิดขึ้น ต่อไปนี้เป็นค่าของพารามิเตอร์ LOGOUTPUT:

- เซิร์ฟเวอร์บันทึกการใช้งานจัดทำบันทึกการใช้งาน: (*JOBLOGSVR)
- ตัวงานเองจัดทำบันทึกการใช้งาน: ถ้างานไม่สามารถสร้างบันทึกการใช้งานของตัวเองได้ 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 (QMHCTLJL) API ในงานเพื่อระบุว่า ข้อความในบันทึกการใช้งานจะถูกบันทึกไปที่ไฟล์ฐานข้อมูล

อะไรควบคุมพารามิเตอร์บันทึกการใช้งาน?

เมื่องานเริ่มต้น งานจะได้ค่า LOGOUTPUT ของงานจากรายละเอียดของงาน ถ้ารายละเอียดของงานระบุ *SYSVAL (ดีฟอลต์สำหรับ CRTJOB) งานจะใช้ค่าเอาต์พุตบันทึกการใช้งานที่ระบุอยู่ในค่ากำหนดของระบบ Job log output (QLOGOUTPUT) (ในขณะที่ค่าที่จัดส่งสำหรับค่ากำหนดของระบบ Job log output (QLOGOUTPUT) คือ *JOBEND แต่ค่าที่แนะนำคือ

*JOBLOGSVR) หลังจากที่ได้สร้างแอ็ดทริบิวต์งาน LOGOUTPUT แล้ว การเปลี่ยนแปลงใดๆที่รายละเอียดของงาน หรือ คำกำหนดของระบบจะไม่มีผลต่องานที่แอ็ดทริบ การเปลี่ยนแปลงที่คำกำหนดของระบบหรือรายละเอียดของงานมีผลต่อ งาน ที่ป้อนเข้าสู่ระบบหลังจากการเปลี่ยนแปลง

คุณสามารถใช้คำสั่ง Change Job (CHGJOB) หรือ API (QWTCCHGJB) ในการเปลี่ยนแอ็ดทริบิวต์งาน LOGOUTPUT หลังจากที่ได้แอ็ดทริบิวต์ได้ถูกเซตในงานแล้ว การเปลี่ยนแปลงที่งานจะมีผล ในทันที

ไม่ว่าคุณจะเลือกวิธีการใด อีพชั่นสำหรับการ จัดการบันทึกการใช้งานจะเหมือนกัน คุณสามารถเซตงานไม่ให้สร้างบันทึกการใช้งาน (*PND) ให้งานสร้างบันทึกการใช้งาน (*JOBEND) หรือให้เซิร์ฟเวอร์บันทึกการใช้งานสร้างบันทึกการใช้งาน (*JOBLOGSVR)

งานที่เกี่ยวข้อง

“การหยุดการจัดทำบันทึกการใช้งานเฉพาะ” ในหน้า 223

ถ้าคุณต้องการหยุดการจัดทำบันทึกการใช้งานเฉพาะขึ้นหนึ่ง ห้ามใช้คำสั่ง End Job Log Server (ENDLOGSVR) คำสั่ง ENDLOGSVR จะจบเซิร์ฟเวอร์บันทึกการใช้งานทั้งหมด ซึ่งส่งผลให้หยุดการจัดทำบันทึกการใช้งานทั้งหมด

“การป้องกันการจัดทำบันทึกการใช้งาน” ในหน้า 223

การป้องกันการจัดทำบันทึกการใช้งานเป็นสิ่งที่มีความประโยชน์ ถ้าคุณทราบอยู่แล้วว่า คุณจะไม่ต้องการบันทึกการใช้งานและ คุณต้องการสวอนรีซอร์สของระบบ เมื่อคุณระบุว่าคุณไม่ต้องการให้จัดทำบันทึกการใช้งาน บันทึกการใช้งานจะไม่ถูกจัดทำ และยังคงค้างอยู่จนกว่าจะถูกลบโดยคำสั่ง Remove Pending Job Log (QWTRMVJL) หรือคำสั่ง End Job (ENDJOB)

“การควบคุมข้อมูลในบันทึกการใช้งาน” ในหน้า 224

เมื่อจัดการกับปัญหา คุณอาจต้องการบันทึกจำนวนสูงสุดของ ข้อมูลสำหรับงานที่เกิดปัญหาขึ้นบ่อย อีกทางหนึ่งคือ คุณ อาจไม่ต้องการสร้างบันทึกการใช้งานสำหรับงานที่จบลงตามปกติ หรือ คุณอาจไม่ต้องการรวมข้อมูล

การคงค้างบันทึกการใช้งาน

สภาพการคงค้างบันทึกการใช้งานมีใช้กันเป็นเวลาหลายปีแล้ว เมื่อแอ็ดทริบิวต์บันทึกการใช้งานของงานเป็น *PND จะไม่มี การสร้างบันทึกการใช้งาน คุณสามารถควบคุมวิธีการ และสถานการณ์ที่มีการสร้างบันทึกการใช้งานสำหรับงานเฉพาะได้

คุณลักษณะพิเศษใหม่นี้มีความประโยชน์เมื่อคุณใช้ระบบในสภาพที่ถูก จำกัด เมื่อระบบอยู่ในสภาพที่ถูกจำกัด, ระบบย่อยจะจบลง และ งานนับพันอาจจบลงพร้อมกัน สิ่งนี้อาจทำให้เกิดภาวะที่ใหญ่หลวง บนเอาต์พุตรีซอร์ส โดยการป้องกันการสร้างบันทึก การใช้งานเหล่านี้ คุณสามารถลดผลกระทบบนรีซอร์สเหล่านี้ได้เป็นอย่างมาก

อีกตัวอย่างหนึ่งที่คุณสามารถใช้คุณลักษณะพิเศษใหม่นี้คือ ในระหว่างที่การสื่อสาร ล้มเหลว บางที่มีงานคล้ายกันจำนวนมากที่ สร้างข้อความแสดงความผิดพลาด บันทึกการใช้งานเหมือนกัน คุณสามารถเซตบันทึกการใช้งานไม่ให้สร้างสพูลไฟล์สำหรับ งานทั้งหมดได้ จากนั้น ถ้าการสื่อสารเกิดล้มเหลว คุณสามารถใช้คำสั่ง Work with Job Log (WRKJOBLOG) เพื่อกำหนด บันทึกที่จะพิมพ์ได้ คุณยังสามารถใช้จอภาพ Work with Job Logs (WRKJOBLOG) เพื่อจัดการ บันทึกการใช้งานได้ด้วย

งานอาจอยู่ในสภาพการคงค้างบันทึกการใช้งาน เนื่องจากการทำงานของคำสั่ง Power Down System (PWRDWNSYS) The System i Navigator ส่วนการติดต่อกับผู้ใช้แสดงสถานะ “Completed - Job log pending” สำหรับงานเหล่านี้ นี้คือเซตย่อย ของ สถานะอินเตอร์เฟซแบบอักษร *OUTQ

การใช้ประโยชน์จากการพัฒนาเหล่านี้ช่วยให้คุณลดจำนวน ของบันทึกการใช้งานที่สร้างขึ้น และจึงช่วยลด contention สำหรับรีซอร์สได้ สิ่งนี้ ส่งผลให้ประสิทธิภาพของระบบดีขึ้น

หลักการที่เกี่ยวข้อง

“เมื่องานจบลงพร้อมกัน” ในหน้า 51

ในบางครั้ง งานหลายงานอาจจบลงในเวลาเดียวกัน ตัวอย่างเช่น เมื่อเกิดข้อผิดพลาด เน็ตเวิร์ก และแอ็ททริบิวต์งานถูกเซ็ท เป็น *ENDJOB หรือ *ENDJOBNO LIST นอกเหนือจาก การจบของงานแล้ว อาจเกิดการแก้ไขอุปกรณ์ดังต่อไปนี้
งานที่เกี่ยวข้อง

“การล้างบันทึกใช้งานที่คงค้าง” ในหน้า 228

การล้างข้อมูล หรือการลบงานออกจากบันทึกการใช้งานที่คงค้างสามารถทำได้หลายวิธี คุณสามารถจบงานด้วยค่า 0 สำหรับพารามิเตอร์ Maximum log entries (LOGLMT) ถ้างานจบลงแล้ว คุณสามารถรัน Remove Pending Job Log (QWTRMVJL) API คุณยังสามารถใช้คำสั่ง Work with Job Logs (WRKJOBLOG)

“การจัดทำพริเตอร์เอาต์พุตจากบันทึกการใช้งานที่คงค้าง” ในหน้า 227

งานที่ไม่มีการตั้งค่า System i Navigator Job Properties – Job Log นั้น ฟิลด์ Produce a job log ที่เลือกจะไม่จัดทำบันทึกการใช้งาน แต่บันทึกการใช้งานจะอยู่ในบันทึกการใช้งานที่คงค้าง ถ้าต้องการจัดทำพริเตอร์เอาต์พุตจากบันทึกการใช้งานที่อยู่ในบันทึกการใช้งานที่คง ค้าง ให้ใช้อินเตอร์เฟซแบบอักษร

เซิร์ฟเวอร์บันทึกการใช้งาน

โดยปกติ เซิร์ฟเวอร์บันทึกการใช้งานจะพิมพ์บันทึกการใช้งานของงานไปที่ สฟูลไฟล์ คุณสามารถเรตบันทึกการใช้งานไปที่พริเตอร์หรือไปที่ outfile (ถ้ามี การระบุให้ทำโดยใช้คำสั่ง QMHCTLJL, Control job log API) อย่างไรก็ตาม ไม่แนะนำให้ใช้วิธีการนี้สำหรับการจัดทำบันทึกการใช้งาน

คุณสามารถดูข้อมูลเกี่ยวกับเซิร์ฟเวอร์บันทึกการใช้งานผ่านทาง System i Navigator จากจอแสดงผล Work Management → Server Jobs หรือจอแสดงผล Work Management → Active Jobs (เพื่อให้ง่ายขึ้นในการระบุงานที่รันอยู่บนเซิร์ฟเวอร์บันทึกการใช้งาน ให้ตรวจสอบ ให้แน่ใจว่าคุณได้รวมคอลัมน์ Server ในจอแสดงผลของคุณ)

จำนวนสูงสุดของเซิร์ฟเวอร์บันทึกการใช้งานที่สามารถแอ็คทีฟได้ในเวลาหนึ่งคือ 30 คุณเริ่มเซิร์ฟเวอร์บันทึกการใช้งานเพิ่มเติมและจัดการเซิร์ฟเวอร์ในวิธีการเดียวกันกับเซิร์ฟเวอร์อื่นในระบบของคุณ สิ่งนี้ทำได้โดยคำสั่งอินเตอร์เฟซแบบอักษร STRLOGSVR

วิธีการเริ่มเซิร์ฟเวอร์บันทึกการใช้งาน

โดยค่าดีฟอลต์ เซิร์ฟเวอร์ บันทึกการใช้งานจะเริ่มโดยอัตโนมัติเมื่อระบบย่อย QSYSWRK เริ่มต้น เซิร์ฟเวอร์ จะสิ้นสุดเมื่อระบบย่อย QSYSWRK สิ้นสุดลง

คำสั่ง Start Job Log Server (STRLOGSVR) เริ่มเซิร์ฟเวอร์บันทึกการใช้งาน เซิร์ฟเวอร์บันทึกการใช้งานจะบันทึกบันทึกการใช้งานสำหรับงานที่อยู่ในบันทึกการใช้งานที่มีสถานะคงค้าง และไม่มีแอ็ททริบิวต์ของ *PND เซิร์ฟเวอร์บันทึกการใช้งานบันทึก บันทึกการใช้งานของงานไปที่สฟูลไฟล์, ไปที่พริเตอร์ หรือไปที่ outfile อย่างใดอย่างหนึ่ง (ถ้ามีการระบุให้ทำโดยใช้ QMHCTLJL, Control job log API)

งานที่เกี่ยวข้อง

“Reconfiguring เซิร์ฟเวอร์บันทึกการใช้งาน” ในหน้า 218

เมื่อจัดส่งมา เซิร์ฟเวอร์บันทึกการใช้งานจะรันใน QSYSWRK QSYSWRK จะแอ็คทีฟอย่างต่อเนื่อง เพื่อเพิ่มประสิทธิภาพ คุณอาจต้อง reconfigure เซิร์ฟเวอร์ บันทึกการใช้งานของคุณเพื่อรันในระบบย่อยอื่น

“การเริ่มเซิร์ฟเวอร์บันทึกการใช้งาน” ในหน้า 219

โดยค่าดีฟอลต์, เซิร์ฟเวอร์ บันทึกการใช้งานจะเริ่มโดยอัตโนมัติเมื่อระบบย่อย QSYSWRK เริ่มต้น คุณสามารถเริ่มเซิร์ฟเวอร์บันทึกการใช้งาน โดยใช้คำสั่ง Start Job Log Server (STRLOGSVR)

“การจบเซิร์ฟเวอร์บันทึกการใช้งาน” ในหน้า 218

คำสั่ง End Job Log Server (ENDLOGSVR) ใช้ในการจบเซิร์ฟเวอร์บันทึกการใช้งานตัวเดียว หรือหลายตัว เซิร์ฟเวอร์บันทึกการใช้งานจะบันทึกบันทึกการใช้งานสำหรับงานที่อยู่ในสถานะคงค้างบันทึกการใช้งาน ถ้ามีงานเซิร์ฟเวอร์บันทึกการใช้งานที่แอคทีฟอยู่มากกว่าหนึ่งงานในขณะนี้ออกคำสั่งนี้ งานเซิร์ฟเวอร์บันทึกการใช้งานทั้งหมดจะถูกจบลง

ข้อมูลที่เกี่ยวข้อง

Control Job Log Output (QMHCTLJL) API

ลักษณะการแสดงของบันทึกการใช้งาน

System i Navigator จัดอินเตอร์เฟซที่ใช้งาน และอ่านได้ง่าย ซึ่งคุณสามารถดูบันทึกการใช้งาน และข้อความของบันทึกการใช้งานได้ คุณยังสามารถดูบันทึกการใช้งานโดยใช้อินเตอร์เฟซแบบอักษรได้ด้วย

คุณสามารถควบคุมคอลัมน์ที่จะปรากฏในรายการบันทึกการใช้งานได้โดยใช้ 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	

อินเตอร์เฟซแบบอักษร

เมื่อคุณใช้คำสั่ง Display Job Log (DSPJOBLOG) คุณจะเห็นจอแสดงผล Job Log จอแสดงผลนี้แสดง ชื่อโปรแกรมพร้อมกับสัญลักษณ์พิเศษ ดังต่อไปนี้:

>>	คำสั่งการรันหรือคำสั่งถัดไปที่จะ รัน ตัวอย่างเช่น ถ้าโปรแกรม CL หรือโปรแกรมภาษาชั้นสูงถูกเรียก การเรียก ไปที่โปรแกรมจะถูกแสดงขึ้น
>	คำสั่งประมวลผลเสร็จสมบูรณ์แล้ว
..	คำสั่งยังไม่ได้ประมวลผล
?	ตอบข้อความ สัญลักษณ์นี้ทำเครื่องหมายทั้งข้อความ ที่ต้องการคำตอบและข้อความที่ได้ตอบไปแล้ว

หัวข้อบันทึกการใช้งาน:

หัวข้อบันทึกการใช้งานตั้งอยู่ที่ส่วนบนสุดของแต่ละหน้าของบันทึกการใช้งาน ที่พิมพ์ หัวข้อเหล่านี้ระบุงานซึ่งบันทึกการใช้งานใช้และลักษณะ ของแต่ละ entry ข้อมูลต่อไปนี้ เป็นรายการของ entry ที่อาจใช้ได้ ในหัวข้อบันทึกการใช้งาน

- ชื่อที่ถูกต้องทั้งหมดของงาน (ชื่องาน, ชื่อผู้ใช้ และหมายเลข งาน)
- ชื่อของรายละเอียดของงานที่ใช้ในการเริ่มงาน
- วันที่และเวลาที่งานเริ่มต้น

- message identifier
- ชนิดข้อความ
- ความรุนแรงของข้อความ
- วันที่และเวลาที่ส่งแต่ละข้อความ
- ข้อความ ถ้าระดับการบันทึกที่ระบุจะรวม second-level text ที่จะปรากฏบนบรรทัดที่ตามมาข้างใต้ข้อความ
- โปรแกรมซึ่งข้อความหรือคำขอถูกส่งมา
- หมายเลขคำสั่งอินเตอร์เฟซเครื่องหรือออฟเซตที่โปรแกรม ซึ่งข้อความถูกส่งไป

หมายเหตุ: หมายเลขคำสั่งอินเตอร์เฟซเครื่อง ปรากฏขึ้นสำหรับข้อความ escape, การแจ้ง และข้อความวินิจฉัยเท่านั้น สำหรับข้อความชนิดอื่นทั้งหมด, หมายเลขคำสั่งอินเตอร์เฟซเครื่องจะถูกเซตเป็นศูนย์

- ถ้างานใช้ APPC หัวข้อจะมีบรรทัดที่แสดงหน่วยของ work identifier สำหรับ APPC

ข้อความ:

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

ข้อความแจ้งเตือน - ข้อความเหล่านี้จะถูกส่งไปยัง QSYSOPR เนื่องจากข้อความเหล่านี้ต้องการการดำเนินการในทันที ข้อความจะประกอบด้วยปัญหา, สาเหตุ และการดำเนินการกู้คืนที่จำเป็น ตัวอย่างเช่น เซิร์ฟเวอร์ไม่สามารถเริ่มทำงานได้ หรือเซิร์ฟเวอร์หยุดทำงานโดยไม่คาดคิด บางเซิร์ฟเวอร์ส่งข้อความเตือนไปที่ QSYSOPR ข้อความเหล่านี้มี Alert Option (ALROPT) ระบุไว้ในรายละเอียดข้อความ คุณสามารถใช้ตัวแจ้งเตือนเพื่อให้การจัดการศูนย์กลางของข้อความที่แจ้งเตือนได้

ข้อความในบันทึกการใช้งาน - ข้อความเหล่านี้มีลักษณะเป็นข้อความวินิจฉัย, หมายความว่าข้อความเหล่านี้ไม่ใช่ข้อความที่สำคัญมากแต่มีไว้เพื่อเตือนผู้ใช้ว่าต้องมีการดำเนินการบางอย่าง ข้อความเหล่านี้อาจถูกสร้างขึ้นโดยระบบและโดยผู้ใช้

ระดับการบันทึกข้อความ

ระดับการบันทึกข้อความกำหนดว่า ข้อความใดและข้อความชนิดใดที่ควรถูกบันทึกสำหรับงาน ตารางต่อไปนี้อธิบายว่าแต่ละระดับแสดงถึงอะไร

ระดับ	รายละเอียด
ระดับ 1	ข้อความทั้งหมดที่ส่งไปที่ message queue ภายนอกของงาน พร้อมกับค่าความรุนแรงมากกว่าหรือเท่ากับค่าความรุนแรงข้อความ (ใน System i Navigator ค่าความรุนแรงของข้อความ (0-99) พบได้ในหน้าต่าง Job Properties - Job Log นี้เป็นค่าที่คุณสามารถควบคุมได้)
ระดับ 2	ข้อความทั้งหมดที่มีคุณสมบัตติดระดับ 1 และข้อความคำขอใดๆ ซึ่งส่งผลให้เกิดข้อความระดับสูงมากกว่า หรือเท่ากับค่าความรุนแรงของข้อความ หมายเหตุ: ข้อความระดับสูงคือข้อความที่จะ ถูกส่งไปที่ program message queue ของโปรแกรมที่ได้รับข้อความ คำขอ (เช่น QCMD เป็นโปรแกรมการประมวลผลคำขอที่ได้มาจาก IBM ที่จะได้รับข้อความคำขอ)

ระดับ	รายละเอียด
ระดับ 3	ข้อความทั้งหมดที่มีคุณสมบัติตรงกับระดับ 1 หรือระดับ 2 และข้อความคำขอทั้งหมด นอกจากนั้น คำสั่งใดๆ จากโปรแกรม CL จะถูกรวมด้วย ถ้ามีการเลือก Log commands from CL programs box (หน้าต่าง Job Properties – Job Log) หมายเหตุ: คำสั่งบันทึกจากกล่องโปรแกรม CL เท่าเทียมกับแอตทริบิวต์การล็อกของโปรแกรม CL
ระดับ 4	ข้อความคำขอทั้งหมดและข้อความทั้งหมดที่มีค่าความรุนแรง มากกว่าหรือเท่ากับค่าความรุนแรงการล็อกข้อความ รวมถึงข้อความ trace นอกจากนั้น คำสั่งใดๆ จากโปรแกรม CL จะถูกรวมด้วย ถ้ามีการเลือก คำสั่งบันทึกจากกล่องโปรแกรม CL (หน้าต่าง Job Properties – Job log) หมายเหตุ: คำสั่งบันทึกสำหรับกล่องโปรแกรม CL เท่าเทียมกับแอตทริบิวต์การล็อกของโปรแกรม CL

งานที่เกี่ยวข้อง

“การเปลี่ยนระดับไฟล์บันทึกของงาน” ในหน้า 225

ระดับไฟล์บันทึกของงานคือระดับตัวเลขที่กำหนดให้กับ ชนิดข้อความที่จะบันทึก คุณสามารถเปลี่ยนระดับไฟล์บันทึกในรายละเอียดของงานได้โดยการใช้อินเตอร์เฟซแบบอักษร อย่างไรก็ตาม ถ้าคุณต้องการเปลี่ยนระดับไฟล์บันทึกของงานเฉพาะ ให้ใช้หน้าต่าง **Job Properties – Job Log** ใน System i Navigator.

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 โปรแกรมจะได้รับการควบคุมเมื่อได้รับข้อความในกรณีนี้ สิ่งสำคัญคือการตรวจสอบให้แน่ใจว่ามีการจัดทำบันทึกการใช้งาน เพื่อให้คุณสามารถกำหนดข้อผิดพลาดเฉพาะที่เกิดขึ้นได้

ตัวอย่างเช่น สมมุติว่า initial program แสดงเมนูที่มีอ็อปชัน signoff ซึ่งมีค่าดีฟอลต์เป็น *NOLIST initial program จะมอนิเตอร์ exceptions ทั้งหมด และรวมคำสั่ง Change Variable (CHGVAR) ที่จะเปลี่ยนอ็อปชัน signoff เป็น *LIST ถ้า exception เกิดขึ้น:

```
PGM
DCLF MENU
DCL &SIGNOFFDPT TYPE(*CHAR) LEN(7)
VALUE(*NOLIST)
.
.
.
MONMSG MSG(CPF000) EXEC(GOTO ERROR)
PROMPT: SNDRCVF RCFMT(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 จะเปลี่ยนอ็อปชันบนคำสั่ง SIGNOFF เป็น *LIST และเซตบนตัวบ่งชี้สามารถใช้ตัวบ่งชี้นี้เพื่อกำหนดเงื่อนไขค่าคงที่ที่จะแสดงข้อความที่อธิบายว่า ข้อผิดพลาดที่ไม่คาดคิดเกิดขึ้นได้อย่างไรและบอกให้ผู้ใช้จอภาพทราบว่าจะทำ อะไร

บันทึกประวัติ (ที่ผ่านมา) QHST

ไฟล์บันทึกประวัติ (QHST) ประกอบด้วย message queue และ ฟิสิคัลไฟล์ที่เรียกว่า ล็อกเวอร์ชัน. 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:

Qxxxxyyddd

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: จำนวน ของเร็คคอร์ดในล็อกเวอร์ชันของบันทึกประวัติถูกระบุในค่ากำหนดของระบบ Maximum records in history log (QHSTLOGSIZ). ค่ากำหนดของระบบนี้ ยังสนับสนุนอ็อปชัน *DAILY ซึ่งสร้างเวอร์ชันใหม่ในแต่ละวันด้วย.

การจัดรูปแบบบันทึกประวัติ (ที่ผ่านมา):

ไฟล์ฐานข้อมูลใช้ในการจัดเก็บข้อความที่ส่งไปที่ไฟล์บันทึกของระบบ. เนื่องจากเร็คคอร์ดทั้งหมดในฟิสิคัลไฟล์มีความยาวเท่ากันและข้อความที่ส่งไปที่ไฟล์บันทึกมีความยาวแตกต่างกัน, ข้อความจึงสามารถแตกออกมากกว่าหนึ่งเร็คคอร์ดได้.

แต่ละเร็คคอร์ดสำหรับข้อความมีฟิลด์อยู่สามฟิลด์:

- 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
ความยาวข้อความ	ไบนารี	2	111-112
ความยาวข้อมูลแบบข้อความ	ไบนารี	2	113-114
สำรองไว้	อักขระ	28	115-142

ตารางที่ 2. รูปแบบของฟิลด์ที่สาม (ข้อมูล) ของเร็กคอร์ดที่เหลือ

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

ข้อความจะไม่แยกออกเมื่อเริ่มเวอร์ชันใหม่ของไฟล์บันทึก. เร็กคอร์ดแรกและเร็กคอร์ดหลังสุดของข้อความจะอยู่ในเวอร์ชัน QHST เดียวกันเสมอ.

การประมวลผลไฟล์ QHST

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

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

ข้อมูลประสิทธิภาพและ QHST:

ข้อมูลประสิทธิภาพไม่ได้แสดงขึ้นเป็นข้อความบนข้อความ CPF1164 เนื่องจากข้อความอยู่ในไฟล์บันทึก QHST ผู้ใช้จึงสามารถเขียนแอฟพลิเคชันโปรแกรม เพื่อดึงข้อมูลนี้ออกมาได้

ข้อมูลประสิทธิภาพถูกส่งผ่านเป็นค่าข้อความการแทนที่ความยาวผันแปรได้ นี่หมายความว่าข้อมูลอยู่ในโครงสร้างภายใน entry แรก ที่เป็นความยาวของข้อมูล ขนาดของฟิลด์ความยาวไม่ได้รวมอยู่ใน ความยาว

เวลา และวันที่: ฟิลด์ข้อมูลแรกในโครงสร้างคือเวลาและวันที่ ที่งานเข้าสู่ระบบ และเวลาที่ขั้นตอนการเรอต์แรกสำหรับงาน ถูกเริ่มขึ้น เวลาแสดงอยู่ในรูปแบบ 'hh:mm:ss' ตัวแบ่งเวลา ในตัวอย่างนี้คือเครื่องหมายโคลอน ตัวแบ่งนี้ถูกกำหนดโดย ค่าที่ระบุในค่ากำหนดของระบบ Date and time (QTIMSEP) วันที่อยู่ใน รูปแบบที่กำหนดในค่ากำหนดของระบบ Date and time (QDATFMT) และ ตัวแบ่งอยู่ในค่ากำหนดของระบบ Date and time (QDATSEP) เวลาและวันที่ ที่งานเข้าสู่ระบบเป็นเวลา และวันที่ก่อนหน้าวันที่และเวลาการเริ่มงานในโครงสร้าง เวลาและวันที่ที่งานเข้าสู่ระบบคือ เวลาที่ระบบรับรู้ ว่างานจะถูกเริ่มต้น (โครงสร้างงานถูกเชื่อมต่ออยู่ข้างงาน) สำหรับ งานแบบโต้ตอบ เวลา job entry คือเวลาที่ระบบรับรู้ หัสมาน สำหรับงานแบ็ตช์ นี้คือเวลาที่คำสั่ง Batch Job (BCHJOB) หรือ Submit Job (SBMJOB) จะถูกประมวลผล สำหรับงานการมอนิเตอร์ เครื่องอ่าน หรือ writer นี้คือเวลาที่คำสั่งการเริ่มต้นที่สอดคล้องกันจะถูกประมวลผล และสำหรับงานแบบ autostart นี้คือเวลาในระหว่างการเริ่มต้นของระบบย่อย

เวลาตอบสนอง และจำนวน transaction ทั้งหมด: เวลาและวันที่ ต่อไปนี้คือเวลาตอบสนองทั้งหมดและจำนวนของ transaction เวลาตอบสนอง ทั้งหมดแสดงในหน่วยวินาทีและมีค่าสะสมของช่วงเวลาทั้งหมดของ งานที่กำลังประมวลผล นับตั้งแต่การกดปุ่ม Enter ที่เวิร์กสเตชัน จนถึงเวลาในการแสดงจอตแสดงผลถัดไป ข้อมูลนี้คล้ายกับข้อมูลที่แสดงบนจอแสดงผล Work with Active Job (WRKACTJOB) ฟิลด์นี้ใช้สำหรับงานแบบโต้ตอบเท่านั้น

ในกรณีที่ระบบล้มเหลวหรือมีการจบงานอย่างผิดปกติ อาจเป็นไปได้ที่ เวลาทั้งหมดจะไม่รวมเวลาของ transaction หลังสุด ในกรณี ได้ตัดการจบงาน จะเป็น 40 หรือมากกว่า นอกจากนี้ การนับ transaction มีความสำคัญสำหรับงานแบบ โต้ตอบมากกว่า งานคอนโซล และคือจำนวน ของช่วงเวลาตอบสนองที่นับโดยระบบในระหว่างงาน

จำนวนการดำเนินการ auxiliary I/O แบบซิงโครนัส: จำนวนของการดำเนินการ auxiliary I/O แบบซิงโครนัสตามจำนวน ของ transaction สำหรับงานที่มีหลาย thread ค่านี้จะรวมเฉพาะการดำเนินการ auxiliary I/O แบบซิงโครนัสจาก initial thread เท่านั้น ฟิลด์นี้เหมือนกับฟิลด์ AUXIO ที่ปรากฏบนจอแสดงผล WRKACTJOB ยกเว้นความแตกต่างดังต่อไปนี้:

- จอตแสดงผล WRKACTJOB แสดงค่าสำหรับ initial thread ของขั้นตอนการเรอต์ปัจจุบัน
- ข้อความ QHST มีผลรวมสะสมสำหรับ initial thread ของแต่ละขั้นตอนการเรอต์ในงาน

ถ้างานจบด้วยไค้ดการจบ 70 ค่านี้อาจไม่มี การนับสำหรับขั้นตอนการเรอต์สุดท้าย นอกจากนั้น ถ้างานมีอยู่ใน IPL (โดยใช้คำสั่ง Transfer Batch Job (TFRBCHJOB)) งานจะจบลงก่อนที่จะแ็คทีฟหลังจาก IPL ค่า คือ 0

สพูลไฟล์

สพูลไฟล์เก็บพักข้อมูลออกจนกว่าข้อมูลจะสามารถพิมพ์ได้ สพูลไฟล์ รวบรวมข้อมูลจากอุปกรณ์ จนกว่าโปรแกรมหรืออุปกรณ์จะสามารถประมวลผล ข้อมูลได้ โปรแกรมจะใช้สพูลไฟล์เมื่ออ่านหรือบันทึก ไปที่อุปกรณ์จริง นี่คือการสพูลอินพุตและเอาต์พุต

ระบบทำการสพูลอินพุตสำหรับฐานข้อมูลและดิสเก็ตไฟล์โปรแกรม ที่ได้มาจาก IBM เรียกว่าเครื่องอ่าน ถูกเริ่มในระบบย่อยการสพูล จะอ่าน streams งานแบ็ตซ์จากอุปกรณ์ และวางงานบนคิวงาน.

ทำการสพูลเอาต์พุตสำหรับพริ้นเตอร์โปรแกรมที่ได้มาจาก IBM เรียกว่า เครื่องเขียนพริ้นเตอร์ ถูกเริ่มในระบบย่อยการสพูล จะเลือกสพูลไฟล์ จากเอาต์พุตคิว, และบันทึกเร็กคอร์ดของเอาต์พุตไฟล์ที่สพูลไปที่พริ้นเตอร์

เมื่อสิ้นสุดงาน, สามารถบันทึกลงในบันทึกการใช้งานไปที่สพูลไฟล์ QPJOBLOG เพื่อให้บันทึกสามารถพิมพ์ได้

Job accounting

ฟังก์ชัน job accounting รวบรวมข้อมูลเพื่อให้คุณสามารถกำหนดว่าใครกำลังใช้ระบบของคุณและเขากำลังใช้รีซอร์สระบบอะไรอยู่ และยังช่วย คุณในการประเมินการใช้โดยรวมของระบบของคุณ Job accounting เป็น สิ่ง que เลือกได้ คุณต้องทำขั้นตอนพิเศษเพื่อตั้งค่า job accounting คุณสามารถร้องขอให้ ระบบรวบรวมข้อมูลการจัดทำบัญชีรีซอร์สของงาน, ข้อมูลการจัดทำบัญชี พริ้นเตอร์ไฟล์ หรือทั้งสองอย่าง คุณยังสามารถกำหนด accounting codes ที่โปรไฟล์ผู้ใช้ หรืองานเฉพาะได้

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

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

เมื่อคุณต้องการวิเคราะห์ข้อมูล job accounting ข้อมูลต้องถูกแยกจากเจอร์นัล QACGJRN โดยใช้คำสั่ง Display Journal (DSPJRN) ด้วยคำสั่งนี้ คุณสามารถบันทึก entry เข้าในไฟล์ฐานข้อมูลได้ คุณต้องบันทึกแอฟพลิเคชันโปรแกรมหรือใช้ยูทิลิตี้ เช่น เคียวรียูทิลิตี้ เพื่อวิเคราะห์ข้อมูล

หลักการที่เกี่ยวข้อง

“การจัดการ job accounting” ในหน้า 229

ฟังก์ชัน job accounting ไม่แเอคทีฟโดยค่าดีฟอลต์ ต้องใช้ขั้นตอนเริ่มต้น บางอย่างเพื่อตั้งค่าฟังก์ชันนี้ ข้อมูลต่อไปนี้ อธิบายวิธีการตั้งค่า job accounting และปฏิบัติการกิจทั่วไปส่วนใหญ่ที่เกี่ยวข้องกับ job accounting

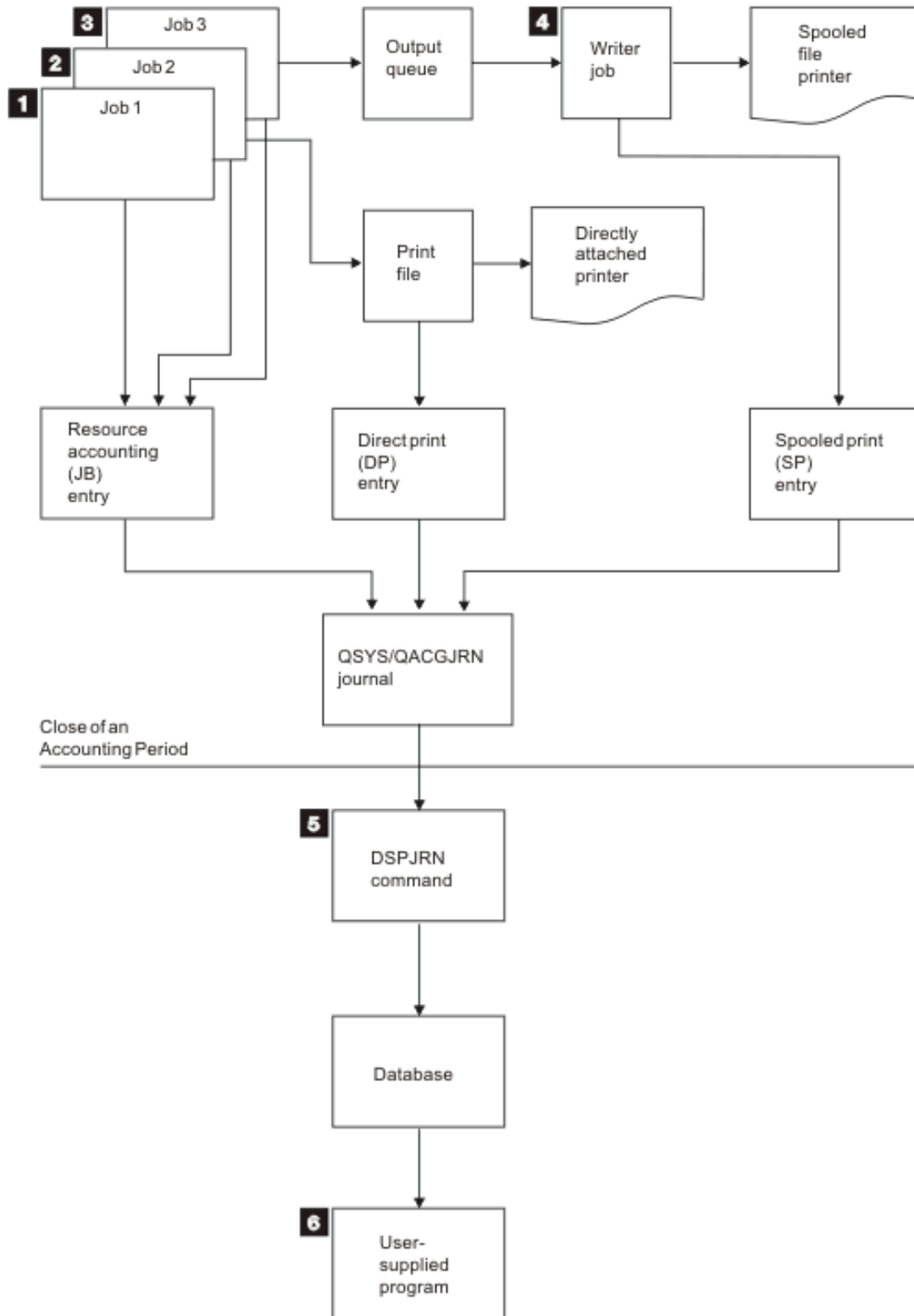
ข้อมูลที่เกี่ยวข้อง

การจัดการเจอร์นัล

การตั้งค่าการทำเจอร์นัล

วิธีการทำงานของ job accounting

สำหรับภาพรวมนี้ของวิธีการทำงานของ job accounting ให้สมมุติว่า มีงานที่เข้าสู่ระบบสามงาน



รูปที่ 4. ภาพรวมของ Job Accounting

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 ดังนั้น จึงไม่มีการสร้าง direct print (DP) หรือ spooled print (SP) journal entry สำหรับ Job1

2. Job2 คือการพิมพ์ไฟล์โดยตรงไปที่พริ้นเตอร์ เมื่อไฟล์เสร็จสมบูรณ์ จะมีการบันทึก DP journal entry ซึ่งสรุปข้อมูลที่ได้พิมพ์ 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 ระบบจะสรุปรีจิสเตอร์ที่ใช้ และบันทึก JB journal entry ถ้าบันทึกการใช้งานถูกสร้างขึ้นเมื่องานเสร็จสมบูรณ์ จะถือว่าบันทึกการใช้งานเป็นสพูลไฟล์ปกติ และมีการสร้าง SP journal entry ถ้าไฟล์ถูกพิมพ์
4. print writer จะถูกเริ่มขึ้นและพิมพ์ไฟล์ที่สร้างขึ้นโดยงานหนึ่งหรือ หลายงาน เมื่อ writer พิมพ์ไฟล์เสร็จแล้ว writer จะจัดทำ SP journal entry SP journal entry ไม่ถูกทำ ถ้าไฟล์ยกเลิกก่อนที่การพิมพ์จะเริ่มขึ้น
5. เมื่อสิ้นสุด accounting period สามารถใช้คำสั่ง Display Journal (DSPJRN) เพื่อบันทึก journal entry ที่สะสมไว้ในไฟล์ฐานข้อมูล
6. สามารถใช้โปรแกรมที่ผู้ใช้เขียนหรือเคียวรี่ยูทิลิตี้เพื่อวิเคราะห์ accounting data รายงาน เช่น รีจิสเตอร์ที่ใช้จะคอมไพล์ข้อมูลตาม accounting code เฉพาะ, ผู้ใช้ หรือชนิดของงาน

ลักษณะการดำเนินการ Job Accounting:

ระบบพยายามจัดสรรแหล่งเก็บข้อมูลหลักให้มีประสิทธิภาพมากที่สุดเท่าที่จะเป็นไปได้ งานอาจไม่ได้ใช้จำนวนของรีจิสเตอร์ที่เท่ากันในแต่ละครั้งที่รัน

ตัวอย่างเช่น ถ้ามีงานแอนด์ที่ฟหลายงานบนระบบของคุณ งานจะใช้เวลา ในการสร้างความต้องการรีจิสเตอร์ที่ต้องใช้สำหรับการรันขึ้นใหม่นานกว่าถ้าใช้สภาวะแวดล้อมระบบเฉพาะงาน ระบบจะใช้งานและระดับความสำคัญการรัน ที่ได้กำหนดที่งานต่างๆ เพื่อช่วยในการจัดการแหล่งเก็บข้อมูลหลัก ดังนั้น งานระดับความสำคัญสูงจึงสามารถใช้รีจิสเตอร์ของระบบน้อยกว่างานระดับความสำคัญต่ำได้

เนื่องจากลักษณะการดำเนินการของระบบเหล่านี้ คุณอาจต้องการใช้ การตีความของคุณเองหรือ algorithm กับข้อมูล job accounting ที่รวบรวมได้ ถ้าคุณคิดค่าธรรมเนียมสำหรับการใช้ระบบของคุณ คุณอาจต้องการคิดค่าธรรมเนียม มากขึ้นสำหรับงานที่มีระดับความสำคัญสูง, งานที่ทำในระหว่างเวลาระบบที่มีการใช้งานมากที่สุด หรือการใช้รีจิสเตอร์ที่สำคัญมาก

การประมวลผล Accounting Journal:

accounting journal QSYS/QACGJRN ถูกประมวลผลเช่นเดียวกับเจอร์นัลอื่น และยังสามารถบันทึกไฟล์ในเจอร์นัลนี้ได้ด้วย แม้ว่าจะแนะนำให้ผู้ใช้เจอร์นัล สำหรับข้อมูลการจัดทำบัญชีเพียงอย่างเดียวเพื่อให้การใช้งานง่าย

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

Job accounting entry

Job accounting entries ถูกวางใน journal receiver โดยเริ่มต้นด้วยงานถัดไปที่เข้าสู่ระบบหลังจากที่คำสั่ง Change System Value (CHGSYSVAL) มีผลบังคับใช้ accounting level ของงานจะถูกกำหนดเมื่องานเข้าสู่ระบบ ถ้าค่ากำหนดของระบบ Journal accounting information (QACGLVL) เปลี่ยนแปลง หลังจากทำงานเริ่มขึ้น ค่ากำหนดของระบบจะไม่มีผลกระทบต่อ ชนิดของการจัดทำบัญชีที่ปฏิบัติการสำหรับงานนั้น direct print (DP) และ spooled print (SP) entry จะเกิดขึ้น ถ้างานที่สร้างไฟล์กำลังดำเนินการภายใต้การจัดทำบัญชี และค่ากำหนดของระบบถูกเซตเป็น *PRINT ถ้าสพูลไฟล์ถูกพิมพ์หลังจากที่ได้เซต accounting level เป็น *PRINT หรือถ้างานที่สร้างไฟล์ เริ่มขึ้นก่อนที่ accounting level จะถูกเปลี่ยน จะไม่มีการทำเจอร์นัล สำหรับสพูลไฟล์เหล่านั้น

ใช้ job accounting เมื่อใด

วิธีเหล่านี้ช่วยคุณกำหนดว่าคุณควรใช้ job accounting และใช้ job accounting หรือไม่

ข้อมูลเพิ่มเติมที่ได้มาจาก job accounting

Job accounting มีข้อมูลทั้งหมดที่ได้มาจาก CPF1164 บวก:

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

ฟังก์ชัน job accounting จะมีประสิทธิภาพเพิ่มมากขึ้นสำหรับการรวบรวม สถิติ job accounting ถ้า:

- ข้อมูลรีจิสเตอร์ที่เกี่ยวกับการใช้ฐานข้อมูล, พรินเตอร์ 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, ซึ่งจะถูกส่งโดยอัตโนมัติไปที่ไฟล์บันทึก 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

คุณสามารถกำหนด job accounting code ได้เฉพาะ ถ้าคุณมีสิทธิ์ในการใช้คำสั่ง Create User Profile (CRTUSRPRF), Change User Profile (CHGUSRPRF) หรือ Change Accounting Code (CHGACGCDE) คำสั่งนี้จำกัด การใช้ accounting codes และนำเสนอพื้นฐานสำหรับการตรวจสอบความถูกต้องของการเปลี่ยนแปลงใดๆ

เฉพาะผู้ใช้ที่มีสิทธิ์พิเศษ *SECADM เท่านั้นที่สามารถใช้คำสั่ง CRTUSRPRF และ CHGUSRPRF ได้ อย่างไรก็ตาม เจ้าหน้าที่ดูแลความปลอดภัยสามารถมอบสิทธิ์นี้ได้โดยการสร้างโปรแกรม CL program ซึ่งอนุญาตให้ผู้ใช้อื่นใช้โปรแกรมของเจ้าหน้าที่ดูแลความปลอดภัย และเปลี่ยนพารามิเตอร์ ACGCDE ในโปรแกรม CL จากนั้น ผู้ใช้แต่ละรายอาจมีสิทธิ์ในการใช้หนึ่งหรือหลายโปรแกรม CL

พารามิเตอร์ ACGCDE ยังมีอยู่ในรายละเอียดของงานด้วย แต่คุณต้องมีสิทธิ์ในการใช้คำสั่ง CHGACGCDE เพื่อป้อนค่าอื่นที่ไม่ใช่ค่าดีฟอลต์ของ *USRPRF CHGACGCDE ถูกจัดส่งมาโดยมีสิทธิ์ PUBLIC เป็น *USE

สิทธิ์ในการใช้งานคำสั่ง CHGACGCDE

หากคุณอนุญาตให้ผู้ใช้ใช้คำสั่ง Change Accounting Code (CHGACGCDE) ผู้ใช้นั้นสามารถ:

- สร้าง หรือเปลี่ยนพารามิเตอร์ ACGCDE ในรายละเอียดของงาน (ต้องใช้สิทธิ์ในการสร้างหรือเปลี่ยนรายละเอียดของงานด้วย)
- เปลี่ยน accounting code ในงานปัจจุบันของตน
- เปลี่ยน accounting code ของงานของคนอื่น ถ้าผู้ใช้มีสิทธิ์พิเศษ *JOBCTL ด้วย

คุณสามารถนำเสนอการรักษาความปลอดภัยเพิ่มเติมได้โดยการใช้คำสั่ง CHGACGCDE ในโปรแกรม CL ซึ่งใช้สิทธิ์ของเจ้าของโปรแกรม สิทธิ์นี้อนุญาตให้ผู้ใช้ที่กำลังรันฟังก์ชันภายนอกสามารถใช้ฟังก์ชันที่ส่งผลกระทบต่อความปลอดภัยได้โดยไม่ต้องได้รับอนุญาตโดยตรงที่คำสั่ง CHGACGCDE

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

งานที่เกี่ยวข้อง

“การควบคุมการกำหนดโค้ดบัญชี” ในหน้า 230

ส่วนที่สำคัญของแอปพลิเคชันการประมวลผลข้อมูลใดๆ คือ การตรวจสอบให้มั่นใจว่ามีการระบุฟิลด์การควบคุมที่ถูกต้อง สำหรับ job accounting codes อาจต้องใช้ ฟังก์ชันการตรวจสอบความถูกต้องที่ซับซ้อน ซึ่งไม่เพียงแต่ตรวจสอบเฉพาะการมีอยู่จริงของโค้ดการพิสูจน์ตัวตนเท่านั้น แต่ยังตรวจสอบผู้ใช้ที่ได้รับอนุญาตให้ใช้โค้ดที่ระบุด้วย

เกี่ยวกับ accounting code

Initial accounting code (ยาวได้ถึง 15 อักขระ) ของงานที่กำหนดโดยค่าพารามิเตอร์ ACGCDE (accounting code) ในรายละเอียดของงาน และโปรไฟล์ผู้ใช้สำหรับงาน

เมื่องานเริ่มต้น รายละเอียดของงานจะถูกกำหนดที่งาน อีอบเจ็กต์รายละเอียดของงานมีค่าสำหรับพารามิเตอร์ ACGCDE ถ้าใช้ค่าดีฟอลต์ของ *USRPRF จะใช้ accounting code ในโปรไฟล์ผู้ใช้ของงาน

หมายเหตุ: เมื่องานถูกเริ่มโดยใช้คำสั่ง Submit Job (SBMJOB) accounting code ของงานจะเหมือนกับโค้ดของงานของผู้ส่ง คุณสามารถเปลี่ยน accounting code หลังจากที่ยานได้เข้าสู่ระบบแล้วได้โดยใช้คำสั่ง Change Accounting Code (CHGACGCDE)

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

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

งานที่เกี่ยวข้อง

“การตั้งค่า job accounting” ในหน้า 229

ถ้าต้องการตั้งค่า job accounting ให้ใช้อินเตอร์เฟซแบบอักขระ

“การควบคุมการกำหนดโค้ดบัญชี” ในหน้า 230

ส่วนที่สำคัญของแอปพลิเคชันการประมวลผลข้อมูลใดๆ คือ การตรวจสอบให้มั่นใจว่ามีการระบุฟิลด์การควบคุมที่ถูกต้อง สำหรับ job accounting codes อาจต้องใช้ ฟังก์ชันการตรวจสอบความถูกต้องที่ซับซ้อน ซึ่งไม่เพียงแต่ตรวจสอบเฉพาะการมีอยู่จริงของโค้ดการพิสูจน์ตัวตนเท่านั้น แต่ยังตรวจสอบผู้ใช้ที่ได้รับอนุญาต ให้ใช้โค้ดที่ระบุด้วย

Resource accounting

ข้อมูล Job resource accounting ถูกสรุปอยู่ใน job (JB) journal entry เมื่องานเสร็จสมบูรณ์ยิ่งไปกว่านั้น ระบบจะสร้าง JB journal entry ที่สรุปทรัพยากรที่ใช้ในแต่ละครั้งที่ใช้คำสั่ง Change Accounting Code (CHGACGCDE) JB journal entry รวมถึง:

- ชื่องานที่ถูกต้องทั้งหมด
- Accounting code สำหรับ accounting segment ที่เฟืองจบ
- เวลาของหน่วยการประมวลผล
- จำนวนของขั้นตอนการเรียด
- วันที่และเวลาที่งานเข้าสู่ระบบ
- วันที่และเวลาที่งานเริ่มต้น
- เวลา transaction ทั้งหมด (รวมเวลาการให้บริการ, เวลาที่ไม่ถูกต้อง, และเวลา แอ็คทีฟ)
- จำนวนของ transactions สำหรับงานแบบโต้ตอบทั้งหมด
- การดำเนินการ Auxiliary I/O
- ชนิดงาน
- โค้ดความสมบูรณ์ของงาน

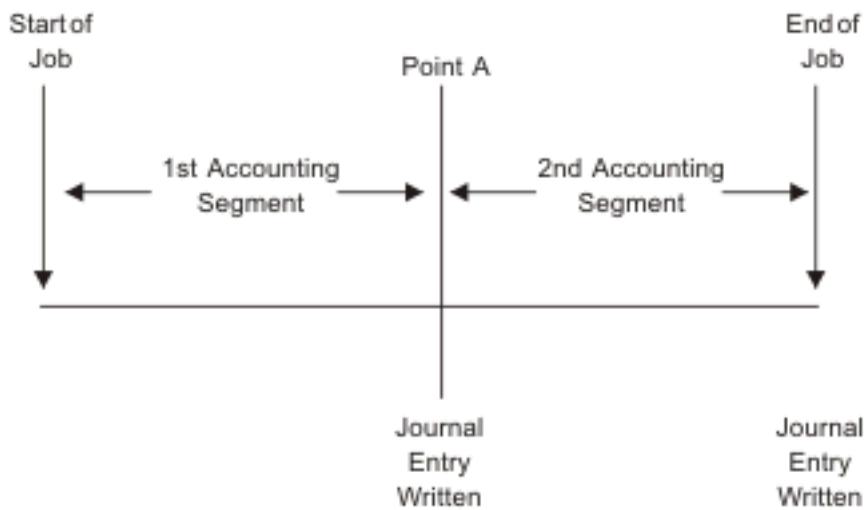
- จำนวนของบรรทัด, หน้า, และไฟล์ของพริ้นเตอร์ที่สร้าง ถ้าสพูลหรือพิมพ์โดยตรง
- จำนวนของการอ่าน, บันทึก, อัปเดต, และลบไฟล์ฐานข้อมูล
- จำนวนของการดำเนินการอ่านและบันทึกไฟล์ ICF

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

ข้อมูล Resource accounting

เมื่อวิเคราะห์ journal entry สิ่งสำคัญคือการทำความเข้าใจว่า journal entry จะถูกบันทึกอย่างไรและเมื่อไร JB journal entry จะถูกบันทึกที่ job accounting journal สำหรับงานในทุกครั้งที่ job accounting code ถูกเปลี่ยนแปลงและเมื่องานจบลง ดังนั้นงานหนึ่งอาจมีหลาย journal entry ได้

แต่ละ resource accounting journal entry มีข้อมูลเกี่ยวกับปริซอร์สที่ใช้ในขณะที่ accounting code ก่อนหน้านี้มีผลบังคับใช้อยู่ ให้พิจารณา ตัวอย่างต่อไปนี้:



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 เดียวจะสรุปปริซอร์สทั้งหมดที่งานใช้ ถ้ามีการเปลี่ยนแปลง job accounting code ในระหว่างที่มีงานอยู่ คุณต้องใส่เพิ่มฟิลต์ในหลาย JB entry เพื่อกำหนดปริซอร์สทั้งหมดที่งานใช้ การสร้างบันทึกการใช้งานไม่นับการใช้ หน่วยการประมวลผลสำหรับงาน หรือเอาต์พุตที่พิมพ์ของงานใน JB accounting entry อย่างไม่ดี ถ้าคุณกำลังใช้ print file accounting บันทึกการใช้งานที่พิมพ์ จะถูกรวมอยู่ใน printer file journal entry

งานแบบ prestart และ job accounting

ถ้าระบบของคุณใช้ job accounting โปรแกรมงานแบบ prestart ควรจะรันคำสั่ง Change Prestart Job (CHGPJ) ด้วยค่าค่าขอ การเริ่มโปรแกรมสำหรับพารามิเตอร์ accounting code parameter (CHGPJ ACGCDE(*PGMSTRRQS)) โดยทันที หลังจาก ที่ค่าขอการเริ่มโปรแกรมแนบกับงานแบบ prestart

action นี้เปลี่ยน accounting code เป็นค่าที่ระบุในโปรไฟล์ผู้ใช้ที่เชื่อมโยงกับค่าขอการเริ่มโปรแกรม ในทันทีก่อนที่โปรแกรม จะเสร็จสิ้นการจัดการค่าขอการเริ่มโปรแกรม โปรแกรมควรจะรันคำสั่ง Change Prestart Job (CHGPJ) โดยมี Prestart Job Entry เป็นค่าสำหรับพารามิเตอร์ accounting code (CHGPJ ACGCDE(*PJE)) สิ่งนี้จะเปลี่ยน accounting code กลับเป็นค่าที่ ระบุในรายละเอียดของงาน ของ prestart job entry

งานแบบ prestart สำหรับแอ็พพลิเคชันบีบีซี

งานแบบ prestart และงานเซิร์ฟเวอร์ที่ใช้งานแบบ prestart โดยปกติแล้วจะถูกตั้งค่าให้เริ่มด้วยโปรไฟล์ผู้ใช้ทั่วไป เช่น QUSER และจากนั้นรอให้จัดการค่าขอ เมื่องานแบบ prestart ถูกส่งค่าขอในการจัดการ งานจะเปลี่ยนโปรไฟล์ผู้ใช้โดยใช้ Set Profile Handle (QWTSETP) API ให้กับโปรไฟล์ของผู้ขอ, เซอร์วิสค่าขอ แล้วจึงเปลี่ยนกลับไปยังโปรไฟล์ผู้ใช้เบื้องต้น หากงานแบบ prestart ถูกตั้งค่าให้ใช้ซ้ำ (พารามิเตอร์ MAXUSE บนคำสั่ง Add Prestart Job Entry (ADDPJE) หรือ Change Prestart Job Entry (CHGPJE) มีค่ามากกว่า 1) งานจะรอการขออีกครั้งหนึ่ง และทำซ้ำข้างต้นอีกครั้ง ในกรณีนี้ งานแบบ prestart งานเดียว สามารถเซอร์วิสผู้ใช้ต่างๆจำนวนได้ ถ้าคุณต้องการให้สามารถเรียกเก็บผู้ใช้แต่ละรายเหล่านี้สำหรับริชอร์สที่ใช้จำนวน ต้องอัปเดต accounting code ก่อน และหลังการขอบริการแต่ละครั้ง งานเซิร์ฟเวอร์แบบ system-defined ดำเนินการนี้ให้คุณ แล้ว

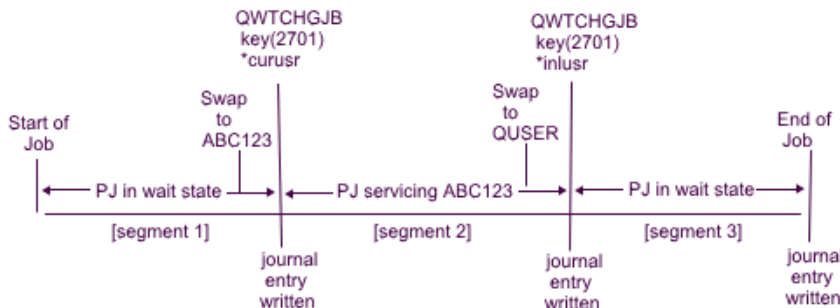


Figure 3. Prestart job with 3 accounting segments

ต่อไปนี้เป็นลักษณะของ journal entry ทั้งสาม ในรูปข้างต้น ถ้า SQL หรือ query ถูกใช้เพื่อจัดรูปแบบ:

ตารางที่ 3. งานแบบ prestart ที่มี accounting สามเซ็กเมนต์

Journal Entry #	Job Name	Job User	Job Number	User Profile	Accounting Code	CPU	Transaction
1	QSVREX1	QUSER	123456	ABC123	QUSER	50	1
2	QSVREX1	QUSER	123456	QUSER	ABC123	3729	120

ตารางที่ 3. งานแบบ prestart ที่มี accounting สามเช็คเมนต์ (ต่อ)

Journal Entry #	Job Name	Job User	Job Number	User Profile	Accounting Code	CPU	Transaction
3	QSVREX1	QUSER	123456	QUSER	QUSER	73	2

รีซอร์สที่ใช้ เช่น CPU และรายการอาจถูกชาร์จกลับไปที่ accounting code แต่ไม่จำเป็นต้องชาร์จผู้ใช้ที่มีรายชื่อในฟิลด์ User Profile (JAUSPF) โปรไฟล์ผู้ใช้เป็นโปรไฟล์ของผู้ใช้ปัจจุบันในขณะที่ journal entry ถูกบันทึก แต่ไม่จำเป็นต้องเป็นโปรไฟล์ผู้ใช้ที่แอดิชั่นระหว่างเช็คเมนต์ accounting ทั้งหมด ในตัวอย่างนี้ โปรไฟล์ผู้ใช้ถูกเปลี่ยนเพียงครั้งเดียวในแต่ละเช็คเมนต์ของสองเช็คเมนต์แรก เนื่องจาก journal entry ถูกบันทึกหลังการเปลี่ยน โปรไฟล์ผู้ใช้ปัจจุบันที่บันทึกใน entry ไม่ใช่ของผู้ใช้ที่รีซอร์สระหว่างเช็คเมนต์ accounting ก่อน

เช่นเดียวกัน Job User ไม่สามารถถูกชาร์จอย่างไว้วางใจสำหรับรีซอร์สที่ใช้ เพราะว่าเป็นผู้ใช้ที่เริ่มงาน และเป็นส่วนหนึ่งของชื่องานที่ถูกต้อง ซึ่งจะไม่เปลี่ยนแปลง แม้ในการเซอวิสผู้ใช้อื่น accounting code เป็นฟิลด์ที่วางใจได้เพียงฟิลด์เดียวที่สามารถใช้ในการชาร์จการใช้รีซอร์สได้ accounting code แตกต่างจากฟิลด์ผู้ใช้อื่น เพราะ accounting code ถูกบันทึกกับงานจนกว่างานจะเปลี่ยนแปลง เวลาเปลี่ยนแปลง accounting code ของงานปัจจุบันจะถูกบันทึกเข้า journal entry ก่อน และจากนั้นจะจัดเก็บ accounting code ใหม่ในงาน

หลักการที่เกี่ยวข้อง

“งานแบบ prestart” ในหน้า 51

งานแบบ prestart คืองานแบ็คชิ่งที่เริ่มต้นก่อนที่จะได้รับคำขอร้อง งานแบบ prestart ถูก เริ่มต้นก่อนงานชนิดอื่นๆ ในระบบย่อย งานแบบ prestart แตกต่างจากงานอื่นๆ เนื่องจาก การใช้ prestart job entry (ส่วนของ subsystem description) ในการกำหนด โปรแกรม, คลาส และพูลของหน่วยความจำที่จะใช้เมื่องานถูกเริ่มต้น

“การจัดการงานแบบ prestart” ในหน้า 137

คุณสามารถใช้งานแบบ prestart เพื่อลดจำนวนเวลาที่ต้องใช้ในการจัดการคำขอการเริ่มโปรแกรม ซึ่งเป็นภารกิจทั่วไปที่เกี่ยวข้องกับงานแบบ prestart ที่คุณสามารถดำเนินการได้

ข้อมูลที่เกี่ยวข้อง

รายงานจากประสบการณ์: การปรับ prestart job entries

รายงานจากประสบการณ์: การทำบัญชีผู้ใช้งาน

การประมวลผลงานระบบสำหรับ job accounting

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

หมายเหตุ: คุณไม่สามารถใช้คำสั่ง Change Accounting Code (CHGACGCDE) เพื่อเปลี่ยน accounting code ของการมอ นิเตอร์ระบบย่อย หรือเครื่องอ่าน หรือ writer อย่างไรก็ดี คุณสามารถเปลี่ยน accounting code ของเครื่องอ่าน หรือ writer ได้โดยการเปลี่ยนรายละเอียดของงานที่ได้มาจาก IBM และ โปรไฟล์ผู้ใช้ จากนั้นเริ่มเครื่องอ่านหรือ writer อีกครั้งหนึ่ง

การประมวลผลแบบแบ็คชิ่งและ job accounting

งานแบ็คชิ่งใดๆ ที่ถูกส่งโดยใช้คำสั่ง Submit Job (SBMJOB) จะใช้ accounting code เหมือนกันกับงานที่ส่งงานแบ็คชิ่งโดยอัตโนมัติ เมื่อใช้คำสั่ง SBJOB, accounting codes ไม่สามารถถูกแทนที่ด้วยค่าเดิม ไม่ว่า job description entry จะลง ใดต่ออย่างไรก็ตาม

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

- ก่อนและหลังจากการใช้คำสั่ง SBMJOB
- ทันทีโดยงานแบ็ตซ์.

งานแบ็ตซ์ที่ส่งโดยใช้เครื่องอ่านหรือคำสั่ง Submit Database Job (SBMDBJOB) จะใช้ accounting code ที่ระบุในรายละเอียดของงานสำหรับงานแบ็ตซ์ ถ้ารายละเอียดของงานระบุ ACGCDE(*USRPRF), accounting code จะถูกนำมาจากโปรไฟล์ผู้ใช้ที่ใช้สำหรับงาน.

การประมวลผลแบบโต้ตอบและ job accounting

ถ้างานแบบโต้ตอบมีอ็อปชันสำหรับผู้ใช้จำนวนคงที่ และ แต่ละอ็อปชันมี accounting code ที่กำหนดให้ คุณอาจต้องการกำหนดโค้ดใหม่โดยอัตโนมัติเมื่อผู้ใช้ร้องขอการทำงานบนฟังก์ชันใหม่

วิธีการปกติสำหรับเมนูอ็อปชันคือ การร้องขอพื้นที่ฟังก์ชันใหม่ จากนั้น จะใช้คำสั่ง Change Accounting Code (CHGACGCDE) ภายในโปรแกรม CL และค่างานที่ใช้สำหรับ accounting ก่อนหน้านี้จะถูกสรุปใน JB accounting journal entry

ถ้าผู้ใช้มีงานที่ได้รับมอบหมายหลายอย่าง ซึ่งเฉพาะผู้ใช้เท่านั้นที่ทราบ accounting codes ที่จะใช้ คุณสามารถ:

- ให้สิทธิในการป้อนคำสั่ง CHGACGCDE
- เขียนโปรแกรมเพื่อพร้อมต์ผู้ใช้สำหรับ accounting code

หมายเหตุ: สำหรับงาน source pass-through, ข้อมูล job accounting ไม่รวม งาน target pass-through job สำหรับงาน target pass-through, ข้อมูล job accounting ไม่รวมงานแบ็ตซ์การสื่อสารที่เชื่อมโยง

Printer file accounting

Journal entry สำหรับ printer file accounting มีอยู่สองชนิด; DP สำหรับ nonspooled printer files และ SP สำหรับ spooled printer files journal entry ทั้งสองชนิดแบ่งใช้รูปแบบ journal entry เดียวกัน แม้ว่าข้อมูลบางอย่างจะพร้อมใช้งานเฉพาะใน SP entry DP และ SP journal entry มีข้อมูลเช่น:

- ชื่องานที่ถูกต้องทั้งหมด
- Accounting code
- ชื่อไฟล์อุปกรณ์และไลบรารี
- ชื่ออุปกรณ์
- ชนิดของอุปกรณ์และรุ่น
- จำนวนทั้งหมดของหน้าและบรรทัดที่พิมพ์ ถ้ามีหลายก๊อปปี้ ข้อมูลเป็นผลรวมของก๊อปปี้ทั้งหมด
- Spooled file name (only in the SP entry)
- Spooled file number (only in the SP entry)
- Output priority (only in the SP entry)
- From type (only in the SP entry)
- Form type (only in the SP entry)

- จำนวนทั้งหมดของไบต์ของข้อมูลการควบคุมและข้อมูลการพิมพ์ที่ส่งไปที่ อุปกรณ์การพิมพ์ ถ้ามีหลายก็อปปี นี่เป็นผลรวมของก็อปปีทั้งหมด(นี้ใช้เฉพาะกับ SP entry เท่านั้น)

DP และ SP journal entry เกิดขึ้นเมื่อไฟล์ถูกพิมพ์ ถ้าไม่เคยพิมพ์สพูลไฟล์ จะไม่มี SP journal entry ปรากฏขึ้น

Journal entry สำหรับ job accounting

ระบบนำเสนอ journal entry 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 file ที่สร้างบน อุปกรณ์การพิมพ์ (nonspooled)
 - Spooled print (SP) journal entry: มีข้อมูลเกี่ยวกับ printer file ที่สร้างบนอุปกรณ์การพิมพ์ (spooled)

ข้อมูลฟิลด์ Job accounting journal entry:

ตารางเหล่านี้แสดงรายชื่อข้อมูลฟิลด์ที่อยู่ใน job journal entry สามารถดูข้อมูลเพิ่มเติมเกี่ยวกับฟิลด์ต่างๆได้ในไฟล์อ้างอิงของฟิลด์ QSYS/QAJBACG และ QSYS/QAJBACG4

ตารางที่ 4. Job journal entry fields

ชื่อฟิลด์ (อักขระ 14)	Description	ฟิลด์แอตทริบิวต์	ความคิดเห็น
JAJOB	ชื่องาน	อักขระ (10)	
JAUSER	ผู้ใช้งาน	อักขระ (10)	
JANBR	หมายเลขงาน	Zoned (6,0)	
JACDE	โค้ดบัญชี	อักขระ (15)	
JACPU	เวลาหน่วยการประมวลผลที่ใช้ (ในมิลลิวินาที)	Packed decimal (11,0)	เวลาหน่วยการประมวลผลไม่รวมการใช้หน่วยการประมวลผล และสถิติพรีนเตอร์สำหรับการสร้างบันทึกการใช้งาน.
JARTGS	จำนวนของขั้นตอนการเรอต์	Packed decimal (5,0)	
JAEDTE	งานที่เข้าสู่ระบบ - วันที่ Job entry (รูปแบบ mmdyy)	อักขระ (6)	
JAETIM	งานที่เข้าสู่ระบบ - เวลา Job entry (รูปแบบ hhmmss)	อักขระ (6)	

ตารางที่ 4. Job journal entry fileds (ต่อ)

ชื่อฟิลด์ (อักขระ 14)	Description	ฟิลด์แอตทริบิวต์	ความคิดเห็น
JASDTE	วันที่และเวลาการเริ่มงาน - วันที่เริ่มงาน (รูปแบบ mmddyy)	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
JASTIM	วันที่และเวลาการเริ่มงาน - เวลาเริ่มงาน (รูปแบบ hhmmss)	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
JATRNT	เวลา transaction ทั้งหมด (ในวินาที)	Packed decimal (11,0)	เวลา transaction ทั้งหมดถูกเซ็ทเป็น -1 เมื่อ: <ul style="list-style-type: none"> • เวลาถูกเซ็ทย้อนหลัง • เกิดโอเวอร์โฟลวในไฟล์เมื่อทำการคำนวณ • ระบบปิดในขณะที่งานแอ็คทีฟ
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: A งานแบบ autostart B งานแบ็คซ์ (รวมการสื่อสารและ MRT) I งานแบบโต้ตอบ M การมอนิเตอร์ระบบย่อย R Spooling reader W Spooling writer หมายเหตุ: ข้อมูลเหล่านี้เหมือนกับข้อมูลที่ใช้ในข้อความ CPF1164, ยกเว้นว่าข้อความ CPF1164 รวมข้อมูลงานระบบบางอย่างที่ไม่รวมอยู่ใน journal entries

ตารางที่ 4. Job journal entry files (ต่อ)

ชื่อไฟล์ (อักษร 14)	Description	ฟิลด์แอตทริบิวต์	ความคิดเห็น
JACCDE	โค้ดความสมบูรณ์	Packed decimal (3,0)	<p>โค้ดความสมบูรณ์, ซึ่งคล้ายกับโค้ดที่ใช้สำหรับข้อความ CPF1164 มีดังนี้:</p> <p>000 การเสร็จสมบูรณ์ตามปกติ</p> <p>010 การเสร็จสมบูรณ์ตามปกติในระหว่างการจบที่ควบคุมหรือการจบระบบย่อยที่ควบคุม</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 งานถูกบังคับให้เสร็จสมบูรณ์หลังจากขีดจำกัดเวลาลิ้นสุดลง</p> <p>099 Accounting entry ที่เกิดจากคำสั่ง CHGACGCDE</p>
JALINE	จำนวนของบรรทัดพิมพ์	Packed decimal (11,0)	จำนวนของบรรทัดพิมพ์ไม่สะท้อนถึงสิ่งที่จะถูกพิมพ์จริง เนื่องจากสพูลไฟล์สามารถถูกยกเลิกและพิมพ์ที่ละหลายก็อปปีได้ ข้อมูลใน JB journal entry สะท้อนให้เห็นเฉพาะสิ่งที่ถูกบันทึกโดย โปรแกรม และไม่รวมบรรทัดใดๆ ที่บันทึกสำหรับบันทึกการใช้งาน ดูคำอธิบายเกี่ยวกับ DP และ SP printer file accounting data ได้ในภายหลังในบทนี้
JAPAGE	จำนวนของหน้าที่พิมพ์	Packed decimal (11,0)	
JAPRTF	จำนวนของไฟล์พิมพ์	Packed decimal (11,0)	
JADBPT	จำนวนของการดำเนินการบันทึกฐานข้อมูล	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), จำนวนเหล่านี้จะแสดงแต่ละบล็อกของเร็กคอร์ดที่อ่าน ไม่ใช่จำนวนของแต่ละเร็กคอร์ดที่อ่าน
JADBGT	จำนวนของการดำเนินการอ่านฐานข้อมูล	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), จำนวนเหล่านี้จะแสดงแต่ละบล็อกของเร็กคอร์ดที่อ่าน ไม่ใช่จำนวนของแต่ละเร็กคอร์ดที่อ่าน

ตารางที่ 4. Job journal entry fields (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) จำนวนเหล่านี้จะแสดงแต่ละบล็อกของเร็กคอร์ดที่อ่าน ไม่ใช่จำนวนของแต่ละเร็กคอร์ดที่อ่าน
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 4. Job journal entry fields (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)	

Printer file accounting data สำหรับ direct print และ spooled print:

Accounting code ที่ใช้สำหรับ direct print (DP) หรือ spooled print (SP) journal entry เป็น accounting code ของงานในเวลาไฟล์ถูกปิด บางครั้ง DP หรือ SP entry ถูกสร้างขึ้นก่อนที่ไฟล์ถูกปิด (เช่น เมื่อ writer ซึ่งกำลังสร้างไฟล์ SCHEDULE (*IMMED) ลีนส์ดลง) เมื่อเกิดกรณีเช่นนี้ จะใช้ accounting code ปัจจุบันของงาน

DP หรือ SP journal entry ถูกสร้างขึ้นสำหรับแต่ละไฟล์ที่พิมพ์ ถ้าบันทึกการใช้งานถูกสพูลแล้วพิมพ์ SP entry จะถูกสร้างขึ้นสำหรับบันทึกการใช้งาน และ SP entry จะถูกบันทึกสำหรับดิสเก็ตสพูลไฟล์ที่ redirect ไปที่พริ้นเตอร์โดย print writer

ข้อมูล DP accounting journal:

ไฟล์ QSYS/QAPTACG5 ประกอบด้วยฟิลด์ที่จะถูกใช้ใน direct print (DP) journal entry ตารางนี้แสดงรายการฟิลด์เหล่านี้และแอตทริบิวต์ของฟิลด์

ตารางที่ 5. ฟิลด์ direct print journal entry

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 5. ฟิลด์ direct print journal entry (ต่อ)

Field Name	Description	Field Attributes
JATLIN	จำนวนทั้งหมดของบรรทัดพิมพ์ที่จัดทำ	Packed decimal (11,0)
JASPFN	พื้นที่เปล่าเสมอ	อักขระ (10)
JASPNB	พื้นที่เปล่าเสมอ	อักขระ (4)
JAOPY	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)
JAS PDT	พื้นที่เปล่าเสมอ	อักขระ (7)
JASPTM	พื้นที่เปล่าเสมอ	อักขระ (6)
JADFASP	พื้นที่เปล่าเสมอ	อักขระ (10)

ข้อมูล SP accounting journal:

ตารางนี้แสดงรายการของฟิลด์ (พบในไฟล์ QSYS/QAPTACG5) ที่ถูกใช้ใน spooled print (SP) journal entry

หมายเหตุ: ข้อมูล SP accounting journal คล้ายกับข้อมูลที่ให้ในข้อมูล direct print (DP) accounting journal ยกเว้นว่า ชื่อสพูลไฟล์, หมายเลขสพูลไฟล์, ระดับความสำคัญของเอาต์พุต, form type และจำนวนทั้งหมดของไบต์ของข้อมูลการควบคุม และข้อมูลการพิมพ์ที่ส่งไปยังพรินเตอร์ SP journal entry จะไม่ถูกบันทึกถ้าสพูลไฟล์ถูกลบออกก่อนที่ writer จะเริ่มบันทึกไฟล์ไปที่อุปกรณ์

ตารางที่ 6. ฟิลด์ Spooled print journal entry

ชื่อฟิลด์	Description	ฟิลด์แอ็ททริบิวต์
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)

ตารางที่ 6. ฟิลด์ Spooled print journal entry (ต่อ)

ชื่อฟิลด์	Description	ฟิลด์แอตทริบิวต์
JADEVN	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	จำนวนทั้งหมดของไบต์ที่ส่งไปที่พรินเตอร์	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
- หน้าและบรรทัดพิเศษที่เกิดขึ้นจากการจัดตำแหน่งจะไม่ถูกรวม ในการนับหน้า, บรรทัด และไบต์
- ถ้าสพูลไฟล์เข้าสู่สถานะ WTR (แต่ถูกเซตเป็น MSGW) หรือถ้าไฟล์ถูกลบออก ในขณะที่อยู่ในสถานะ MSGW SP journal entry จะปรากฏขึ้นใน DP accounting journal เพื่อบ่งชี้ว่ามี 0 หน้า และ 0 บรรทัดที่พิมพ์
- ในขณะที่ใช้พรินเตอร์ที่ตั้งค่า AFP(*YES) ถ้าคุณลบ หรือพักไฟล์ในทันที หลังจากที่ไฟล์มีหน้าที่พิมพ์ SP entry สำหรับไฟล์นั้นอาจบ่งชี้ 0 หน้า และ 0 บรรทัดที่พิมพ์แม้ว่าได้มีการพิมพ์บางหน้าไปแล้ว
- การนับหน้า, บรรทัด และไบต์สำหรับงานและตัวแบ่งไฟล์ถูกรวมอยู่ในการนับ สำหรับไฟล์ที่เชื่อมโยงด้วย
- เมื่อไฟล์ IPDS มีกราฟิกส์หรือบาร์โค้ด และถูกส่งไปที่พรินเตอร์ IPDS ที่ไม่สนับสนุนกราฟิกส์หรือบาร์โค้ด, การนับหน้า, บรรทัด และไบต์จะรวมกราฟิกส์และบาร์โค้ดที่ไม่ได้พิมพ์ด้วย
- ถ้า printer configuration เป็น AFP(*YES) ฟิลด์สำหรับจำนวนทั้งหมดของ บรรทัดพิมพ์ที่จัดทำคือศูนย์ ฟิลด์จำนวนทั้งหมดของหน้าที่จัดทำคือ แก่ไข

การจัดการงาน

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

หัวข้อในส่วนนี้ให้ข้อมูลเกี่ยวกับชนิดของภารกิจจัดการระบบงานรายงาน และภารกิจอื่นที่คุณอาจต้องทำบนระบบของคุณ ในหัวข้อย่อยแต่ละอันจะบอกถึงความสำคัญของภารกิจแต่ละอย่าง รวมถึงวิธีการทำงานให้ภารกิจสำเร็จอีกด้วย

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 ทั้งข้อความและเริ่มระบบย่อยอื่นเมื่อฟังก์ชันการกู้คืนของคุณ เสร็จสมบูรณ์

```
1.00 /* SPCRECOV - Autostart program to call special recovery program */
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('Recovery program in operation-do not +
7.00                      start subsystems until notified') +
8.00                      TOMSGQ(QSYSOPR)
9.00          CALL    RECOVERY
10.00         SNDPGMMSG MSG('Recovery complete-jobs may be started') +
11.00         TOMSGQ(QSYSOPR)
12.00         ENDDO /* Recover */
13.00         ENDPGM
```

ข้อมูลที่เกี่ยวข้อง

การเปลี่ยนโปรแกรมเริ่มต้น IPL

การมอนิเตอร์ system ระบบ

การมอนิเตอร์ activity ระบบเป็นภารกิจประจำวันอันหนึ่งที่มีความสำคัญของผู้ดูแล การมอนิเตอร์การไหลของงานผ่านระบบ เป็นข้อมูลขั้นหนึ่งที่สามารถได้รับการมอนิเตอร์อยู่เป็นประจำทุกวัน คุณสามารถบรรลุสิ่งนี้ได้หลายวิธี เช่น การใช้ System i Navigator and System i Navigator Management Central

ถ้าใช้แบบจำลองที่ครึ่งบนของจอแสดงผล Work with System Status (WRKSYSSTS) ในอินเตอร์เฟซแบบอักษร หน้าต่าง System Status นำเสนอวิธีที่ง่าย และรวดเร็วในการตรวจสอบสถานะของระบบ Management Central ช่วยให้คุณสามารถ มอนิเตอร์ฟังก์ชันได้ละเอียดมากขึ้นโดยการใช้การมอนิเตอร์ระบบ

คุณสามารถเข้าใช้งานหน้าต่าง System Status จากโพลเดอร์ System หรือโพลเดอร์ Work Management .

ถ้าต้องการเรียก System Status จากโพลเดอร์ System :

1. จาก System i Navigator ให้ขยาย My Connections.
2. คลิกขวาการเชื่อมต่อซึ่งคุณต้องการทำงานและคลิก System Status.

ถ้าต้องการเรียกสถานะระบบจากโพลเดอร์ Work Management:

1. จาก System i Navigator ให้ขยาย Work Management.
2. คลิกขวา Work Management และคลิก System Status.

สำหรับข้อมูลเพิ่มเติมเกี่ยวกับภารกิจต่างๆที่คุณสามารถทำให้เสร็จสมบูรณ์โดยใช้สถานะระบบ ให้ดูคำอธิบาย System i Navigator

การตรวจสอบการใช้พูลหน่วยความจำ

การตรวจสอบจำนวนของหน่วยความจำที่พูลหน่วยความจำของคุณใช้เป็นประจำเป็นสิ่งสำคัญ. การตรวจสอบดังกล่าว, จะช่วยให้คุณปรับแต่งพูลให้รันอย่างมีประสิทธิภาพ สูงสุด, ซึ่งสิ่งนี้, จะรักษาให้วงรอบของงานรันอย่างราบรื่น. ใน System i Navigator คุณสามารถมอนิเตอร์จำนวนของหน่วยความจำที่พูลของคุณกำลังใช้อยู่ได้โดยง่าย

หากต้องการตรวจสอบการใช้งานหน่วยความจำให้ทำตามขั้นตอนต่อไปนี้:

1. จาก System i Navigator, ให้ขยาย **My Connections** → *your system* → **Work Management** → **Memory Pools** → **Active Pools** หรือ **Shared Pools**
2. คลิกขวาที่พูลหน่วยความจำที่คุณต้องการทำงานด้วย (ตัวอย่างเช่น, Interactive) และคลิก **Properties**.
3. คลิกแท็บ **Configuration**. ฟิลด์ **Current**, ซึ่งตั้งอยู่ภายในกลุ่ม **Size**, แสดงจำนวนของหน่วยความจำที่พูล มีอยู่ในปัจจุบัน.

หมายเหตุ: คุณสามารถดูขนาดปัจจุบันของพูลหน่วยความจำ เมื่อคลิก **Active Pools** หรือ **Shared Pools**. **Current Size** (เป็นเมกะไบต์) เป็นคอลัมน์ดีฟอลต์ที่คุณจะเห็น เมื่อรายการของพูลของหน่วยความจำแสดงขึ้นในระนาบด้านขวาของ System i Navigator

การควบคุมระดับของ activity ระบบ

คุณสามารถควบคุมจำนวน activity บนระบบได้โดยการควบคุมจำนวนงานที่สามารถแอ็คทีฟได้พร้อมกันในระบบย่อย หรือโดยการควบคุมการใช้ หน่วยการประมวลผลของงานซึ่งถูกเริ่มต้นแล้ว

ตารางที่ 7. วิธีการควบคุมระดับของ activity ระบบ

ฉันสามารถควบคุมอะไรได้บ้าง?	ฉันสามารถใช้อะไรในการควบคุม?	วิธีการใช้อินเตอร์เฟซแบบอักษร	System i Navigator interface method
จำนวนของงานแอคทีฟ	Subsystem Description	คำสั่ง: CHGSBSD MAXJOBS ใช้ พารามิเตอร์นี้ในการระบุจำนวนงานที่สามารถแอคทีฟได้พร้อมกันใน ระบบย่อย สำหรับระบบย่อยที่แอคทีฟ ผลรวมของงานทั้งหมดที่แอคทีฟ ในเวลาเดียวกันซึ่งถูกเริ่มผ่านทาง work entries ในระบบย่อย ไม่สามารถเกินกว่าค่าพารามิเตอร์ MAXJOBS จำนวนนี้ไม่รวมงานแบบ autostart ซึ่งอาจทำให้จำนวนงานเกินกว่าขีดจำกัดชั่วคราวเมื่อเริ่มต้น ระบบย่อย	ใช้หน้าต่าง Run Command คลิกขวาที่ ระบบจุดปลาย → Run Command พิมพ์คำสั่ง CHGSBSD แล้วจึงคลิก Prompt.
	Job Queue Entry	คำสั่ง: CHGJOBQE MAXACT ใช้ พารามิเตอร์นี้เพื่อระบุจำนวนงานแบ็คคิงจากคิวงานที่สามารถแอคทีฟได้พร้อมกัน ในระบบย่อย MAXACT เป็น 1 สำหรับคิวงานจะทำงาน ถูกเลือกเป็นลำดับตามระดับความสำคัญของงานจากคิวงาน พารามิเตอร์ MAXPTYn ใช้ในการระบุจำนวนงานที่สามารถแอคทีฟสำหรับระดับความสำคัญของงาน ที่ระบุ	ใช้หน้าต่าง Run Command คลิกขวาที่ ระบบจุดปลาย → Run Command พิมพ์คำสั่ง CHGJOBQE แล้วจึงคลิก Prompt.
	Workstation Entry	คำสั่ง: CHGWSE MAXACT ใช้ พารามิเตอร์นี้ถ้ามีการระบุพารามิเตอร์ WRKSTNTYPE พารามิเตอร์นี้ระบุจำนวนงานแบบโต้ตอบที่สามารถแอคทีฟได้พร้อมกันในระบบย่อย สำหรับ entry นั้น	ใช้หน้าต่าง Run Command คลิกขวาที่ ระบบจุดปลาย → Run Command พิมพ์คำสั่ง CHGWSE แล้วจึงคลิก Prompt.
	Communications Entry	คำสั่ง: CHGCMNE MAXACT ใช้ พารามิเตอร์นี้เพื่อระบุจำนวนของงานแบ็คคิงการสื่อสารที่สามารถแอคทีฟได้พร้อมกันสำหรับ entry นั้น	ใช้หน้าต่าง Run Command คลิกขวาที่ ระบบจุดปลาย → Run Command พิมพ์คำสั่ง CHGCMNE แล้วจึงคลิก Prompt.
	Routing Entry	คำสั่ง: CHGRTGE MAXACT ใช้ คำสั่งนี้เพื่อระบุจำนวนงานที่สามารถแอคทีฟได้พร้อมกันโดยใช้ routing entry ที่ให้	ใช้หน้าต่าง Run Command คลิกขวาที่ ระบบจุดปลาย → Run Command พิมพ์คำสั่ง CHGRTGE แล้วจึงคลิก Prompt.
	Prestart job entry	คำสั่ง: CHGPJE MAXJOBS ใช้ คำสั่งนี้เพื่อระบุจำนวนของงานแบบ prestart ที่สามารถแอคทีฟได้พร้อมกัน สำหรับ entry นั้น	ใช้หน้าต่าง Run Command คลิกขวาที่ ระบบจุดปลาย → Run Command พิมพ์คำสั่ง CHGPJE แล้วจึงคลิก Prompt.

ตารางที่ 7. วิธีการควบคุมระดับของ activity ระบบ (ต่อ)

ฉันสามารถควบคุมอะไรได้บ้าง?	ฉันสามารถใช้อะไรในการควบคุม?	วิธีการใช้อินเตอร์เฟซแบบอักษร	System i Navigator interface method
จำนวนของงานแอคทีฟ (ต่อเนื่อง)	ระบบ	ค่ากำหนดของระบบ Maximum eligible threads (QMAXACTLVL) ใช้เพื่อระบุจำนวนของ threads ที่สามารถแชร์แหล่งเก็บข้อมูลหลักและโพสเซเซอร์รี่ซอร์สได้ในเวลาเดียวกันงานแอคทีฟทั้งหมด (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: CHGSBSD POOLS ใช้ คำสั่งนี้เพื่อระบุ activity level สำหรับพูลแหล่งเก็บข้อมูลหลักที่ผู้ใช้กำหนด	ใช้หน้าต่าง Run Command คลิกขวาที่ ระบบจุดปลาย → Run Command พิมพ์คำสั่ง CHGSBSD แล้วจึงคลิก Prompt.

ตัวอย่าง: ความสัมพันธ์ของการควบคุม activity:

ตัวอย่างเหล่านี้แสดงความสัมพันธ์ของการควบคุม activity บางอย่าง สมมุติว่า 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 ใ้ทำงาน ทั้งหมดจะถูกเริ่มขึ้น

ตัวอย่างงานสิ่งานในระบบย่อย

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

ตัวอย่างระบบย่อยแบบเบ็ดเสร็จ MAXACT(1)

ระบบย่อย SBSE คือระบบย่อยแบบเบ็ดเสร็จซึ่งมีการระบุ 1 สำหรับ MAXACT แม้ว่า job queue entry ไม่ได้ระบุ MAXACT ซิด จำกัดคือหนึ่งงาน เนื่องจากมีการระบุ 1 สำหรับ MAXACT สำหรับระบบย่อย ดังนั้นงานในคิวงานจะถูกประมวลผลตามระดับความสำคัญที่ละหนึ่งงาน

การกำหนดสถานะของงาน

การมอนิเตอร์งานของคุณจะช่วยให้คุณเข้าใจว่างานดังกล่าวกำลังทำอะไรอยู่ สถานะของงานเป็นข้อมูลชิ้นสำคัญที่คุณสามารถใช้ดูว่างานชิ้นหนึ่งๆ กำลังทำงานอะไรอยู่

ถ้าต้องการตรวจสอบสถานะของงานแอคทีฟหรืองานเซิร์ฟเวอร์ ให้ปฏิบัติตามข้อแนะนำดังต่อไปนี้:

1. จาก System i Navigator ให้ขยาย **My Connections** → *your system* → **Work Management** → **Active Jobs** or **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 ในระบบย่อยโดยการเซตค่าจำนวนงานแอคทีฟสูงสุด เมื่อจำนวนงานบนระบบของคุณเพิ่มขึ้น คุณอาจต้องการปรับเปลี่ยนค่าจำนวนงานแอคทีฟสูงสุดในระบบย่อยของคุณ จำนวนที่คุณป้อนที่นี่ควรถูกเซต เพื่อให้ใช้ประโยชน์รีซอร์สพร้อมใช้งานได้สูงสุด การเพิ่มจำนวนของงานแอคทีฟโดยไม่ได้เพิ่มรีซอร์สพร้อมใช้งาน อาจทำให้ประสิทธิภาพการทำงานของระบบของคุณลดลง

ในการตรวจสอบจำนวนงานแอคทีฟสูงสุดในระบบย่อยของคุณ คุณสามารถใช้ System i Navigator หรืออินเตอร์เฟซแบบอักษร

System i Navigator:

1. จาก System i Navigator ให้ขยาย **My Connections** → *your system* → **Work Management** → **Subsystems** → **Active Subsystems**.
2. คลิกขวาระบบย่อยที่คุณต้องการมอนิเตอร์
3. เลือก **Properties**.

หมายเหตุ: ให้แน่ใจว่า คุณเซตอ็อปชันนี้อย่างระมัดระวังมาก ถ้าคุณเซตค่างานแอคทีฟสูงสุดสูงเกินไป, อาจทำให้ระบบของคุณทำงานช้า อย่างไรก็ตาม ถ้าคุณเซตงานแอคทีฟสูงสุดของคุณ ต่ำเกินไปงานของคุณอาจติดขัดและมีประสิทธิภาพ ต่ำ

อินเตอร์เฟซแบบอักษร:

คำสั่ง: Display Subsystem Description (DSPSBSD)

เลือก อีพพชัน 1: แอ็ททริบิวต์การดำเนินการ, เพื่อดูจำนวนงานสูงสุดใน ระบบย่อย

การกำหนดจำนวนของระบบย่อยที่ใช้พูลหน่วยความจำ

ระบบย่อยจะได้รับการจัดสรรหน่วยความจำจำนวนหนึ่งเพื่อใช้รันงาน เป็นสิ่งสำคัญที่ต้องทราบจำนวนของระบบย่อยที่กำลังดึงหน่วยความจำออกมาจาก พูลหน่วยความจำเดียวกัน หลังจากที่คุณทราบจำนวนของระบบย่อยที่กำลังส่งงาน ไปที่พูลและจำนวนของงานที่กำลังรันในพูล คุณอาจต้องการลด contention ริชอร์สโดยการปรับเปลี่ยนขนาดและ activity level ของพูล

System i Navigator:

ในการใช้ System i Navigator เพื่อมอนิเตอร์จำนวนของระบบย่อยที่กำลังใช้พูลหน่วยความจำ ให้ปฏิบัติตามข้อแนะนำต่อไปนี้:

1. จาก System i Navigator ให้ขยาย My Connections → Connection → Work Management → Memory Pools → Active Pools or Shared Pools.
2. คลิกขวาพูลหน่วยความจำที่คุณต้องการทำงานด้วยและคลิก Subsystems.
จากหน้าต่างนี้ คุณสามารถกำหนดจำนวนของระบบย่อยที่กำลังใช้แต่ละหน่วยความจำเพื่อรันงานของตน

อินเตอร์เฟซแบบอักษร:

คำสั่ง: Work with Subsystems (WRKSBS)

คำสั่งนี้แสดงรายการของระบบย่อยทั้งหมดและพูลของระบบย่อย

การดูสถิติประสิทธิภาพของงาน

ประสิทธิภาพของงานเป็นสิ่งสำคัญสำหรับทุกคนที่ใช้ผลิตภัณฑ์ System i Navigator เนื่องจากการรันงานหนึ่งที่ไม่ดีสามารถส่งผลกระทบต่องานอื่นบนระบบ การดูงานที่อาจเกิดปัญหาจะช่วยให้คุณป้องกันปัญหาประสิทธิภาพการทำงานก่อนที่จะเกิดขึ้นได้

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

ถ้าต้องการแสดงสถิติประสิทธิภาพที่ผ่านมา ให้ปฏิบัติตามข้อแนะนำต่อไปนี้:

1. จาก System i Navigator ให้ขยาย My Connections → *your system* → 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 สิ่งนี้จะช่วยให้คุณมอนิเตอร์งานที่อาจมีปัญหาหลายงานได้พร้อมกัน หน้าต่างแต่ละบานแสดงข้อมูลของงานเพียงหนึ่งงานเท่านั้น

สถิติประสิทธิภาพที่ผ่านมามีวิธีการหนึ่งในการดูประสิทธิภาพของงานเมื่องานนั้นเคลื่อนผ่านระบบ อีกวิธีหนึ่งในการดูงานบนระบบคือดูผ่านทางโพลเดอร์ Management Central คุณสามารถมอนิเตอร์งานใน Management Central รวมถึงการมอนิเตอร์ ประสิทธิภาพและข้อความของระบบ

การดูสถานะระบบโดยรวม

System i Navigator ใส่ข้อมูลทั้งหมดที่เกี่ยวข้องกับสถานะระบบไว้ในที่ที่เดียว การทำเช่นนี้ช่วยให้คุณสามารถ มอนิเตอร์วิธีการปฏิบัติการของระบบของคุณ ระบุพื้นที่ที่อาจเกิดปัญหา ได้ง่ายขึ้น และสามารถกำหนด action ที่คุณต้องทำเพื่อปรับปรุงประสิทธิภาพ ได้อย่างรวดเร็ว

หน้าต่าง System Status แบ่งสถานะระบบโดยรวมออกเป็นหกพื้นที่ ดังนี้:

ทั่วไป นี่คือเปอร์เซ็นต์การใช้งาน CPU ที่ผ่านไป, จำนวนของงานแอ็คทีฟ, เปอร์เซ็นต์ แอดเดรสที่ใช้, เปอร์เซ็นต์การใช้งานดิสก์พูลระบบ, งานทั้งหมดบนระบบ, เปอร์เซ็นต์ของแอดเดรสถาวรและแอดเดรสชั่วคราวที่ใช้, พื้นที่ว่างดิสก์ทั้งหมด และความจุของดิสก์พูลระบบ

งาน นี่คือจำนวนทั้งหมดของงาน, จำนวนของงานแอ็คทีฟ, จำนวนสูงสุดของงาน และจำนวนของ thread ที่แอ็คทีฟ

โพรเซสเซอร์

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

หน่วยความจำ

นี่คือหน่วยความจำทั้งหมด (แหล่งเก็บข้อมูลหลัก) บนระบบของคุณ และปุ่ม ที่ช่วยให้คุณเข้าถึงรายการของพูลหน่วยความจำแอ็คทีฟบนระบบ

พื้นที่ดิสก์

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

แอดเดรส

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

ในการดูสถานะระบบทั่วไป ให้ปฏิบัติตามข้อแนะนำต่อไปนี้:

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

หน้าต่าง System Status จะปรากฏขึ้น สำหรับข้อมูลเพิ่มเติมเกี่ยวกับหน้าต่างนี้ ให้ดู System i Navigator คำอธิบายออนไลน์

การตรวจสอบสถานะของดิสก์:

คุณสามารถตรวจสอบประสิทธิภาพของดิสก์ยูนิทบนระบบของคุณ หรือดูข้อมูลสถานะของดิสก์ยูนิทได้ตลอดเวลา

ในการดูหน้าต่าง Disk Status window ให้ปฏิบัติตามขั้นตอนต่อไปนี้:

1. จาก System i Navigator, ให้ขยาย **My Connections**.
2. คลิกขวาที่ **your system** และเลือกแท็บ **Disk Space** → **System Status**

3. บนหน้าต่าง Disk Space ให้คลิก **Disk Status**. หน้าต่าง Disk Status จะปรากฏขึ้น

คุณสามารถใช้ออปชัน **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 การจัดการงานเป็นสิ่งที่มากกว่า การพักผ่อนไว้และการย้ายงานจากคิวงานหนึ่งไปอีก คิวงานหนึ่ง หัวข้อนี้อธิบายภารกิจการจัดการงานที่ใช้ทั่วไปและ ภารกิจที่เกี่ยวข้องเพิ่มเติม ซึ่งสามารถช่วยพัฒนาประสิทธิภาพของระบบของคุณได้

ภารกิจงานทั่วไป

ภารกิจงานทั่วไปเป็นภารกิจทั่วไปที่สุดที่คุณสามารถดำเนินการกับงานได้ ช้อแนะนำใช้ได้สำหรับทั้ง System i Navigator (ในที่ที่พร้อมใช้งาน) และอินเตอร์เฟซแบบอักษร

การเริ่มงาน:

งานแบบโต้ตอบจะเริ่มเมื่อผู้ใช้ signs on ที่เวิร์กสเตชัน คุณเริ่มงานแบบ prestart และงานแบ็ตช์ โดยใช้ System i Navigator หรืออินเตอร์เฟซแบบอักษร ขึ้นอยู่กับสถานการณ์

การเริ่มงานแบ็ตช์ที่กำลังรออยู่ในคิวงาน:

ในบางโอกาส คุณอาจต้องการบังคับให้งานเริ่มในทันที ในขณะที่การย้ายงานไปที่คิวงานที่ไม่ยุ่งเป็นวิธีการที่มีประสิทธิภาพสูงสุด แต่มีวิธีการอื่นที่คุณสามารถใช้ได้

เพื่อเริ่มงานแบ็ตช์ในอันดับแรก ให้ตรวจสอบสถานะของคิวงานที่งานตั้งอยู่ และพิจารณาว่าการย้ายงานไปยังคิวอื่นเป็นวิธีที่เหมาะสมที่สุดสำหรับสถานการณ์ ของคุณหรือไม่ (**My Connections** → **server** → **Work Management** → **Job Queues** → **Active Job Queues** หรือ **All Job Queues**)

ถ้าการย้ายงานไปที่คิวงานอื่นเป็นสิ่งที่ไม่เหมาะสม, คุณสามารถพักงานที่รันอยู่ แล้วเปลี่ยนระดับความสำคัญของงานที่คุณต้องการเริ่ม อย่างไรก็ตาม ระวังเมื่อใช้วิธีการนี้ เนื่องจากงานที่ถูกพัก ยังคงรวมอยู่ในการนับงานแอคทีฟสูงสุด

ในการเปลี่ยนระดับความสำคัญของงาน และระยะเวลาที่งานควรรัน ให้ใช้ข้อแนะนำต่อไปนี้:

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 ให้ใช้ไคต์ delay loop แล้วลองใหม่ถ้าคำสั่ง STRPJ ล้มเหลว

จำนวนของ งานแบบ prestart ที่สามารถแอคทีฟได้ในเวลาเดียวกันถูกจำกัดโดยแอตทริบิวต์ MAXJOBS บน prestart job entry และโดยแอตทริบิวต์ MAXJOBS สำหรับระบบย่อย แอตทริบิวต์ MAXACT บน communications entry ควบคุม จำนวนของคำขอการเริ่มโปรแกรมที่สามารถให้บริการได้ผ่านทาง communications entry ในเวลาเดียวกัน

หมายเหตุ: ถ้าคุณระบุ *NO บนแอตทริบิวต์ 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 ต้องแอคทีฟในขณะที่ออก คำสั่งนี้ จำนวนของงานที่เริ่มต้นคือจำนวนที่ระบุในค่า INLJOBS ของ prestart job entry PJPGM ระบบย่อย จะเริ่มโปรแกรม PJPGM ในไลบรารี PJLIB

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

การจบงาน:

คุณสามารถใช้ System i Navigator หรืออินเตอร์เฟซแบบอักษรเพื่อจบงาน งานสามารถแอคทีฟหรืออยู่บนคิวงาน คุณสามารถจบงาน โดยทันทีหรือโดยการระบุช่วงเวลาเพื่อให้จบการประมวลผลงาน

System i Navigator:

ในการใช้ System i Navigator, ใช้ข้อแนะนำดังต่อไปนี้:

1. จาก System i 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) คำสั่งนี้จะจบงานทั้งหมดในระบบย่อย
- End System (ENDSYS) คำสั่งนี้จะจบ activity ส่วนใหญ่บนระบบ และปล่อยให้ระบบอยู่ในสภาพซึ่งมีเฉพาะคอนโซลเท่านั้นที่แอ็คทีฟในระบบย่อยควบคุม
- Power Down System (PWRDWN SYS) คำสั่งนี้จะเตรียมระบบสำหรับการสิ้นสุด แล้วเริ่มลำดับการปิดเครื่อง

งานสามารถจบลงโดยทันทีหรือจบในลักษณะที่ควบคุม อย่างไรก็ตามหนึ่ง ขอ แนะนำอย่างยิ่งให้คุณพยายามจบงานในลักษณะที่ควบคุมเสมอ

การจบงาน: แบบควบคุม:

การจบงานในลักษณะที่ควบคุมช่วยให้โปรแกรมที่กำลังรันอยู่ในงานสามารถปฏิบัติการล้าง end-of-job ได้ สามารถระบุเวลา หนึ่ง เพื่ออนุญาตให้งานจบลงในลักษณะที่ควบคุม ถ้าเวลาหนึ่งหมดลงก่อนที่งานจะจบ งานจะถูกจบในทันที

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

การดึงข้อมูล End Status แบบซิงโครนัส

ที่บางจุด แอฟพลิเคชันสามารถตรวจสอบ End Status แบบซิงโครนัสของงานซึ่งแอฟพลิเคชันกำลังรันอยู่ได้ คุณสามารถเรียกข้อมูลสถานะการจบของงานโดยใช้คำสั่ง Retrieve Job Attributes (RTVJOBA) CL นอกจากนี้ คุณ สามารถใช้หนึ่งจากหลาย APIs ที่เรียกข้อมูลสถานะการจบของงาน คุณสามารถหาข้อมูลเพิ่มเติมเกี่ยวกับ APIs เหล่านี้ในรายงานจากประสบการณ์ *แอ็คทีวิตี งานการจัดการระบบงาน*

ตรวจสอบคำสั่งคืนหลักและร่องอย่างซิงโครนัส หลังจากการดำเนินการ I/O

สำหรับทั้ง display I/O และ ICF communications I/O คำสั่งคืนหลักของ 02, หรือคำสั่งคืนหลักของ 03 ที่มีคำสั่งคืน ร่อง 09 บ่งชี้ว่างานกำลังจบในลักษณะที่ควบคุม

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 ที่เหมาะสมเพื่อให้แอ็พพลิเคชันสามารถจบในลักษณะแบบควบคุม

งานที่เกี่ยวข้อง

“การหยุดระบบย่อย” ในหน้า 178

คุณสามารถใช้ System i Navigator หรืออินเทอร์เฟซแบบอักษรเพื่อหยุดหนึ่ง หรือหลายระบบย่อย และระบุสิ่งที่จะเกิดขึ้นกับงานแอ็พพลิเคชันที่กำลังจะประมวลผล ไม่มีงานใหม่ หรือขั้นตอนการเรอัติใหม่ถูกเริ่มในระบบย่อยหลังจากที่ระบบย่อยถูกหยุด

ข้อมูลที่เกี่ยวข้อง

ค่ากำหนดของระบบงาน: เวลาสูงสุดสำหรับการจบในทันที

การจบงาน:ทันที:

เมื่องานจบโดยทันที คุณอาจได้ผลลัพธ์ที่ไม่ต้องการ เช่น ข้อมูลแอ็พพลิเคชันที่ถูกอัปเดตเป็นบางส่วนใช้อ็อพชันจบแบบทันที เฉพาะถ้าการจบแบบควบคุมไม่ประสบความสำเร็จ เท่านั้น

ก่อนการจบงาน คุณควรตรวจสอบว่าไม่มี logical unit of work อยู่ในสถานะที่น่าสงสัย เนื่องจากการดำเนินการ two-phase commit ที่อยู่ในระหว่างดำเนินการ ถ้ามี ค่าของ Action ifENDJOB commitment option อาจส่งผลกระทบต่ออย่างมาก ต่อการประมวลผล ENDJOB อ็อพชันนี้เป็นส่วนหนึ่งของ Change Commitment Options (QTNCHGCO) API ตัวอย่างเช่น ถ้า Action ifENDJOB commitment option เป็นค่าดีฟอลต์ของ WAIT งานนี้จะถูกพักไว้ และจะทำให้การจบการประมวลผลงานไม่สมบูรณ์ จนกว่าการดำเนินการ commitment control จะเสร็จสมบูรณ์ สิ่งนี้ช่วยให้มั่นใจถึง integrity ฐานข้อมูลบนระบบที่เกี่ยวข้องทั้งหมด

เมื่อคุณใช้อ็อพชันจบแบบทันที ระบบจะประมวลผล minimal end-of-job ซึ่งสามารถรวมถึง:

- การปิดไฟล์ฐานข้อมูล
- การสพูลบันทึกการใช้งานที่เอาต์พุตคิว
- การล้างอ็อบเจกต์ภายในในระบบปฏิบัติการ
- การแสดงจอตแสดงผล end-of-job (สำหรับงานแบบโต้ตอบ)
- การทำกระบวนการ commitment control ให้เสร็จสมบูรณ์

ข้อมูลที่เกี่ยวข้อง

Change Commitment Options (QTNCHGCO) API

การหางาน:

สิ่งสำคัญ คือ ต้องทำความเข้าใจวิธีหางานบนระบบของคุณ ไม่ว่าด้วยเหตุผลใดก็ตาม บางเวลาคุณอาจต้องการข้อมูลจากงานเฉพาะใดๆ

ใน System i Navigator คุณสามารถใช้ Find บนงานทั้งหมดของคุณ หรือคุณสามารถทำการค้นหาให้แคบลงโดยการใช้ฟังก์ชัน Include ตามด้วย Find ฟังก์ชัน Include อนุญาตให้คุณใส่ข้อจำกัดของสิ่งที่จะแสดงใน System i Navigator. ตัวอย่างเช่น แทนที่จะใช้ Find เพื่อค้นหาหลายร้อยงาน คุณสามารถรัน Include เพื่อให้แสดงเฉพาะ งานบางชนิดได้ หรือคุณสามารถ แสดงเฉพาะงานที่มี job user ID ที่ระบุเท่านั้น

เพื่อประสิทธิภาพการทำงานสูงสุด ถ้าคุณ มีงานอยู่เป็นจำนวนมากบนระบบ ขอแนะนำให้คุณใช้ฟังก์ชัน Include เพื่อลดจำนวนของงานที่จะค้นหา ถ้าคุณมีงานอยู่เป็นจำนวนมากบนระบบ การค้นหาจาก งานทั้งหมดอาจทำให้ประสิทธิภาพการทำงานลดลง

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

System i Navigator:

ในการค้นหางานโดยใช้ตัวเลือก Find (Ctrl+F) ให้ใช้คำแนะนำดังต่อไปนี้:

1. จาก System i Navigator ให้ขยาย My Connections → *your system* → Work Management → Active Jobs.
2. บนเมนู Edit ให้คลิก Find (Ctrl+F).
3. ในฟิลด์ Search for: ให้พิมพ์ job ID ที่คุณต้องการค้นหา (เช่น Qqqtemp1) งานของคุณจะถูกค้นหาในคอลัมน์งาน ทั้งหมด
4. คลิก Find.. System i Navigator ไฮไลต์งานของคุณในทันทีที่พบ

เตือนความจำ: จะคำนึงถึงขนาดตัวพิมพ์ของชื่องาน เฉพาะเมื่ออยู่ในเครื่องหมายคำพูดเท่านั้น (ตัวอย่างเช่น MyJob) แต่หากชื่องานไม่ได้อยู่ในเครื่องหมายคำพูด คำสั่งจะไม่คำนึงขนาดตัวพิมพ์เล็กใหญ่

การจำกัดข้อมูลที่จะแสดง:

ถ้าต้องการจำกัดข้อมูลที่จะแสดง ให้ใช้ฟังก์ชัน Include

1. จาก System i Navigator ให้ขยาย My Connections → *your system* → Work Management → Active Jobs or Server Jobs.
2. จากเมนู View ให้คลิก Customize this View → Include. หน้าต่าง Include จะแสดงขึ้น
3. ในหน้าต่าง Include ให้เลือกตัวเลือกที่คุณต้องการในการค้นหางานของคุณ
4. คลิก OK.

อินเตอร์เฟซแบบอักษร:

ในการค้นหาบนระบบ ให้ใช้คำสั่ง Work with Active Job (WRKACTJOB), Work with User Job (WRKUSRJOB) หรือ Work with Submitted Job (WRKSBMJOB) อย่างไม่อย่างหนึ่ง

การดูงานบนคิวงาน:

คิวงานกรองงานบางส่วนที่จะถูกประมวลผลในการจัดการระบบงาน (ยกตัวอย่างเช่น งานแบบแบ็คซ์บางงาน) การดูงานในคิวงานได้ทำให้คุณสามารถดูว่างานใดที่กำลังรอที่จะถูกส่งไปยังระบบย่อย

System i Navigator:

ในการดูงานบนคิวงาน ให้ปฏิบัติตามคำแนะนำต่อไปนี้:

1. จาก System i Navigator ให้ขยาย My Connections → *your system* → Work Management → Job Queues → Active Job Queues or All Job Queues.
2. คลิกคิวงานซึ่งคุณต้องการแสดงงาน (เช่น Jobqueue1) งานภายในคิวงานจะปรากฏขึ้น

อินเตอร์เฟซแบบอักษร:

คำสั่ง: Work with Job Queue (WRKJOBQ)

คำสั่งนี้จะ แสดงรายการของคิวงานทั้งหมดที่พร้อมใช้งานบนระบบ หลังจากที่คุณระบุตำแหน่งคิวงานที่มีงานของคุณแล้ว คุณสามารถเลือกอีก 5=ทำงานกับ และแสดงงานทั้งหมดใน คิวงาน

คุณยังสามารถใช้คำสั่ง Work with Subsystems Job เพื่อแสดง รายการคิวงานและงานของคิวงานได้ด้วย

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

การดูงานบนระบบย่อย:

ระบบย่อยจะประสานการไหลของงานกับรีจิสเตอร์ที่ใช้ในการรัน System i Navigator ช่วยให้คุณสามารถดูงานที่แฉีกที่พอย์ในปัจจุบัน (แต่ไม่จำเป็นต้องกำลังรัน) ในระบบย่อย

System i Navigator:

ถ้าต้องการดูงานในระบบย่อย ให้ทำตามขั้นตอนเหล่านี้:

1. จาก System i Navigator ให้ขยาย My Connections → *your system* → Work Management → Subsystems → Active Subsystems.
2. คลิกระบบย่อยที่มีงานซึ่งคุณต้องการแสดง

อินเตอร์เฟซแบบอักษร:

คำสั่ง: Work with Active Jobs (WRKACTJOB SBS (subsystem name))

คำสั่ง: Work with Subsystem Descriptions (WRKSBSD)

ใช้คำสั่ง Work with Subsystem Descriptions เพื่อแสดงรายการของ ระบบย่อย หลังจากที่คุณพบระบบย่อยที่มีงานของคุณ ให้ใช้ออปชัน 8=ทำงาน กับงานระบบย่อย เพื่อแสดงข้อมูลงาน

หมายเหตุ: ระบบย่อย ต้องแฉีกที่ฟเพื่อแสดงข้อมูลงาน

การดูแอ็ททริบิวต์งาน:

แอ็ททริบิวต์งานมีข้อมูลเกี่ยวกับวิธีการประมวลผลงาน แอ็ททริบิวต์งานจะถูกระบุไว้ตั้งแต่ตอนที่งานได้ถูกสร้างขึ้น บางแอ็ททริบิวต์ มาจากรายละเอียดของงาน หลังจากทำงานถูกสร้างขึ้น สามารถดู และจัดการแอ็ททริบิวต์งานผ่านทางจัดการระบบงานใน System i Navigator. หน้าคุณสมบัติงานใน System i Navigator ทำให้งานของผู้ควบคุมระบบงานง่ายขึ้น โดยการนำเสนอฟังก์ชันที่มีประสิทธิภาพ และใช้งานได้ง่ายสำหรับการจัดการงาน

ข้อมูลที่เกี่ยวข้อง

รายงานจากประสบการณ์: แอ็ททริบิวต์งานการจัดการระบบงาน

System i Navigator:

ในการดูแอ็ททริบิวต์งาน ให้ใช้ข้อแนะนำดังต่อไปนี้:

1. จาก System i Navigator ให้ขยาย My Connections → server → Work Management → Active Jobs or Server Jobs ขึ้นอยู่กับชนิดของงานที่คุณต้องการทำงานด้วย
2. ค้นหงานที่มีคุณสมบัติที่คุณต้องการดูหรือเปลี่ยนแปลง

3. คลิกขวาที่ Job Name และคลิก Properties.

แอ็ททริบิวต์งานสามารถดูได้โดยผู้ใช้ใดๆ , แต่สามารถเปลี่ยนได้โดยผู้ใช้ที่มีสิทธิ์ที่ถูกต้องเท่านั้น ในลักษณะคล้ายกัน ผู้ใช้ที่ได้รับสิทธิ์สามารถจัดการ งานผ่านทาง การปฏิบัติงาน แอ็ททริบิวต์สำหรับงานระบบไม่สามารถเปลี่ยนแปลงได้ใน System i Navigator. อย่างไรก็ตาม ระดับความสำคัญของงานระบบบางงานสามารถเปลี่ยนได้ในอินเตอร์เฟซแบบอักษรโดยใช้คำสั่ง Change System Job (CHGSYSJOB)

อินเตอร์เฟซแบบอักษร:

คำสั่ง: Work with Job (WRKJOB) เมื่องานแอ็คทีฟ คุณสามารถดูข้อมูลดังนี้: แอ็ททริบิวต์การรันงาน, ข้อมูล call stack, ข้อมูลการลือกงาน, ข้อมูลรายชื่อไลบรารี, ข้อมูลบันทึกการใช้งาน, ข้อมูลไฟล์เปิด, ข้อมูลไฟล์ override, สถานะ commitment control, สถานะการสื่อสาร, ข้อมูล activation group, ข้อมูล mutex และข้อมูล thread

คำสั่ง: Display Job (DSPJOB)

คำสั่งนี้ แสดงข้อมูลเกี่ยวกับงานดังต่อไปนี้: แอ็ททริบิวต์สถานะงาน, job definition attributes, แอ็ททริบิวต์การรันงาน, ข้อมูลสพูลไฟล์, ข้อมูลบันทึกการใช้งาน, ข้อมูล call stack, ข้อมูลการลือกงาน, ข้อมูล รายชื่อไลบรารี, ข้อมูลไฟล์เปิด, ข้อมูลไฟล์ override, สถานะ commitment control, สถานะการสื่อสาร, ข้อมูล activation group, ข้อมูล mutex, ข้อมูล thread, ไลบรารีลือบันทึก และข้อมูลแอ็ททริบิวต์

การดู call stack:

คุณสามารถดูข้อมูลเกี่ยวกับงาน หรือ call stack ของ thread ได้โดยใช้ System i Navigator หรืออินเตอร์เฟซแบบอักษร

หลักการที่เกี่ยวข้อง

“Call stacks” ในหน้า 35

call stack คือรายการที่จัดลำดับของโปรแกรมหรือโปรซีเจอร์ ทั้งหมดที่รันอยู่ในปัจจุบันสำหรับงาน โปรแกรมและโปรซีเจอร์สามารถ เริ่มต้นได้โดยตรงด้วยคำสั่ง CALL หรือโดยทางอ้อมจากเหตุการณ์อื่น

System i Navigator:

1. จาก System i Navigator ให้ขยาย My Connections → server → Work Management → Active Jobs or Server Jobs ขึ้นอยู่กับชนิดของงานที่คุณต้องการทำงานด้วย
2. คลิกขวาที่ชื่องาน แล้วคลิก Details → Call Stack.

ถ้าคุณต้องการดู call stack สำหรับ thread ให้ทำตามขั้นตอนเหล่านี้:

1. จาก System i Navigator ให้ขยาย My Connections → server → Work Management → Active Jobs or Server Jobs ขึ้นอยู่กับชนิดของงานที่คุณต้องการทำงานด้วย
2. คลิกขวาที่ชื่องาน แล้วคลิก Details → Threads.
3. จากรายการของ thread คลิกขวา thread เฉพาะ แล้วคลิก Details → Call Stack.

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

อินเตอร์เฟซแบบอักษร:

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

เลือกอ็อปชัน 11: แสดงผล call stack, ถ้า แอ็คทีฟ

ถ้าคุณต้องการดู call stack สำหรับ thread หลังจากการออกคำสั่ง WRKJOB หรือ DSPJOB ให้เลือกอ็อปชัน 20: ทำงานกับ thread ถ้าแอ็คทีฟ จากนั้น เลือกอ็อปชัน 10: แสดงอ็อปชัน call stack สำหรับ thread ที่เลือก

การวางแผนบนคิวงาน:

งานถูกวางแผนบนคิวงานโดยการย้ายงานที่มีอยู่จากคิวงานหนึ่งไปยังคิวงานอื่น หรือโดยการส่งงานใหม่ อย่างไม่อย่างหนึ่ง ใช้ System i Navigator เพื่อย้ายงานระหว่างคิวต่างๆ ใช้อินเตอร์เฟซแบบอักษรเพื่อส่งงานใหม่

System i Navigator:

ในการใช้ System i Navigator อินเตอร์เฟซงานต้องมีอยู่แล้วในคิวงานอื่น จากนั้นคุณสามารถย้ายงานจากคิวหนึ่งไปยังคิวงานอื่นได้ (ถ้าต้องการวางแผนใหม่บนคิวงาน ให้ใช้อินเตอร์เฟซบรรทัดรับคำสั่ง)

1. จาก System i Navigator ให้ขยาย **Work Management** → **Job Queues** → **All Job Queues**.
2. คลิกขบวนการที่คุณต้องการย้าย หน้าต่าง Move ที่คุณสามารถระบุคิวปลายทางจะเปิดขึ้น

อินเตอร์เฟซแบบอักษร:

ข้อมูลต่อไปนี้เป็นรายการของวิธีการใช้อินเตอร์เฟซแบบอักษร สำหรับการวางแผนใหม่บนคิวงานใหม่

- Submit Job (SBMJOB): อนุญาตให้งานที่กำลังรันอยู่ส่งงานอื่นไปที่คิวงานเพื่อรันในภายหลังเป็นงานแบ็คชิ่งสามารถวางได้เพียงหนึ่งองค์ประกอบของข้อมูลคำขอเท่านั้นบน message queue ของงานใหม่ ข้อมูลคำขอสามารถเป็นคำสั่ง CL ถ้า routing entry ที่ใช้สำหรับงานระบุ โปรแกรมการประมวลผลคำสั่ง CL (เช่น โปรแกรม QCMD ที่ได้มาจาก IBM)
- Add Job Schedule Entry (ADDJOBSCDE): ระบบส่งงานไปที่คิวงานโดยอัตโนมัติในเวลา และวันที่ที่ระบุไว้ใน job schedule entry
- Submit Database Jobs (SBMDBJOB): ส่งงานไปที่คิวงานเพื่อให้งานสามารถรันเป็นงานแบ็คชิ่งได้ input stream ถูกอ่านจาก ไฟล์ฐานข้อมูลฟิสิกส์หรือจากไฟล์ฐานข้อมูลโลจิคัล อย่างไม่อย่างหนึ่ง ที่มีรูปแบบเรกคอร์ดเดียว คำสั่งนี้อนุญาตให้คุณระบุชื่อไฟล์ของฐานข้อมูล และเมมเบอร์ของไฟล์, ชื่อของคิวงานที่จะใช้ และเลือกตัดสินใจว่างานที่จะส่งสามารถถูกแสดงโดยคำสั่ง Work with Submitted Jobs (WRKSBMJOB) หรือไม่
- Start Database Reader (STRDBRDR): อ่าน batch input stream จากฐานข้อมูล และวางแผนหนึ่ง หรือหลายงานบนคิวงาน
- Transfer Job (TFRJOB): ย้ายงานปัจจุบันไปยังคิวงานอื่นในระบบย่อยแอ็คทีฟ
- Transfer Batch Job (TFRBCHJOB): ย้ายงานปัจจุบันไปยังคิวงานอื่น

การย้ายงานไปที่คิวงานอื่น:

มีหลายเหตุผลที่คุณอาจต้องการย้ายงานไปที่คิวอื่น ตัวอย่างเช่น ในบางครั้งงานอาจถูก backlogged ในคิวเนื่องจากเป็นงานที่ใช้เวลานาน บางครั้งรันไทม์ที่จัดตารางเวลาของงานขัดแย้งกับ งานใหม่ที่มีระดับความสำคัญสูงกว่า วิธีหนึ่งในการจัดการสถานการณ์นี้คือ การย้ายงานที่รอไปยังคิวอื่นที่ไม่ยุ่ง

คุณสามารถใช้อินเตอร์เฟซ System i Navigator หรืออินเตอร์เฟซแบบอักษรเพื่อย้ายงานจากคิวงานหนึ่งไปยังคิวอื่น

System i Navigator:

ในการใช้ System i Navigator ให้ปฏิบัติตามข้อแนะนำต่อไปนี้:

1. จาก System i Navigator ให้ขยาย **Work Management** → **Job Queues** → **All Job Queues**.
2. ระบุตำแหน่งและเปิดคิวที่มีงานอยู่ในปัจจุบัน
3. คลิกขบวนงานที่คุณต้องการย้าย 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)

การเลื่อนระดับความสำคัญของงานขึ้นภายในคิวงาน:

งานทั้งหมดในคิวงานรอในแถวสำหรับการประมวลผล เมื่องานแต่ละงาน ในคิวเสร็จสมบูรณ์ งานถัดไปในแถวจะเริ่มขึ้น ลำดับการประมวลผล ของงานในคิวขึ้นอยู่กับระดับความสำคัญของงาน และจำนวนสูงสุด ของงานที่สามารถรันได้พร้อมกันบนระบบย่อย

ในบางกรณี ความสำคัญของงานก็มีการเปลี่ยนไปเมื่องานดังกล่าวผ่านไปในช่วงวงรอบชีวิต ระดับความสำคัญอาจเพิ่มขึ้นหรือลดลงเมื่อเทียบกับงานอื่น เนื่องจากมีการ เปลี่ยนแปลงดังกล่าวเกิดขึ้น คุณจึงต้องทราบวิธีการเปลี่ยนระดับความสำคัญ ของงานในคิวงาน

ระดับความสำคัญของงานในคิวงานจะช่วยกำหนดได้ว่าเมื่อไรที่งานสมควรจะวิ่ง ไปยังระบบย่อยเพื่อทำงาน ช่วงตั้งแต่ศูนย์ถึงเก้า (ศูนย์คือสำคัญมากที่สุด) กำหนดระดับความสำคัญของงานบนคิวงาน

System i Navigator:

คุณสามารถใช้ System i Navigator เพื่อเปลี่ยนระดับความสำคัญของงานในคิวงานได้

1. จาก System i Navigator ให้ขยาย **My Connections** → *your system* → **Work Management** → **Job Queues** → **Active Job Queues or All Job Queues** → คิวงานซึ่งงานของคุณตั้งอยู่.
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 จะเป็นการย้ายงานไปอยู่ที่รายการงานทั้งหมดที่มีระดับความสำคัญ 3
6. กด F5 เพื่อรีเฟรชหน้าต่าง Job Queue

อินเทอร์เฟซแบบอักษร:

คำสั่ง: Change Job (CHGJOB)

พารามิเตอร์: JOBPTY

ตัวอย่าง: คำสั่งนี้จะเปลี่ยนระดับความสำคัญการจัดตารางเวลาสำหรับงาน PAYROLL เป็น 4 เนื่องจากมีการระบุเฉพาะชื่อ
ธรรมดาของงาน จึงมีได้เพียงหนึ่งงานที่มีชื่อว่า PAYROLL เท่านั้นในระบบ ถ้ามีมากกว่าหนึ่ง คำดีฟอลต์ของ DUPJOB OPT
(*SELECT) จะทำให้พาดูการเลือกถูกแสดงขึ้น ในงานแบบโต้ตอบ

CHGJOB JOB(PAYROLL) JOBPTY(4)

คำแนะนำสำหรับการตั้งค่าระดับความสำคัญงาน:

โดยปกติ ระดับความสำคัญสำหรับงานที่รันในสถานะแวดล้อมแบ็คกราวด์ต่ำกว่าระดับความสำคัญสำหรับงานในสถานะแวดล้อมแบบโต้ตอบ นอกจากนี้ การแบ่งเวลา ควรน้อยพอที่โปรแกรมการวนซ้ำจะไม่ครอบครองเวลาของโพรเซสเซอร์ และ activity level

คุณอาจต้องการให้ระดับความสำคัญสำหรับงานของผู้ควบคุมระบบสูงกว่า ระดับความสำคัญของงานอื่น เพื่อให้ผู้ควบคุมระบบสามารถตอบสนองต่อความต้องการ การระบบได้อย่างมีประสิทธิภาพ

ถ้าคุณใช้ QCTL เป็นระบบย่อยการควบคุม ผู้ควบคุมเครื่องจะรันที่ ระดับความสำคัญสูงกว่าโดยอัตโนมัติหลังจากการ signing on ที่คอนโซลที่เป็นเช่นนี้เพราะ QCTL เรตงานคอนโซลโดยใช้คลาส QCTL ซึ่งระบุระดับความสำคัญที่สูงกว่า

อีกวิธีหนึ่งที่คุณสามารถตั้งค่าระบบของคุณเพื่อให้ผู้ควบคุมเครื่องสามารถรันที่ระดับความสำคัญสูงกว่าได้ คือขั้นตอนต่อไป
นี้:

1. ใส่เพิ่ม routing entry ที่ระบบย่อยพร้อมด้วยข้อมูลการเรตเฉพาะ และ ระบุคลาส QSYS/QCTL
2. สร้างรายละเอียดของงานใหม่สำหรับผู้ควบคุมเครื่อง โดยระบุข้อมูลการเรตเฉพาะที่เหมือนกับข้อมูลการเรตที่คุณใช้ใน routing entry
3. เปลี่ยนโปรไฟล์ผู้ใช้ของผู้ควบคุมเครื่องเพื่อระบุรายละเอียดของงานใหม่
4. ขณะนี้เมื่อผู้ควบคุมเครื่อง sign on ที่ระบบย่อยนั้น งานจะเรตโดยใช้คลาส QCTL ซึ่งระบุระดับความสำคัญที่สูงกว่า คลาสที่ใช้โดยงานแบบโต้ตอบปกติ

ระดับความสำคัญการรันงานคือระดับความสำคัญสูงสุดซึ่ง thread ใดๆ ในงานสามารถรันได้แต่ละ thread อาจมีระดับความสำคัญ thread ของตนเองซึ่ง ต่ำกว่าระดับความสำคัญงาน คำสั่ง Change Job (CHGJOB) จะเปลี่ยนเฉพาะระดับความสำคัญของงานเท่านั้น สามารถใช้คำสั่ง Change Job (QWTCHEGJB) API เพื่อเปลี่ยนระดับความสำคัญงานหรือระดับความสำคัญ thread อย่างใดอย่างหนึ่ง

การส่งงานครั้งเดียว:

เมื่อคุณต้องการรันงานเพียงครั้งเดียว ไม่ว่าจะรันในทันที หรือรันในวันที่ และเวลาที่จัดตารางไว้ ให้ใช้คำสั่ง Submit Job (SBMJOB) วิธีนี้จะ วางงานบนคิวงานในทันที

ถ้าต้องการส่งงานแบ็คทีเพียงครั้งเดียว ให้ใช้อินเทอร์เฟซแบบอักษร

คำสั่ง: Submit Job (SBMJOB)

คำสั่ง SBMJOB จะส่งงานไปที่คิวงานแบ็ตช์ โดยระบุรายละเอียดของงาน และโดยระบุคำสั่ง CL หรือข้อมูลคำขอ หรือระบุข้อมูลการเรดท์ที่จะรันโปรแกรม ถ้าคุณต้องการรันคำสั่ง CL เดียวในงานแบ็ตช์หนึ่ง ให้ใช้พารามิเตอร์ CMD บน SBMJOB ซึ่งจะทำให้การ ตรวจสอบไวยากรณ์และอนุญาตการพร้อมท์

ตัวอย่าง: ในตัวอย่างต่อไปนี้ คำสั่ง SBMJOB จะส่งงานที่มีชื่อว่า WSYS ไปยังคิวงาน QBATCH โดยใช้รายละเอียดของงาน QBATCH พารามิเตอร์ CMD ให้คำสั่ง CL ที่จะรันในงาน

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

หลักการที่เกี่ยวข้อง

“คำสั่งการส่งงาน” ในหน้า 65

คำสั่งอินเตอร์เฟซแบบอักษรนี้จะควบคุมเวลาที่งาน จะถูกรีลิสในคิวงาน คำสั่งนี้เป็นวิธีที่ง่ายในการจัดตารางเวลางานที่ต้องรัน เพียงครั้งเดียวเท่านั้น คำสั่งนี้อนุญาตให้คุณใช้แอตทริบิวต์งานที่กำหนดสำหรับ งานปัจจุบันของคุณได้หลายแอตทริบิวต์

การดูข้อมูล job affinity:

แต่ละงานบนระบบมีหน่วยความจำ และโพรเซสเซอร์ของข้อมูล affinity

ข้อมูล affinity อธิบายว่า thread จะมีความสัมพันธ์กับกลุ่มของโพรเซสเซอร์ และหน่วยความจำเดียวกันกับ initial thread เมื่อ thread ถูกเริ่มต้นหรือไม่ และยังระบุถึงระดับของระบบที่พยายามจะรักษาไว้ซึ่งความสัมพันธ์ระหว่าง thread และ เซ็ตย่อยของรีซอร์สระบบที่ได้กำหนดไว้แล้ว นอกจากนั้น ข้อมูล affinity ยังระบุว่า งานจะถูกแบ่งกลุ่มพร้อมกับ งานอื่น เพื่อให้งานมีความสัมพันธ์กับเซ็ตย่อยเดียวกันกับเซ็ตย่อยของรีซอร์สของระบบหรือไม่

โดยการรวมกลุ่ม thread เข้าด้วยกันที่จะแบ่งใช้งานชุดของข้อมูลทั่วไปในแหล่งเก็บข้อมูลหลัก การแคชของระบบของคุณและอัตราการเข้าถึงข้อมูลหน่วยความจำจะดีขึ้น

System i Navigator:

1. จาก System i Navigator, ให้ขยาย My Connections → your system → Work Management → Active Jobs.

หมายเหตุ: คุณสามารถดูข้อมูล affinity ของงานจากตำแหน่งใดๆ ก็ได้ภายในการจัดการระบบงาน ที่ซึ่งคุณสามารถเรียกดูงานได้

2. คลิกขบวนการที่คุณต้องการดู และคลิก Properties.
3. บนหน้า Resources คุณสามารถดูข้อมูล ความสัมพันธ์ของหน่วยความจำ และโพรเซสเซอร์ได้

อินเตอร์เฟซแบบอักษร:

คำสั่ง: Work with Job (WRKJOB)

เลือก อ็อปชัน 3: แสดงแอตทริบิวต์ job run ถ้าแอ็คทีฟ

การจัดการรายละเอียดของงาน

เนื่องจากรายละเอียดของงานมีชุดเฉพาะของแอตทริบิวต์ที่เกี่ยวข้องกับงาน รายละเอียดของงานเดียวกันจึงสามารถถูกใช้โดยงานหลายงานได้ ดังนั้น ถ้า คุณใช้รายละเอียดของงาน คุณไม่ต้องระบุพารามิเตอร์เดียวกันซ้ำสำหรับ แต่ละงาน คุณสามารถสร้างรายละเอียดของงานเพื่ออธิบายงานแบ็ตช์ หรืองานแบบโต้ตอบ คุณยังสามารถสร้าง descriptions เฉพาะสำหรับ ผู้ใช้แต่ละรายของระบบได้ด้วย รายละเอียดของงานถูกสร้างและจัดการโดยใช้อินเตอร์เฟซแบบอักษร

การสร้างรายละเอียดของงาน:

คุณสามารถใช้อินเตอร์เฟซแบบอักษร คำสั่ง Work With Job Description (WRKJOB) หรือคำสั่ง Create Job Description (CRTJOB) เพื่อเปลี่ยนรายละเอียดของงาน

คำสั่ง: Create Job Description (CRTJOB)

ตัวอย่าง: ในตัวอย่างนี้ รายละเอียดของงานถูกสร้างโดยมีชื่อว่า INT4 ในไลบรารีปัจจุบันของผู้ใช้ รายละเอียดของงานนี้ใช้สำหรับงานแบบโต้ตอบ และถูกใช้โดย Department 127 เมื่อคุณ sign on คุณต้องพิมพ์รหัสผ่าน อักษร QCMDI ถูกใช้เป็นข้อมูลการเรดซึ่งจะนำไปเปรียบเทียบกับตาราง การเรดของระบบย่อยที่งานรันอยู่ ข้อความสอบถามทั้งหมดจะถูกเปรียบเทียบกับ entries ในรายการคำตอบของระบบ เพื่อกำหนดว่าจะมีออกคำตอบโดยอัตโนมัติหรือไม่

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

คำสั่งนี้จะสร้างรายละเอียดของงานที่มีชื่อว่า BATCH3 ใน ไลบรารีปัจจุบันของผู้ใช้ งานที่ใช้ description นี้จะถูกวางบน คิวงาน NIGHTQ ระดับความสำคัญสำหรับงานที่ใช้ description นี้และเอาต์พุต ที่สพูลคือ 4. QCMDDB คือข้อมูลการเรดที่จะนำไปเปรียบเทียบกับ entries ในตารางการเรดของระบบย่อยที่งานจะรัน จะใช้ไค้ดบัญชีผู้ใช้ NIGHTQ012345 เมื่อบันทึกสถิติเกี่ยวกับบัญชีผู้ใช้สำหรับงานที่ใช้รายละเอียดของงานนี้

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

หมายเหตุ: ค่าในรายละเอียดของงานโดยปกติจะถูกใช้เป็นค่าดีฟอลต์ของพารามิเตอร์ที่สอดคล้องกันในคำสั่ง Batch Job (BCHJOB) และ Submit Job (SBMJOB) เมื่อไม่มีการระบุพารามิเตอร์ ค่าในรายละเอียดของงานสามารถถูกแทนที่ค่าเดิมโดยค่าที่ระบุบนคำสั่ง BCHJOB และ SBMJOB

หลักการที่เกี่ยวข้อง

“รายละเอียดของงาน” ในหน้า 33

รายละเอียดของงานอนุญาตให้คุณสร้างชุดของแอ็ททริบิวต์งาน ที่จะถูกบันทึกและพร้อมใช้งานสำหรับการใช้ในลักษณะต่างๆ รายละเอียดของงานสามารถใช้เป็น ซอร์สสำหรับแอ็ททริบิวต์งานบางอย่างที่บอกให้ระบบทราบถึงวิธีการ รันงาน แอ็ททริบิวต์บอกให้ระบบทราบว่า จะเริ่มงานเมื่อไร เรียกว่างาน มาจากที่ไหน และจะรันงานอย่างไร คุณสามารถคิดว่ารายละเอียดของงาน เป็นเท็มเพลตที่งานหลายงานสามารถใช้ได้ ดังนั้นจึงลดจำนวนของพารามิเตอร์ เฉพาะที่คุณต้องเซ็ตสำหรับงานแต่ละงาน

การเปลี่ยนรายละเอียดของงาน:

คุณสามารถใช้อินเตอร์เฟซแบบอักษร คำสั่ง Work With Job Description (WRKJOB) หรือคำสั่ง Change Job Description (CHGJOB) เพื่อเปลี่ยนรายละเอียดของงาน

คำสั่ง: Change Job Description (CHGJOB)

งานทั้งหมดซึ่งใช้รายละเอียดของงานที่ถูกเริ่มหลังจากการเปลี่ยน รายละเอียดของงานจะได้รับผลกระทบ ถ้าคุณเปลี่ยนพารามิเตอร์งานเป็นค่าที่แตกต่างจาก ค่าที่ระบุในรายละเอียดของงาน พารามิเตอร์นั้นจะไม่มีผล

การใช้รายละเอียดของงาน:

วิธีทั่วไปในการใช้รายละเอียดของงานคือ โดยการระบุรายละเอียดของงาน ในคำสั่ง Submit Job (SBMJOB) พารามิเตอร์ job description (JOBDESC) คือ ที่ที่คุณจะระบุรายละเอียดของงานที่คุณต้องการให้งานนี้ใช้ เมื่อคุณกำหนดงานแบ็ตซ์ คุณสามารถใช้รายละเอียดของงานในหนึ่งจากสอง วิธีดังนี้:

- ใช้รายละเอียดของงานที่ระบุโดยไม่แทนทับค่าเดิมของแอ็ททริบิวต์ ตัวอย่างเช่น:

```
SBMJOB JOB(OEDAILY) JOBDESC(QBATCH)
```

- ใช้รายละเอียดของงานที่ระบุแต่แทนทับบางแอ็ททริบิวต์ (การใช้คำสั่ง BCHJOB หรือ SBJOB) ตัวอย่างเช่น ถ้าต้องการแทนทับค่าเดิมของไฟล์บันทึกข้อความในรายละเอียดของงาน QBATCH ให้คุณระบุ:

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

ข้างล่างนี้เป็นคำสั่งเพิ่มเติมที่สนับสนุนพารามิเตอร์รายละเอียดของงาน :

- Batch Job (BCHJOB): คำสั่งนี้บังคับจุดเริ่มต้นของงานแบ็ตซ์ใน batch input stream คำสั่งนี้ยังสามารถระบุค่าที่แตกต่างสำหรับ แอ็ททริบิวต์สำหรับงานแทนแอ็ททริบิวต์ที่ระบุในรายละเอียดของงาน หรือโปรไฟล์ผู้ใช้สำหรับงานนี้ ค่าที่มีอยู่ในรายละเอียดของงาน หรือในโปรไฟล์ผู้ใช้ที่มีชื่อในรายละเอียดของงานจะถูกใช้สำหรับพารามิเตอร์ส่วนใหญ่ที่ไม่ได้ระบุในคำสั่ง BCHJOB
- Add Prestart Job Entry (ADDPJE): คำสั่ง Add Prestart Job Entry (ADDPJE) ใส่เพิ่ม prestart job entry ที่ subsystem description ที่ระบุ entry จะระบุงานแบบ prestart ที่อาจถูกเริ่มขึ้น เมื่อระบบย่อยเริ่มต้น หรือเมื่อมีการป้อนคำสั่ง 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: คุณไม่สามารถแทนทับค่าเดิมแอ็ททริบิวต์รายละเอียดของงานสำหรับ งานแบบ autostart, งานเวิร์กสเตชัน หรืองานสื่อสาร

หลักการที่เกี่ยวข้อง

“รายละเอียดของงาน” ในหน้า 33

รายละเอียดของงานอนุญาตให้คุณสร้างชุดของแอ็ททริบิวต์งาน ที่จะถูกบันทึกและพร้อมใช้งานสำหรับการใช้ในลักษณะต่างๆ รายละเอียดของงานสามารถใช้เป็น ซอร์สสำหรับแอ็ททริบิวต์งานบางอย่างที่บอกให้ระบบทราบถึงวิธีการ ใช้งาน แอ็ททริบิวต์บอกให้ระบบทราบว่า จะเริ่มงานเมื่อไร เรียกงาน มาจากที่ไหน และจะรันงานอย่างไร คุณสามารถคิดว่ารายละเอียดของงาน เป็นเท็มเพลตที่งานหลายงานสามารถใช้ได้ ดังนั้นจึงลดจำนวนของพารามิเตอร์ เฉพาะที่คุณต้องเซ็ทสำหรับงานแต่ละงาน

การควบคุมซอร์สของแอ็ททริบิวต์งาน:

แอ็ททริบิวต์ที่ระบุจะกำหนดให้กับงานมาจากซอร์สห้าซอร์ส; รายละเอียดของงาน, โปรไฟล์ผู้ใช้ของผู้ใช้, ค่ากำหนดของระบบ, งานที่ออกคำสั่ง Submit Job (SBMJOB) และเวิร์กสเตชัน (งานแบบโต้ตอบเท่านั้น) คุณควบคุมซอร์สที่ระบุจะดึงข้อมูลแอ็ททริบิวต์งานเฉพาะ มาได้ โดยการระบุซอร์สในรายละเอียดของงาน ถ้าต้องการ แก้ไขรายละเอียดของงาน ให้ใช้อินเตอร์เฟซแบบอักษร

คำสั่ง: Change Job Description (CHGJOBDESC)

เพื่อควบคุม แอ็ตทริบิวต์งานและบอกให้ระบบย่อยทราบว่าจะเรียกแอ็ตทริบิวต์งานจากอ็อบเจกต์ ระบบอื่นได้ที่ไหนและเมื่อใดให้ใช้อย่างใดอย่างหนึ่งดังต่อไปนี้:

- *JOBID: บอกให้งานเรียกแอ็ตทริบิวต์ของงานมาจากรายละเอียดของงาน
- *USRPRF: บอกให้งานเรียกแอ็ตทริบิวต์ของงานมาจากโปรไฟล์ผู้ใช้ของผู้ใช้
- *SYSVAL: บอกให้งานเรียกแอ็ตทริบิวต์ของงานมาจากค่ากำหนดของระบบ
- *CURRENT: บอกให้งานเรียกแอ็ตทริบิวต์ของงานจากงานที่ออกคำสั่ง Submit Job (SBMJOB)
- *WRKSTN: บอกให้งานเรียกแอ็ตทริบิวต์ของงานมาจากเวิร์กสเตชันที่มีงาน (งานแบบโต้ตอบเท่านั้น)

การลบรายละเอียดของงาน:

คุณสามารถใช้อินเตอร์เฟซแบบอักษร, คำสั่ง Work With Job Description (WRKJOBID), หรือคำสั่ง Delete Job Description (DLTJOBID) เพื่อลบรายละเอียดของงาน

คำสั่ง: Delete Job Description (DLTJOBID)

หมายเหตุ: งานที่อยู่ระหว่างการประมวลผลแล้วจะไม่ได้รับผลกระทบโดยคำสั่งนี้

การจัดการงานแบ็ตซ์

งานที่ไม่ต้องการข้อมูลผู้ใช้ในการรันสามารถถูกประมวลผล เป็นงานแบ็ตซ์โดยปกติ งานแบ็ตซ์เป็นงานที่มีระดับความสำคัญต่ำและอาจต้องการสถานะแวดล้อมระบบพิเศษในการรัน

การส่งงานแบ็ตซ์:

เนื่องจากโดยปกติ งานแบ็ตซ์เป็นงานที่มีระดับความสำคัญต่ำที่ต้องการ สถานะแวดล้อมระบบพิเศษในการรัน (เช่น การรันในเวลากลางคืน) งานแบ็ตซ์จึงถูกวางในคิวงานแบ็ตซ์ในคิวงาน งานแบ็ตซ์จะได้รับตารางเวลารันไทม์ และระดับความสำคัญ ถ้าต้องการส่งงานไปที่คิวงานแบ็ตซ์ให้คุณ ใช้อินเตอร์เฟซแบบอักษรและหนึ่งในสองคำสั่งต่อไปนี้

คำสั่ง: Submit Job (SBMJOB)

คำสั่ง: Submit Database Job (SBMDBJOB)

ความแตกต่าง ระหว่างสองคำสั่งนี้คือชื่อของงาน:

- คำสั่ง SBMJOB จะส่งงานไปที่คิวงานแบ็ตซ์โดยระบุรายละเอียดของงาน และโดยระบุคำสั่ง CL หรือข้อมูลคำขอ หรือระบุข้อมูลการเรดที่จจะรันโปรแกรม ถ้าคุณต้องการรันคำสั่ง CL เดียวในงานแบ็ตซ์หนึ่ง ให้ใช้พารามิเตอร์ CMD บน SBMJOB ซึ่งจะทำการ ตรวจสอบไวยากรณ์และอนุญาตการพร้อมดี
- คำสั่ง SBMDBJOB สามารถใช้เพื่อส่งงานจากไฟล์ฐานข้อมูลไปที่คิวงานแบ็ตซ์สำหรับงานเหล่านี้ รายละเอียดของงาน จะมาจากข้อความ BCHJOB ใน input stream

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

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

หมายเหตุ: หากคุณได้รับข้อความแจ้งว่างานไม่ถูกส่ง คุณสามารถแสดง สพลไฟล์บันทึกการใช้งานเพื่อค้นหาข้อผิดพลาด ใช้คำสั่ง WRKJOB ระบุงานที่ไม่ถูกจัดตารางเวลา เลือกอ็อปชัน 4 สำหรับสพลไฟล์ แสดงสพลไฟล์บันทึกการใช้งานเพื่อค้นหาข้อผิดพลาด

หลักการที่เกี่ยวข้อง

“วิธีการเริ่มงานแบ็ตช์” ในหน้า 43

เมื่อผู้ใช้ส่งงานแบ็ตช์งานจะรวบรวมข้อมูลจากอ็อบเจ็กต์ระบบต่างๆ ก่อนที่งานจะถูกวางบนคิวงาน

“คำสั่งการส่งงาน” ในหน้า 65

คำสั่งอินเตอร์เฟสแบบอักขระนี้ควบคุมเวลาที่งาน จะถูกรีลีส์ในคิวงาน คำสั่งนี้เป็นวิธีที่ง่ายในการจัดตารางเวลางานที่ต้องรัน เพียงครั้งเดียวเท่านั้น คำสั่งนี้อุญาตให้คุณใช้แอ็ดทริบิวต์งานที่กำหนดสำหรับ งานปัจจุบันของคุณได้หลายแอ็ดทริบิวต์

ข้อมูลที่เกี่ยวข้อง

งาน QPRTJOB

ไฟล์ข้อมูลออนไลน์:

ไฟล์ข้อมูลออนไลน์คือไฟล์ข้อมูลที่อยู่เป็นส่วนหนึ่งของงานแบ็ตช์ เมื่องานถูกอ่านโดยเครื่องอ่านหรือคำสั่งการส่งงาน คุณใช้ SBMDBJOB หรือ STRDBRDR เพื่อจัดคิว CL batch stream (stream ของคำสั่ง CL ที่จะรัน) CL batch stream นั้นสามารถรวมข้อมูลที่จะวางเข้าไฟล์ออนไลน์ (ไฟล์ชั่วคราว) เมื่องานสิ้นสุดลง ไฟล์ข้อมูลออนไลน์จะถูกลบออก

ไฟล์ข้อมูลออนไลน์ถูกค้นในงานด้วยคำสั่ง //DATA ที่ตอนต้นของไฟล์ และถูกค้นโดย end-of-data delimiter ที่ตอนท้ายของไฟล์

end-of-data delimiter อาจเป็นสตริงอักขระที่ผู้ใช้กำหนด หรือค่าดีฟอลต์ของ // // ต้องปรากฏในตำแหน่ง 1 และ 2 ถ้าข้อมูลของคุณมี // ในตำแหน่ง 1 และ 2 คุณควรจะใช้เซตของอักขระที่ไม่ซ้ำกัน เช่น // *** END OF DATA เพื่อระบุว่าสิ่งนี้เป็น end-of-data delimiter ที่ไม่ซ้ำกัน พารามิเตอร์ ENDCHAR บนคำสั่ง //DATA ควรเป็นโค้ด เช่น:

```
ENDCHAR(‘// *** END OF DATA’)
```

หมายเหตุ: ไฟล์ ข้อมูลออนไลน์สามารถเข้าถึงได้เฉพาะในระหว่างขั้นตอนการเรอต์แรกของงานแบ็ตช์เท่านั้น ถ้างานแบ็ตช์มีคำสั่ง Transfer Job (TFRJOB), Reroute Job (RRTJOB) หรือ Transfer Batch Job (TFRBCHJOB) ไฟล์ข้อมูลออนไลน์จะไม่สามารถเข้าถึงได้ในขั้นตอนการเรอต์ใหม่.

ไฟล์ข้อมูล ออนไลน์สามารถเป็นแบบระบุชื่อหรือแบบไม่ระบุชื่อ ใดอย่างหนึ่ง สำหรับไฟล์ข้อมูลออนไลน์แบบไม่ระบุชื่อ อาจมีการระบุ QINLINE เป็นชื่อไฟล์ในคำสั่ง //DATA หรือไม่มีการระบุชื่อใดอย่างหนึ่ง สำหรับไฟล์ข้อมูลออนไลน์ที่ระบุชื่อ จะมีการระบุชื่อไฟล์

ตัว *ไฟล์ข้อมูลออนไลน์แบบระบุชื่อ* มีลักษณะดังต่อไปนี้:

- มีชื่อที่ไม่ซ้ำกันในงาน ไฟล์ข้อมูลออนไลน์ต้องมีชื่อที่ไม่ซ้ำกัน
- สามารถใช้ได้มากกว่าหนึ่งครั้งในงานหนึ่ง
- ในแต่ละครั้งที่เปิดไฟล์ ไฟล์จะถูกจัดตำแหน่งอยู่ที่เร็กคอร์ดแรก

ในการใช้ไฟล์ข้อมูลออนไลน์แบบระบุชื่อ คุณต้องระบุชื่อไฟล์ในโปรแกรม หรือใช้คำสั่งแทนที่เพื่อเปลี่ยนชื่อไฟล์ที่ระบุ ในโปรแกรมเป็นชื่อของไฟล์ข้อมูลออนไลน์ ใดอย่างหนึ่ง ไฟล์ต้องถูกเปิด สำหรับการอินพุตเท่านั้น

ตัว ไฟล์ข้อมูลออนไลน์แบบไม่ระบุชื่อ มีลักษณะดังต่อไปนี้:

- ชื่อของไฟล์คือ QINLINE (ในงานแบ็คซัพไฟล์ข้อมูลออนไลน์แบบไม่ระบุชื่อทั้งหมดจะ ถูกตั้งชื่อเหมือนกัน)
- สามารถใช้ได้ครั้งเดียวเท่านั้นในงานหนึ่ง
- เมื่อมีไฟล์ข้อมูลออนไลน์แบบไม่ระบุชื่ออยู่มากกว่าหนึ่งไฟล์ในงาน ไฟล์ต้องอยู่ใน input stream ในลำดับเดียวกับลำดับเมื่อไฟล์ถูกเปิด

ถ้าต้องการใช้ไฟล์ข้อมูลออนไลน์แบบไม่ระบุชื่อ ให้ทำอย่างไรอย่างหนึ่งดังนี้:

- ระบุ QINLINE ในโปรแกรม
- ใช้คำสั่งการแทนที่เพื่อเปลี่ยนชื่อที่ระบุในโปรแกรมเป็น QINLINE

ถ้าภาษาชั้นสูงของคุณต้องการชื่อไฟล์ที่ไม่ซ้ำกันภายในหนึ่งโปรแกรม, คุณสามารถใช้ QINLINE เป็นชื่อไฟล์ได้เพียงครั้งเดียวเท่านั้น ถ้าคุณต้องการใช้ไฟล์ข้อมูลออนไลน์แบบไม่ระบุชื่อมากกว่าหนึ่งไฟล์ คุณสามารถใช้คำสั่งการแทนที่ไฟล์ในโปรแกรมเพื่อระบุ QINLINE สำหรับไฟล์ข้อมูลออนไลน์แบบไม่ระบุชื่อเพิ่มเติม

หมายเหตุ: ถ้าคุณรันคำสั่งแบบมีเงื่อนไขและประมวลผลไฟล์ข้อมูลออนไลน์แบบไม่ระบุชื่อ มากกว่าหนึ่งไฟล์ จะไม่สามารถคาดการณ์ผลลัพธ์ได้ถ้าใช้ไฟล์ข้อมูลออนไลน์แบบไม่ระบุชื่อที่ไม่ถูกต้อง

ข้อควรพิจารณาในการเปิดไฟล์ข้อมูลออนไลน์:

คุณต้องพิจารณาองค์ประกอบเหล่านี้เมื่อคุณเปิดไฟล์ข้อมูลออนไลน์

- ความยาวเร็กคอร์ดระบุความยาวของอินพุตเร็กคอร์ด (ความยาวเร็กคอร์ดเป็นสิ่งที่เลือกได้) เมื่อความยาวเร็กคอร์ดเกินกว่าความยาวของข้อมูล จะมีการส่ง ข้อความไปที่โปรแกรมของคุณ ข้อมูลจะถูกเสริมด้วยพื้นที่เปล่า เมื่อความยาวเร็กคอร์ด น้อยกว่าความยาวของข้อมูล เร็กคอร์ดจะถูกตัดปลาย
- เมื่อมีการระบุไฟล์ในโปรแกรม ระบบจะค้นหาไฟล์ตามไฟล์ข้อมูลออนไลน์ที่ระบุชื่อ ก่อนที่จะค้นหาไฟล์ในไลบรารี ดังนั้น ถ้าไฟล์ข้อมูลออนไลน์ที่ระบุชื่อมีชื่อเหมือนกับไฟล์ที่ไม่ใช่ไฟล์ข้อมูลออนไลน์ ไฟล์ข้อมูลออนไลน์จะถูกใช้เสมอ แม้ว่าชื่อไฟล์ จะถูกต้องตามชื่อไลบรารี
- ไฟล์ข้อมูลออนไลน์ที่ระบุชื่อสามารถถูกแบ่งใช้ระหว่างโปรแกรมต่างๆ ในงานเดียว กันได้โดยการระบุ SHARE(*YES) บน คำสั่งการสร้างไฟล์หรือการแทนที่ไฟล์ ตัวอย่างเช่น ถ้าคำสั่งการแทนที่ไฟล์ระบุว่าไฟล์ที่มีชื่อว่า INPUT และ SHARE (*YES) อยู่ในงานแบ็คซัพพร้อมกับไฟล์ข้อมูลออนไลน์ที่มีชื่อว่า INPUT โปรแกรมใดๆที่รันในงานที่ระบุชื่อไฟล์ INPUT จะแบ่งใช้ไฟล์ข้อมูลออนไลน์ที่มีชื่อเหมือนกัน ไฟล์ข้อมูลออนไลน์ที่ไม่ได้ระบุชื่อไม่ สามารถถูกแบ่งใช้ระหว่างโปรแกรมต่างๆ ในงานเดียวกัน
- เมื่อคุณใช้ไฟล์ข้อมูลออนไลน์ คุณควรตรวจสอบให้แน่ใจว่าได้ระบุชนิดของไฟล์ที่ถูกต้องบนคำสั่ง //DATA ตัวอย่างเช่น ถ้าไฟล์จะถูกใช้เป็น ไฟล์ต้นฉบับ ชนิดของไฟล์บนคำสั่ง //DATA ต้องเป็นต้นฉบับ
- ไฟล์ข้อมูลออนไลน์ต้องถูกเปิดสำหรับอินพุตเท่านั้น

การเริ่มงานแบ็คซัพที่กำลังรออยู่ในคิวงาน:

ในบางโอกาส คุณอาจต้องการบังคับให้งานเริ่มในทันที ในขณะที่การย้ายงานไปที่คิวงานที่ไม่ยุ่งเป็นวิธีการที่มีประสิทธิภาพสูงสุด แต่มีวิธีการอื่นที่คุณสามารถใช้ได้

เพื่อเริ่มงานแบ็ตซ์ในอันดับแรกให้ตรวจสอบสถานะของคิวงานที่งานตั้งอยู่ และพิจารณาว่าการย้ายงานไปยังคิวอื่นเป็นวิธีที่เหมาะสมที่สุดสำหรับสถานการณ์ของคุณหรือไม่ (My Connections → server → Work Management → Job Queues → Active Job Queues หรือ All Job Queues)

ถ้าการย้ายงานไปที่คิวงานอื่นเป็นสิ่งที่ไม่เหมาะสม, คุณสามารถพักงานที่รันอยู่ แล้วเปลี่ยนระดับความสำคัญของงานที่คุณต้องการเริ่ม อย่างไรก็ตาม ระวังเมื่อใช้วิธีการนี้ เนื่องจากงานที่ถูกพัก ยังคงรวมอยู่ในการนับงานแอ็คทีฟสูงสุด

ในการเปลี่ยนระดับความสำคัญของงาน และระยะเวลาที่งานควรรัน ให้ใช้ข้อแนะนำต่อไปนี้:

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**.

หลักการที่เกี่ยวข้อง

“วิธีการเริ่มงานแบ็ตซ์” ในหน้า 43

เมื่อผู้ใช้ส่งงานแบ็ตซ์ งานจะรวบรวมข้อมูลจากอ็อบเจกต์ระบบต่างๆ ก่อนที่งานจะถูกวางบนคิวงาน

ข้อมูลที่เกี่ยวข้อง

งาน QPRTJOB

การจัดการงานแบบโต้ตอบ

งานแบบโต้ตอบเริ่มต้นเมื่อคุณ sign on ที่ระบบ หรือถ่ายโอน ไปที่งานรองหรืองานกลุ่ม งานแบบโต้ตอบจบลงเมื่อคุณ sign off การทำงานจากจอภาพช่วยให้คุณโต้ตอบกับระบบโดยการออก คำสั่ง, การใช้ฟังก์ชันคีย์ และการรันโปรแกรม และแอ็พพลิเคชัน ข้อมูลต่อไปนี้ อธิบายวิธีการต่างๆ สำหรับการจัดการและการควบคุม งานแบบโต้ตอบ

การควบคุมงาน และเวิร์กสเตชันที่ inactive:

คุณสามารถควบคุมจำนวนเวลาที่เวิร์กสเตชันยังคง inactive ก่อนที่ระบบย่อยจะส่งข้อความ (เรียกว่าโทมเอดต์) ได้ โดยการระบุช่วงเวลา ในค่ากำหนดของระบบ Time-out interval for inactive jobs (QINACTITV) การควบคุมงาน inactive เป็นการรักษาความปลอดภัยเพื่อไม่ให้ผู้ใช้ปล่อยให้ จอแสดงผลที่ signed on inactive

ระบบกำหนดว่าเวิร์กสเตชัน inactive ได้อย่างไร

ระบบย่อย จะกำหนดว่าเวิร์กสเตชันนั้น inactive ถ้าเงื่อนไขต่อไปนี้ เป็นจริง:

- งานไม่ได้ประมวลผล transactions เพิ่มเติมใดๆ ในระหว่าง ช่วงของตัวจับเวลา

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

- สถานะของงานคือรอการแสดงผล
- งานไม่ถูกตัดการเชื่อมต่อ
- สถานะของงานไม่เปลี่ยนแปลง

- ระบบย่อยที่งานกำลังรันอยู่ไม่ได้อยู่ในสภาพที่ถูกจำกัด

การจัดการงาน inactive

เพื่อจัดการงาน inactive ที่พบในระบบ ให้ใช้ค่ากำหนดของระบบ When a job reaches time-out (QINACTMSGQ) เพื่อกำหนดอ็อปชันการประมวลผล ให้เลือกจากอ็อปชันดังต่อไปนี้:

- ตั้งค่ากำหนดของระบบ QINACTMSGQ เป็นชื่อ message queue
ถ้าคุณระบุชื่อ message queue สำหรับค่ากำหนดของระบบ QINACTMSGQ, ผู้ใช้หรือ โปรแกรมสามารถ จะมอนิเตอร์ message queue และดำเนินการ action ที่จำเป็น เช่น การจบงาน ได้
ถ้าเวิร์กสเตชันที่มี secondary job pair มีสถานะ inactive ระบบจะส่งข้อความสองข้อความ (หนึ่งข้อความสำหรับแต่ละงานของ secondary job pairs) ไปที่ message queue จากนั้น ผู้ใช้ หรือโปรแกรมสามารถใช้คำสั่ง ENDJOB กับหนึ่ง หรือทั้งสองงานรอง หรือคำสั่ง DSCJOB กับงานแอคทีฟที่จอแสดงผล
- ตั้งค่ากำหนดของระบบ QINACTMSGQ เป็น *DSCJOB
ถ้าคุณระบุ *DSCJOB สำหรับค่ากำหนดของระบบ QINACTMSGQ ระบบจะตัดการเชื่อมต่องานทั้งหมดที่เวิร์กสเตชันระบบส่งข้อความที่ระบุว่างานทั้งหมดที่เวิร์กสเตชันได้ถูกตัดการเชื่อมต่อ จาก QSYSOPR หรือ message queue ที่ตั้งค่า (message queue ที่ตั้งค่า คือ message queue ที่ระบุในพารามิเตอร์ MSGQ ของ display device description โดยค่าดีฟอลต์ค่านี้คือ QSYS หรือ QSYSOPR) ถ้างานแบบโต้ตอบไม่สนับสนุนการตัดการเชื่อมต่อ (เช่น TELNET sessions ที่ใช้ QPADEVxxxx device descriptions) งานจะจบแทน
ข้อความ จะถูกส่งสำหรับแต่ละช่วงเวลาทำงาน 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 นอกจากนี้งาน MRT ในสถานะแวดล้อม System/36 ยังถูกแยกออก ด้วยเนื่องจากเป็นงานแบ็คซ์

การจบงานแบบโต้ตอบ:

คุณสามารถใช้วิธีต่างๆหลายแบบในการจบงานแบบโต้ตอบ

คุณสามารถใช้ System i Navigator เพื่อจบงาน

1. จากหน้าต่าง Confirm Delete/End คุณสามารถระบุได้ว่า คุณต้องการให้งานแบบโต้ตอบนี้จบในลักษณะที่ควบคุม หรือจบแบบทันที
2. คุณสามารถใช้คำสั่งอินเตอร์เฟสแบบอักขระ End Job (ENDJOB)
3. ในการจบงานแบบโต้ตอบโดยทันทีโดยใช้อินเตอร์เฟสแบบอักขระ ให้ใช้คำสั่ง Sign Off (SIGNOFF) ที่เวิร์กสเตชัน ถ้าต้องการจบการเชื่อมต่อผ่านทางเน็ตเวิร์ก ให้ใช้พารามิเตอร์ end connection parameter (ENDCNN) บนคำสั่ง SIGNOFF
4. ในการตัดการเชื่อมต่อทั้งหมดจากอุปกรณ์ ให้ใช้คำสั่ง Disconnect Job (DSCJOB)

ในการใช้ System i Navigator และหน้าต่าง Confirm Delete/End ให้ใช้ข้อแนะนำต่อไปนี้:

1. ขยาย My Connections → End point system → Work Management → Active Jobs.

2. คลิกขวางานที่คุณต้องการจบ และคลิก **Delete/End**. หน้าต่าง Confirm Delete/End จะปรากฏขึ้น โดยคุณสามารถระบุวิธีและเวลาที่คุณต้องการให้งานแบบโต้ตอบจบได้ในหน้าต่างนี้

หมายเหตุ: ในการจบงานแบบโต้ตอบที่เกี่ยวกับเวิร์กสเตชัน หรืองานทั้งหมดที่เกี่ยวข้องกลุ่ม (หากงานเป็นงานกลุ่ม) ให้ใช้ค่าของฟิลด์ **Action for related interactive jobs** ที่ End for group jobs หรือ End all (เท่ากับพารามิเตอร์ ADLINTJOBS บนคำสั่ง ENDJOB) อย่างใดอย่างหนึ่ง

และคุณยังสามารถร้องขอให้ระบบย่อย ส่งข้อความไปที่ message queue เมื่องานแบบโต้ตอบ inactive ตามช่วงเวลาที่เราบุ จากนั้นคุณ หรือโปรแกรมที่มอนิเตอร์ message queue นั้น สามารถจบหรือตัดการเชื่อมต่องานได้

หลักการที่เกี่ยวข้อง

“การตัดการเชื่อมต่องานแบบโต้ตอบ” ในหน้า 46

เมื่อใช้คำสั่ง Disconnect Job (DSCJOB) งานจะถูกตัดการเชื่อมต่อ และจอแสดงผล sign-on จะแสดงขึ้นอีกครั้งหนึ่ง ถ้าต้องการ เชื่อมต่อกับงานอีกครั้ง ให้ sign on ที่อุปกรณ์เดิมซึ่งคุณตัดการเชื่อมต่องานแบบโต้ตอบอื่นอาจถูกเริ่มบนอุปกรณ์ภายใต้ชื่อผู้ใช้ที่แตกต่างอื่น

การตัดการเชื่อมต่องานทั้งหมดจากอุปกรณ์:

คำสั่ง Disconnect Job (DSCJOB) อนุญาตให้ผู้ใช้แบบโต้ตอบตัดการเชื่อมต่องานแบบโต้ตอบทั้งหมดที่เวิร์กสเตชัน และกลับไปจอแสดงผล sign-on โลกที่สลับจะถูกยกเลิกเฉพาะถ้ามีการระบุใน workstation device description ของเวิร์กสเตชันนี้ และถ้า ไม่มีเวิร์กสเตชันอื่นบนโลกนี้แอ็คทีฟ ถ้างานถูกตัดการเชื่อมต่อเมื่อใช้งานถึง ช่วงตัดการเชื่อมต่อในค่ากำหนดของระบบ Time-out interval for disconnected jobs (QDSCJOBITV) งานจะสิ้นสุดลงและไม่มีกรรมบันทึกรการใช้งาน กับสฟูลเอาต์พุตของงาน

ข้อจำกัด:

1. งานที่จะตัดการเชื่อมต่อต้องเป็นงานแบบโต้ตอบ
2. งานที่ถูกพักไว้ไม่สามารถตัดการเชื่อมต่อได้
3. งานแบบ pass-through ไม่สามารถตัดการเชื่อมต่อ ยกเว้นว่าผู้ใช้ได้ใช้ฟังก์ชัน การร้องขอระบบเพื่อกลับไประบบซอร์สจากระบบเป้าหมายแบบ 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)

หลักการที่เกี่ยวข้อง

“การตัดการเชื่อมต่องานแบบโต้ตอบ” ในหน้า 46

เมื่อใช้คำสั่ง Disconnect Job (DSCJOB) งานจะถูกตัดการเชื่อมต่อ และจอแสดงผล sign-on จะแสดงขึ้นอีกครั้งหนึ่ง ถ้าต้องการ เชื่อมต่อกับงานอีกครั้ง ให้ sign on ที่อุปกรณ์เดิมซึ่งคุณตัดการเชื่อมต่องานแบบโต้ตอบอื่นอาจถูกเริ่มบนอุปกรณ์ภายใต้ชื่อผู้ใช้ที่แตกต่างอื่น

ข้อควรพิจารณาการตัดการเชื่อมต่องาน:

มีหลายปัจจัยที่คุณต้องพิจารณาในทุกครั้งที่คุณตัดการเชื่อมต่องาน

- อีอ็อปชันบนเมนู System Request ช่วยให้คุณสามารถตัดการเชื่อมต่องานแบบโต้ตอบ ซึ่งทำให้จอแสดงผล sign-on ปรากฏขึ้น อีอ็อปชันจะเรียกคำสั่ง Disconnect Job DSCJOB
 - เมื่อเชื่อมต่องานอีกครั้ง ค่าที่ระบุบนจอ sign-on 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 ที่ผู้ใช้ระบุชื่อโดยใช้วิธีใดวิธีหนึ่งดังนี้:
 - การใช้ Network Station กับพารามิเตอร์ DISPLAY NAME
 - การใช้ System i Client Access support พร้อมฟังก์ชัน workstation ID
 - การใช้ TCP/IP TELNET Device Initialization exit point เพื่อระบุ ชื่อเวิร์กสเตชัน
 - ถ้าไม่สามารถตัดการเชื่อมต่องานเนื่องจากเหตุผลใดๆ งานจะถูกจบแทน
 - งานที่ตัดการเชื่อมต่อทั้งหมดในระบบย่อยจะจบลง เมื่อระบบย่อยสิ้นสุดลง ถ้าระบบย่อยกำลังสิ้นสุด จะไม่สามารถใช้คำสั่ง DSCJOB ในงานใดๆในระบบย่อย
 - สามารถใช้ค่ากำหนดของระบบ Disconnect Job Interval (QDSCJOBITV) เพื่อระบุช่วงเวลาที่สามารถตัดการเชื่อมต่องาน ได้ ถ้าถึงช่วงเวลานั้นแล้ว งานที่ตัดการเชื่อมต่อจะจบลง
 - งานที่ตัดการเชื่อมต่อที่ไม่เกินกว่าค่ากำหนดของระบบ QDSCJOBITV จะจบ ลง เมื่อระบบย่อยจบลงหรือเมื่อเกิดการ IPL
- หลักการที่เกี่ยวข้อง**
- “การตัดการเชื่อมต่องานแบบโต้ตอบ” ในหน้า 46
- เมื่อใช้คำสั่ง Disconnect Job (DSCJOB) งานจะถูกตัดการเชื่อมต่อ และจอแสดงผล sign-on จะแสดงขึ้นอีกครั้งหนึ่ง ถ้า ต้องการ เชื่อมต่อกับงานอีกครั้ง ให้ sign on ที่อุปกรณ์เดิมซึ่งคุณตัดการเชื่อมต่อ งานแบบโต้ตอบอื่นอาจถูกเริ่มบนอุปกรณ์ ภายใต้อีอ็อปชันที่แตกต่างอื่น

การหลีกเลี่ยงฟังก์ชันที่ใช้เวลารันนานจากเวิร์กสเตชัน:

เพื่อหลีกเลี่ยงฟังก์ชันที่ใช้เวลารันนาน (เช่น บันทึก/คืนสภาพ) จาก เวิร์กสเตชันโดยไม่ทำให้ระบบชะงัก ผู้ควบคุมระบบ สามารถส่งงานไปที่ คิวงาน

คำอธิบายระบบย่อย QSYS/QBATCH หรือ QSYS/QBASE ซึ่งจัดให้โดย IBM มีคิวงาน QSYS/QBATCH ที่สามารถใช้ สำหรับวัตถุประสงค์นี้ได้ หากคุณสร้าง ระบบย่อยของคุณเอง คุณควรอ้างอิงที่คิวงานสำหรับระบบย่อยนั้น ผู้ควบคุมระบบ สามารถส่งคำสั่งจากเมนูผู้ควบคุมระบบ

ข้างล่างนี้เป็นตัวอย่างของการส่งคำสั่งที่ใช้เวลารันนาน:

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

หลักการที่เกี่ยวข้อง

“วิธีการเริ่มงานแบบโต้ตอบ” ในหน้า 45

เมื่อผู้ใช้ signs on ที่ระบบ ระบบย่อยจะรวบรวมข้อมูลจาก อีอ็อปเจอร์ระบบต่างๆ ก่อนที่งานแบบโต้ตอบจะพร้อมใช้งาน

การจัดการงานแบบ prestart

คุณสามารถใช้งานแบบ prestart เพื่อลดจำนวนเวลาที่ต้องใช้ในการจัดการคำขอการเริ่มโปรแกรม ซึ่งเป็นภารกิจทั่วไปที่เกี่ยวข้องกับงานแบบ prestart ที่คุณสามารถดำเนินการได้

หลักการที่เกี่ยวข้อง

“งานแบบ prestart และ job accounting” ในหน้า 101

ถ้าระบบของคุณใช้ job accounting โปรแกรมงานแบบ prestart ควรจะรันคำสั่ง Change Prestart Job (CHGPJ) ด้วยค่าคำขอการเริ่มโปรแกรมสำหรับพารามิเตอร์ accounting code parameter (CHGPJ ACGCDE(*PGMSTRRQS)) โดยทันทีหลังจากที่คำขอการเริ่มโปรแกรมแนบกับงานแบบ prestart

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 ให้ใช้ไคต์ delay loop แล้วลองใหม่ถ้าคำสั่ง STRPJ ล้มเหลว

จำนวนของ งานแบบ prestart ที่สามารถแอนด์ที่ไฟได้ในเวลาเดียวกันถูกจำกัดโดยแอ็ททริบิวต์ MAXJOBS บน prestart job entry และโดยแอ็ททริบิวต์ MAXJOBS สำหรับระบบย่อย แอ็ททริบิวต์ MAXACT บน communications entry ควบคุม จำนวนของคำขอการเริ่มโปรแกรมที่สามารถให้บริการได้ผ่านทาง communications entry ในเวลาเดียวกัน

หมายเหตุ: ถ้าคุณระบุ *NO บนแอ็ททริบิวต์ 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 ต้องแอนด์ที่ไฟในขณะที่ออก คำสั่งนี้ จำนวนของงานที่เริ่มต้นคือจำนวนที่ระบุในค่า INLJOBS ของ prestart job entry PJPGM ระบบย่อย จะเริ่มโปรแกรม PJPGM ในไลบรารี PJLIB

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

หลักการที่เกี่ยวข้อง

“งานแบบ prestart” ในหน้า 51

งานแบบ prestart คืองานแบ็ตซ์ที่เริ่มต้นก่อนที่จะได้รับคำขอ งานแบบ prestart ถูก เริ่มต้นก่อนงานชนิดอื่นๆ ในระบบย่อย งานแบบ prestart แตกต่างจากงานอื่นๆ เนื่องจาก การใช้ prestart job entry (ส่วนของ subsystem description) ในการกำหนด โปรแกรม, คลาส และพูลของหน่วยความจำที่จะใช้เมื่องานถูกเริ่มต้น

ข้อมูลที่เกี่ยวข้อง

รายงานจากประสบการณ์: การปรับ prestart job entries

การจัดคิว หรือการปฏิเสธคำขอการเริ่มโปรแกรม:

ถ้าคำขอการเริ่มโปรแกรมมาถึงเมื่อจำนวนปัจจุบันของงานแบบ prestart น้อยกว่าจำนวนที่ระบุในแอ็ททริบิวต์ MAXJOBS บน prestart job entry และไม่มีงานแบบ prestart พร้อมใช้ในการจัดการคำขอการเริ่ม โปรแกรม คุณมีทางเลือกที่จะปฏิเสธหรือจัดคิวคำขอใหม่

เพื่อปฏิเสธหรือจัดคิวคำขอการเริ่มโปรแกรม ให้ใช้แอ็ททริบิวต์ 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 ระบุว่างานแบบ 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 SBSD(QGPL/PJSBS) PGM(QGPL/PGM1) INLJOBS(15)
THRESHOLD(5) ADLJOBS(10) WAIT(*NO)
```

การปรับ prestart job entry:

คุณควรมีงาน prestart เพียงพอที่จะให้ระบบย่อยเริ่มเพื่อจัดการงาน เมื่องานมาถึงแทนที่จะรองานใหม่ให้เริ่ม คำแนะนำเหล่านี้แสดงถึงวิธีปรับงานแบบ prestart เพื่อประสิทธิภาพที่ดีที่สุด

การตั้งจำนวนงานแบบ prestart:

ในขณะที่ระบบจัดการเวิร์กโหลดปกติ และข้อมูลเกี่ยวกับเวิร์กโหลดที่มีอยู่ให้ปฏิบัติตามขั้นตอนต่อไปนี้:

1. ให้ใช้คำสั่ง Work with Subsystems (WRKSBS) เพื่อรับรายการระบบย่อยที่แอ็คทีฟทั้งหมด สำหรับแต่ละระบบย่อยในรายการระบบย่อยที่แอ็คทีฟ ให้ใช้อ็อปชัน 5 เพื่อแสดง subsystem description
บนพาเนล Display Subsystem Description ให้ใช้อ็อปชัน 10 เพื่อแสดง prestart job entry หากไม่มี prestart job entry สำหรับ subsystem description นั้น ให้ดำเนินการต่อด้วยระบบย่อยถัดไปในรายการ WRKSBS
2. บนพาเนล Display Prestart Job Entries ให้ใช้อ็อปชัน 5 เพื่อแสดงรายละเอียดสำหรับ prestart job entry บันทึกข้อความเกี่ยวกับการตั้งค่าปัจจุบันสำหรับจำนวนเริ่มต้นของงาน, Threshold และจำนวนเพิ่มเติมของงาน
3. สำหรับแต่ละ prestart job entry ใน subsystem description ให้ป้อนคำสั่ง Display Active Prestart Jobs (DSPACTPJ) ตัวอย่างเช่น:

```
DSPACTPJ SBS(SUBSYSTEM) PGM(PJPGMLIB/PJPROGRAM)
```

หากไม่อนุญาตให้ใช้คำสั่ง DSPACTPJ ในขณะนั้น prestart job entry จะไม่แอ็คทีฟ และไม่จำเป็นต้องเปลี่ยนแปลงดำเนินการต่อด้วย prestart job entry ต่อไป หรือ subsystem description

4. ให้ใช้ข้อมูล DSPACTPJ เพื่อขอประมาณการเวิร์กโหลดของคุณ คำสั่ง DSPACTPJ จะจัดทำผลการแสดงผลที่มีลักษณะเช่นนี้:

```

-----
                                Display Active Prestart Jobs                                SYSTEM
                                                                                   08/06/03 07:35:00
Subsystem . . . . . : SUBSYSTEM      Reset date . . . . . : 08/06/03
Program . . . . . : PJPROGRAM       Reset time . . . . . : 07:23:03
Library . . . . . : PJPGMLIB       Elapsed time . . . . . : 0000:11:57

Prestart jobs:
Current number . . . . . : 122
Average number . . . . . : 21.4
Peak number . . . . . : 122

Prestart jobs in use:
Current number . . . . . : 120
Average number . . . . . : 17.7
Peak number . . . . . : 120

More...

Press Enter to continue.

F3=Exit  F5=Refresh  F12=Cancel  F13=Reset statistics
-----

```

```

-----
                                Display Active Prestart Jobs                                SYSTEM
                                                                                   08/06/03 07:35:00
Subsystem . . . . . : SUBSYSTEM      Reset date . . . . . : 08/06/03
Program . . . . . : PJPROGRAM       Reset time . . . . . : 07:23:03
Library . . . . . : PJPGMLIB       Elapsed time . . . . . : 0000:11:57

Program start requests:
Current number waiting . . . . . : 0
Average number waiting . . . . . : .0
Peak number waiting . . . . . : 1
Average wait time . . . . . : 00:00:00.0
Number accepted . . . . . : 120
Number rejected . . . . . : 0

Press Enter to continue.

F3=Exit  F5=Refresh  F12=Cancel  F13=Reset statistics
-----

```

คันทางานแบบ prestart ในหัวข้อที่ใช้อยู่ และค่าสำหรับจำนวนสูงสุด ในตัวอย่างนี้ ค่า คือ 120 ตัวเลขนี้เป็นประมาณการเวริกโหลตสูงสุดของคุณ บันทึกค่านี้เพื่อใช้ในขั้นตอนต่อไป

ค้นหาหัวข้อการขอเริ่มใช้โปรแกรม และค่าสำหรับจำนวนสูงสุดที่รออยู่ คุณอาจจำเป็นต้องเลื่อนหน้าลงเพื่อดูฟิลด์นี้ ในตัวอย่างนี้ ค่า คือ 1 ตัวเลขนี้บอกคุณว่าจะจัดการกับงานใหม่ที่มาได้ดีเพียงใด บันทึกค่านี้เพื่อใช้ในขั้นตอนต่อไป

5. หาก DSPACTPJ แสดงเป็นศูนย์ (0) สำหรับจำนวนสูงสุดของงานแบบ prestart ที่ใช้อยู่ เวิร์กโหลตของคุณจะไม่ใช้ prestart job entry และดังนั้นจึงไม่จำเป็นต้องเปลี่ยนแปลง ดำเนินการต่อด้วย prestart job entry ต่อไป หรือ subsystem description ต่อไป
6. เลือกค่าสำหรับพารามิเตอร์ THRESHOLD เมื่อพูลของงานที่มีอยู่ลดลงต่ำกว่าจำนวนนี้ งานใหม่จำนวนมากขึ้นจะเริ่มขึ้น การเริ่มงานใช้เวลานาน ในขณะที่นั้น อาจมีค่าของงานเพิ่มขึ้น เซ็ต THRESHOLD เป็นค่าอย่างน้อยเท่ากับหนึ่งบวกกับจำนวนคำขอที่สามารถมาถึงในขณะที่งานใหม่กำลังเริ่ม

ในตัวอย่างนี้ ค่าที่ถูกเลือกเป็น 10 ตัวเลขนี้เป็นประมาณการมาถึงของค่าของงาน ซึ่งเป็นการคาดเดาตามจำนวนสูงสุดของงานที่ใช้อยู่ซึ่งไม่ใช่การวิเคราะห์ที่แม่นยำเกี่ยวกับการวัดที่ยากจะทำได้

ให้ดูบันทึกที่คุณจัดไว้ในขั้นตอนก่อนหน้า ถ้าการตั้งค่าปัจจุบันสำหรับ THRESHOLD สูงเพียงพอจำนวนสูงสุดที่รออยู่คือศูนย์ ถ้าจำนวนสูงสุดที่รออยู่ไม่ใช่ศูนย์ ให้เพิ่มจำนวนนี้กับค่า THRESHOLD ปัจจุบัน และเปรียบเทียบผลกับค่าประมาณการตามการมาถึง ใช้ค่าที่มากกว่า ข้อมูล DSPACTPJ ตัวอย่างแสดงค่าเป็น 1 ซึ่งหมายความว่าค่าปัจจุบันสำหรับ THRESHOLD ต่ำเกินไป การตั้งค่าปัจจุบันบวกหนึ่งน้อยกว่าค่าประมาณที่ 10 สำหรับตัวอย่างนี้เราใช้ค่า 10

7. ให้เลือกค่าสำหรับพารามิเตอร์ initial number of jobs (INLJOBS) INLJOBS จะระบุจำนวนงานที่เริ่มขึ้นเมื่อระบบย่อยเริ่มต้น และ INLJOBS เป็นส่วนหนึ่งของระบบย่อยที่ใช้เพื่อตัดสินใจว่างานแบบ prestart มากมายที่รอทำงานอยู่หรือไม่ ให้ดูบันทึกที่คุณจัดไว้ในขั้นตอนก่อนหน้า ให้ใช้จำนวนสูงสุดของงานแบบ prestart job ที่ใช้อยู่เป็นประมาณการเวิร์กโหนดสูงสุด ให้เพิ่มค่าสำหรับ THRESHOLD และใช้ผลเป็นค่าใหม่สำหรับ INLJOBS ข้อมูล DSPACTPJ แสดงจำนวนงานแบบ prestart ที่ใช้อยู่สูงสุดที่ 120 และเราได้เลือก THRESHOLD ที่ 10 ดังนั้นค่าใหม่ที่เลือกสำหรับ INLJOBS คือ 130
8. ให้เลือกค่าสำหรับพารามิเตอร์ additional number of jobs (ADLJOBS) ADLJOBS ระบุจำนวนงานแบบ prestart เพิ่มเติมที่เริ่ม เมื่อจำนวนงานแบบ prestart ที่มีอยู่ลดลงต่ำกว่าค่าที่ระบุไว้บนพารามิเตอร์ Threshold (THRESHOLD) เมื่อ INLJOBS และ THRESHOLD สูงพอที่จะหลีกเลี่ยงการทำให้เกิดค่าขอที่รออยู่ ADLJOBS อาจค่อนข้างต่ำได้ ถ้า INLJOBS ต่ำกว่าเวิร์กโหนดสูงสุดมาก ADLJOBS อาจจำเป็นต้องเท่ากับ THRESHOLD ในตัวอย่างนี้ ค่าที่เลือก คือ 5 พยายามหลีกเลี่ยงจำนวนสูง ถ้าคุณระบุค่าสำหรับ ADLJOBS ระบบย่อยจะเริ่มจำนวนสูงกับงานทั้งหมดทันที ซึ่งทำให้เกิดผลกระทบในทางลบต่อประสิทธิภาพของระบบ และทำให้การจัดการระบบย่อยของงานอื่นช้าลง
9. เปรียบเทียบค่าที่เลือกใหม่กับค่าที่ตั้งไว้ใน prestart job entry เพื่อให้มั่นใจว่ามีงานแบบ prestart job เพียงพอให้ใช้จำนวนสูงสุดสำหรับแต่ละพารามิเตอร์ เปลี่ยนค่าที่ตั้งไว้โดยใช้คำสั่ง Change Prestart Job Entry (CHGPJE)
CHGPJE SBSDB(SBSLIB/SUBSYSTEM) PGM(PJPGMLIB/PJPROGRAM)
INLJOBS(130) THRESHOLD(10) ADLJOBS(5)
10. ดำเนินการต่อด้วย prestart job entry ต่อไป หรือ subsystem description ต่อไป

รายละเอียด

รายละเอียดเพิ่มเติมบางประการอาจช่วยให้คุณทำการตัดสินใจได้ดี เมื่อปฏิบัติตามกระบวนการต่อไปนี้

- ถ้าค่า THRESHOLD น้อยเกินไป งานจะรอจนใหม่เริ่ม ในบางกรณี ข้อผิดพลาดเกิดขึ้นได้ เพราะสิ้นสุดระยะเวลาของคำขอ

ให้พิจารณาตัวอย่างที่ THRESHOLD เป็น 2 และมีเพียงสองงานที่รอดำเนินการอยู่เท่านั้น เมื่อคำขอดำเนินการต่อไปมาถึง คำขอนั้นจะถูกส่งให้งานที่รออยู่งานหนึ่ง และงานเพิ่มเติมจะถูกเริ่มขึ้น ในตัวอย่างนี้ คำขออีกสองข้อมาถึงก่อนงานใหม่พร้อม คำขอแรกถูกจัดการโดยงานที่รออยู่ คำขอที่สองรอหนึ่งในงานใหม่พร้อม สำหรับตัวอย่างเวิร์กโหนด THRESHOLD ควรกำหนดไว้ที่อย่างน้อย 3: หนึ่งเพื่อกระตุ้นการสร้างงานเพิ่มขึ้นบวกสองสำหรับจำนวนคำขอที่มาถึงในขณะที่งานเริ่มขึ้น

- เนื่องจากระบบย่อยเริ่มงานเมื่อจำเป็น ระบบย่อยยังจบงานเมื่อไม่จำเป็นด้วย ลักษณะนี้เกิดขึ้นกับ prestart job entry ที่ระบุ maximum number of uses (MAXUSE) มากกว่าหนึ่ง ค่าสำหรับพารามิเตอร์ INLJOBS จะบอกระบบย่อยถึงจำนวนงานที่จำเป็น คุณจำเป็นต้องเซต INLJOBS ให้ถูกต้องเพื่อป้องกันระบบย่อยจากการจบงานมากเกินไป

หากค่า INLJOBS ต่ำเกินไป ระบบย่อยจะเริ่มงานตามระยะ เนื่องจากมีจำนวนงานน้อยเกินไป และจบงานลงเพราะมีจำนวนมากเกินไป นอกจากนั้น ระบบทำให้เกิดต้นทุนการเริ่มงานใหม่ในเวลาทีระบบยุ่งที่สุด

- ในตัวอย่างเอาต์พุตจากคำสั่ง DSPACTPJ จำนวนสูงสุดของงานแบบ prestart ที่ใช้อยู่คือ 120 ในขณะที่ค่าเฉลี่ยของจำนวนงานแบบ prestart ที่ใช้อยู่คือ 17.7 ตัวเลขนี้ไม่ใช่ตัวเลขสูงสุด แต่เป็นค่าเฉลี่ยต่ำ โดยค่าดีฟอลต์ DSPACTPJ แสดงสิ่งที่เกิดขึ้นนับจากระบบย่อยเริ่ม ค่าเฉลี่ยรวมช่วงระยะเวลาเวิร์กโหลดเป็นศูนย์
เมื่อใดก็ตามที่คุณใช้ F13 ในการรีเซ็ตสถิติ และเมื่อคุณควบคุมรอบระยะเวลาตัวอย่างอย่างรอบคอบ ค่าเฉลี่ยของจำนวนงานแบบ prestart ที่ใช้อยู่มีแนวโน้มที่จะต่ำกว่าจำนวนที่คุณควรปรับให้เป็น เวิร์กโหลดหนึ่งอาจมีค่าเฉลี่ยจำนวนงานใดระหว่าง 40 และ 60 และยังมีจำนวนสูงสุดมากมายระหว่าง 100 และ 120 งาน
เมื่อ INLJOBS ถูกเซตที่เวิร์กโหลดสูงสุดประมาณการบวก THRESHOLD ระบบย่อยไม่จำเป็นต้องเริ่มงานเพิ่ม เว้นแต่ว่าเวิร์กโหลดจริงเกินกว่าจำนวนเวิร์กโหลดสูงสุดที่ประมาณการ ถ้าเวิร์กโหลดของคุณมีจำนวนสูงสุดที่สูง และไม่บ่อยครั้ง คุณอาจต้องการเซต INLJOBS เป็นจำนวนต่ำกว่า
- กระบวนการที่ให้นับชั่วโมงอยู่บนสมมติฐานว่าโหลดสูงสุดในวันปกติเป็นโหลดสูงสุดปกติ ถ้าคุณรวบรวมข้อมูลได้เพิ่มเติม คุณอาจจัดทำประมาณการเวิร์กโหลดของคุณที่แม่นยำขึ้นได้
คุณสามารถใช้ List Job (QUSLJOB) API หรือ Open List of Jobs (QGYOLJOB) API เพื่อทดลองเวิร์กโหลดของคุณเป็นระยะได้ สำหรับเวิร์กโหลดบางงาน เวิร์กโหลดจะช่วยทำกราฟผลลัพธ์ให้คุณไม่จำเป็นต้องทำการคาดการณ์จำนวนงานแบบ prestart อย่างสมบูรณ์ คุณเพียงแค่ทำให้ใกล้เคียงเพียงพอที่จะหลีกเลี่ยงความล่าช้า และเวลาที่น้อยลง
- ถ้า THRESHOLD และ INLJOBS สูงเกินไป จะมีงานที่แอ็คทีฟในระบบย่อยที่ไม่จำเป็น การเริ่ม และจบงานพิเศษใช้เวลามากขึ้นเมื่อเริ่ม หรือจบระบบย่อย หรือเมื่อเริ่ม หรือจบ prestart job entry
การใช้ค่าที่สูงกว่าที่จำเป็นเล็กน้อยดีกว่าการใช้ค่าที่ต่ำกว่าที่จำเป็น การมีงานพิเศษจำนวนน้อยเกินไปไม่ใช่ปัญหา เนื่องจากงานเหล่านี้รอการดำเนินการอยู่และไม่แย่งหน่วยความจำ หรือโปรเซสเซอร์
- เนื่องจากงานแบบ prestart ใช้เป็นครั้งแรกในอุปกรณ์การสื่อสาร คำขอดำเนินการจะเรียกเป็นคำขอเริ่มโปรแกรม และงานแบบ prestart จะแสดงสถานะ PSRW (รอคำขอโปรแกรมเริ่ม) เมื่องานรอการดำเนินการอยู่

การเปลี่ยนแอ็คทีวิตีงานสำหรับงานแบบ prestart:

คิวข้อความงานขนาดใหญ่ใช้หน่วยความจำสูง ทำให้บันทึกการใช้งานมีขนาดใหญ่ที่ใช้หน่วยความจำมากด้วย และทำให้เกิดปัญหาประสิทธิภาพ IPL เมื่อคิวข้อความงานจำเป็นต้องกู้คืน หรือล้างระหว่าง IPL ตัวอย่างนี้แสดงวิธีเปลี่ยน job message queue full action (JOBMSGQFL) และค่า job message queue maximum size (JOBMSGQMX) สำหรับงานแบบ prestart

หมายเหตุ: คำอธิบายงาน QDFTSVR เริ่มมีในรีลีส V5R3M0 เพื่อทำหน้าที่บางอย่างนี้ให้คุณ

ในการจำกัดคิวข้อความงานสำหรับงานแบบ prestart โดยไม่มีผลกระทบต่องานอื่น ให้ปฏิบัติขั้นตอนเหล่านี้:

1. หางานแบบ prestart jobs ที่คุณต้องการให้เปลี่ยน และกำหนดรายละเอียดของงานที่ใช้โดย prestart job entry (ในการทำเช่นนี้ ให้ใช้คำสั่ง Display Subsystem Description (DSPSBSD))
2. กำหนดรายละเอียดของงานที่ใช้โดย prestart job entry รายการเดียว (ในกรณีที่คุณสามารถแก้ไขรายละเอียดของงานนั้นได้) หรือใช้โดยการอ้างอิงหลายประเภท เช่น โปรไฟล์ผู้ใช้, prestart job entries, SBSBD entries อื่นๆ ฯลฯ (คุณสามารถสร้างรายละเอียดของงานอีกอันหนึ่งสำหรับ "don't know" ได้เสมอ แต่ถ้าคุณทราบว่าการเปลี่ยนแปลงรายละเอียดของงานที่มีอยู่นั้นกระทบเฉพาะงานที่คุณต้องการให้เปลี่ยนแปลงเท่านั้น คุณควรแก้ไขรายละเอียดของงานเฉพาะนั้น)
3. สร้างรายละเอียดของงานใหม่ที่ใช้โดย prestart job entries ที่คุณต้องการให้เปลี่ยนแปลง คุณสามารถใช้คำสั่ง Create Job Description (CRTJOB) command แต่ในตัวอย่างนี้ เราได้ทำสำเนารายละเอียดของงานที่ใช้อยู่ในขณะนี้ไว้

หมายเหตุ: ถ้าคุณมีรายละเอียดของงาน JOB(*USRPRF) คุณสามารถใช้คำสั่ง Display User Profile (DSPUSRPRF) เพื่อกำหนดรายละเอียดของงานที่ใช้อยู่ในขณะนี้ Configuration ที่มีดีฟอลต์ไว้ใช้รายละเอียดของงาน QDFTJOB หรือ QDFTSVR

เพื่อหลีกเลี่ยงความสับสนกับอ็อบเจกต์ที่จัดให้โดย IBM ให้หลีกเลี่ยงชื่อที่เริ่มด้วยตัวอักษร 'Q' ตัวอย่างนี้ใช้ชื่อ PJJOBID เป็นชื่อรายละเอียดของงานสำหรับ prestart job entries ให้ใช้คำสั่ง Create Duplicate Object (CRTDUPOBJ) เพื่อทำสำเนารายละเอียดของงานที่ใช้อยู่ในขณะนี้โดยโปรแกรมเมอร์ผู้ใช้ QUSER

```
CRTDUPOBJ OBJ(QDFTSVR) FROMLIB(QGPL) OBJTYPE(*JOBID)
TOLIB(QGPL) NEWOBJ(PJJOBID)
```

- จับคู่ความเป็นเจ้าของอ็อบเจกต์ และสิทธิของรายละเอียดของงานที่คุณสำเนา เนื่องจาก QDFTSVR และ QDFTJOBID มี QPGMR เป็นเจ้าของ ตัวอย่าง (ด้านล่าง) แสดงวิธีเปลี่ยนรายละเอียดของงานที่สร้างขึ้นใหม่ที่ QPGMR เป็นเจ้าของ ให้ใช้คำสั่ง Change Object Owner (CHGOBJOWN) และคำสั่ง Grant Object Authority (GRTOBJAUT) เพื่อครอบครองอ็อบเจกต์ และสิทธิพัลลิกที่กำหนดไว้อย่างถูกต้อง คุณสามารถหาเจ้าของ และสิทธิโดยการให้คำสั่ง Display Object Authority (DSPOBJAUT)

```
CHGOBJOWN OBJ(QGPL/PJJOBID) OBJTYPE(*JOBID) NEWOWN(QPGMR)
```

```
GRTOBJAUT OBJ(QGPL/PJJOBID) OBJTYPE(*JOBID) USER(*PUBLIC) AUT(*USE)
```

- ให้ใช้คำสั่ง Change Job Description (CHGJOBID) เพื่อปรับแก้ทริบิวต์งานตามความต้องการในตัวอย่างนี้เราใช้ค่า 8 เมกะไบต์สำหรับขนาดสูงสุดของคิวข้อความงาน ค่าอื่นๆสามารถใช้ได้ด้วย ตราบเท่าที่ขีดจำกัดน้อยกว่า 64 เมกะไบต์มาก

```
CHGJOBID JOBID(QGPL/PJJOBID) JOBMSGQMX(8) JOBMSGQFL(*WRAP)
TEXT('แอดทริบิวต์งานสำหรับ prestart job entries')
```

- พิจารณาผ่าน prestart job entries ทั้งหมดที่แอดที่ฟบนระบบของคุณ ให้ใช้คำสั่ง Work with Subsystems (WRKSBS) เพื่อรับรายการระบย่อยที่แอดที่ฟทั้งหมด ให้ใช้อ็อบชัน 5 เพื่อแสดง subsystem description ให้ใช้อ็อบชัน 10 เพื่อแสดง prestart job entry และใช้อ็อบชัน 5 เพื่อแสดงรายละเอียดสำหรับ prestart job entry

หาก prestart job entry ระบุ USER(QUSER) และ JOBID(*USRPRF) ให้ใช้คำสั่ง Change Prestart Job Entry (CHGPJE) เพื่อระบุรายละเอียดของงานใหม่

```
CHGPJE SBSID(SBSLIB/SUBSYSTEM) PGM(PJPGMLIB/PJPROGRAM)
JOBID(QGPL/PJJOBID)
```

หาก prestart job entry ระบุรายละเอียดของงาน ให้ใช้คำสั่ง Change Job Description (CHGJOBID) เพื่อเปลี่ยนค่า JOBMSGQMX และ JOBMSGQFL ในรายละเอียดของงานนั้น

```
CHGJOBID JOBID(JOBDLIB/JOBIDNAME) JOBMSGQMX(8) JOBMSGQFL(*WRAP)
```

รายละเอียด

รายละเอียดของงาน QDFTJOBID ที่ใช้โดย prestart job entry หลากหลาย และใช้สถานที่อื่นมากมายในระบบ ตัวอย่างนี้สร้างรายละเอียดของงานใหม่หนึ่งเดียวที่ชื่อว่า PJJOBID รายละเอียดของงานใหม่ที่ใช้โดย prestart job entry หลากหลาย แต่ไม่ถูกใช้ที่อื่น ในการใช้คำสั่งต่างๆสำหรับ prestart job entry ต่างๆ ให้ใช้รายละเอียดของงานแตกต่างกันสำหรับแต่ละ entry Prestart job entry มีรายละเอียดของงานแบบเฉพาะแล้ว

แอดทริบิวต์งานบางข้อสำหรับงานแบบ prestart สามารถเปลี่ยนแปลงโดยการใช้กระบวนการ เนื่องจากไม่ได้มาจากรายละเอียดของงานที่ใช้เมื่อเริ่มทำงาน เซิร์ฟเวอร์มากมายที่ใช้ prestart job เปลี่ยนโปรแกรมเมอร์ผู้ใช้ แล้วใช้ Change Job (QWTCHGJB) API เพื่อเปลี่ยนเซตย่อยของแอดทริบิวต์งาน แอดทริบิวต์งานที่ถูกเปลี่ยนแปลงจากรายละเอียดของงานที่ใช้โดยโปรแกรมเมอร์ผู้ใช้ที่ prestart job ถูกเปลี่ยน อ้างถึงรูปแบบ JOBC0300 เกี่ยวกับ Change Job API เพื่อดูข้อมูลเพิ่มเติม

สำหรับบางแอ็ดทริบิวต์งาน รายละเอียดของงานอาจจะมูลค่าที่ได้รับจากค่าระบบ เมื่อคุณเปลี่ยนค่าระบบ การเปลี่ยนจะกระทบกับงานทั้งหมดที่ได้รับแอ็ดทริบิวต์งานจากค่าระบบ การเปลี่ยนค่าในรายละเอียดของงานกระทบกับกลุ่มงานที่รับแอ็ดทริบิวต์งานจากรายละเอียดของงานนั้นเท่านั้น

การจบงานแบบ prestart job:

คุณสามารถใช้อินเตอร์เฟสแบบอักษร เพื่อจบงานแบบ prestart ในระบบย่อยที่แอ็ดคทีฟอยู่

งานสามารถกำลังรอสำหรับคำขอหรืออาจ เชื่อมโยงกับคำขอแล้ว Spooled output files ที่เกี่ยวข้องกับงานที่กำลังจะ จบยังสามารถถูกจบหรืออนุญาตให้คงอยู่ต่อไปบนเอาต์พุตคิวได้ นอกจากนี้ยังสามารถเปลี่ยนข้อจำกัดจำนวนของข้อความที่จะบันทึกที่แต่ละบันทึกการใช้งานได้ด้วย

หมายเหตุ: ในการจบงานทั้งหมดสำหรับ prestart job entry ในระบบย่อยที่แอ็ดคทีฟ ให้ใช้คำสั่ง End Prestart Job (ENDPJ) หากคุณประสงค์จะจบงานแบบ prestart เฉพาะที่มีปัญหาอยู่ ให้ใช้คำสั่ง End Job (ENDJOB) กับงานแบบ prestart นั้น

คำสั่ง : End Prestart Job (ENDPJ)

ตัวอย่าง: คำสั่งนี้จะจบงานทั้งหมดที่เชื่อมโยงกับ prestart job entry PJPGM ในระบบย่อย SBS1 โดยทันที สพูลเอาต์พุตที่สร้างขึ้นโดยงานแบบ prestart เหล่านี้จะถูกลบออกและมีการบันทึกบันทึกการใช้งาน

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

ตัวอย่าง: คำสั่งนี้จะจบงานทั้งหมดที่เชื่อมโยงกับ prestart job entry PJPGM2 ในระบบย่อย SBS2 สพูลเอาต์พุตสำหรับงานแบบ prestart เหล่านี้จะถูกบันทึกสำหรับการประมวลผลปกติโดย spooling writer งานต่างๆ มีเวลา 50 วินาทีในการปฏิบัติที่การล้างข้อมูล หลังเวลานี้งานจะถูกจบลงในทันที

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

หลักการที่เกี่ยวข้อง

“งานแบบ prestart” ในหน้า 51

งานแบบ prestart คืองานแบ็คที่เริ่มต้นก่อนที่จะได้รับคำขอ งานแบบ prestart ถูก เริ่มต้นก่อนงานชนิดอื่นๆ ในระบบย่อย งานแบบ prestart แตกต่างจากงานอื่นๆ เนื่องจาก การใช้ prestart job entry (ส่วนของ subsystem description) ในการกำหนด โปรแกรม, คลาส และพูลของหน่วยความจำที่จะใช้เมื่องานถูกเริ่มต้น

ข้อมูลที่เกี่ยวข้อง

รายงานจากประสบการณ์: การปรับ prestart job entries

การจัดการคลาสอ็อบเจ็กต์งาน

คลาสอ็อบเจ็กต์มีแอ็ดทริบิวต์การรันที่ควบคุมสถานะแวดล้อมรันไทม์ ของงาน คลาสอ็อบเจ็กต์ที่ได้มาจาก IBM หรือคลาสใช้ได้กับทั้ง แอ็พพลิเคชันแบบโต้ตอบและแอ็พพลิเคชันแบบแบ็คตามปกติ คลาสที่งานใช้ ถูกระบุอยู่ใน subsystem description routing entry ที่ใช้ในการเริ่มงาน ถ้างานมีขั้นตอนการเรอต์หลายขั้นตอน คลาสที่ใช้โดย ขั้นตอนการเรอต์แต่ละขั้นตอนจะถูกระบุอยู่ใน routing entry ที่ใช้ใน การเริ่มต้นขั้นตอนการเรอต์

การสร้างคลาสอ็อบเจ็กต์:

คุณสามารถสร้างคลาสอ็อบเจกต์โดยใช้อินเตอร์เฟซแบบอักษร คลาสกำหนดแอตทริบิวต์การประมวลผลสำหรับงานที่ใช้ คลาส คลาสที่งานใช้ ถูกระบุอยู่ใน subsystem description routing entry ที่ใช้ในการเริ่มงาน ถ้างานมีขั้นตอนการเรอต์หลายชั้น ตอน คลาสที่ใช้โดย ขั้นตอนการเรอต์แต่ละชั้นตอนจะถูกระบุอยู่ใน routing entry ที่ใช้ในการเริ่มต้นขั้นตอนการเรอต์

คำสั่ง: Create Class (CRTCLS)

ตัวอย่าง: ตัวอย่างนี้จะสร้างคลาสที่มีชื่อว่า CLASS1. คลาสถูกจัดเก็บ อยู่ในไลบรารีปัจจุบันที่ระบุสำหรับงาน ข้อความผู้ใช้ 'This class for all batch jobs from Dept 4836' เป็นคำอธิบายคลาส แอตทริบิวต์ ของคลาสนี้แสดงระดับความสำคัญการรัน เป็น 60 และการแบ่งเวลาเป็น 900 มิลลิวินาที ถ้างานยังรันไม่เสร็จสิ้นหลังจากสิ้นสุดช่วงการแบ่งเวลา งานจะไม่สามารถ ถูก ย้ายออกจากแหล่งเก็บข้อมูลหลักได้ จนกว่างานจะได้รับการจัดสรรช่วงการแบ่งเวลาอื่น สมมุติว่าใช้ค่าดีฟอลต์สำหรับพารามิเตอร์อื่น

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

หลักการที่เกี่ยวข้อง

“คลาสอ็อบเจกต์” ในหน้า 35

คลาสอ็อบเจกต์มีแอตทริบิวต์การรันที่ควบคุมสถานะแวดล้อมรันใหม่ ของงาน คลาสอ็อบเจกต์ที่ได้มาจาก IBM, หรือ คลาส, ใช้ได้กับทั้ง แอ็พพลิเคชันแบบโต้ตอบและแอ็พพลิเคชันแบบแบ็ตช์ตามปกติ คลาสที่ให้มาพร้อมกับระบบ (ตาม ชื่อ) มีดังนี้:

การเปลี่ยนคลาสอ็อบเจกต์:

คุณสามารถเปลี่ยนแอตทริบิวต์ของคลาสอ็อบเจกต์ได้โดยใช้ อินเตอร์เฟซแบบอักษร สามารถเปลี่ยนแอตทริบิวต์ต่างๆ ยก เว้นแอตทริบิวต์สิทธิ์พับลิก อ้างอิงคำสั่ง Revoke Object Authority (RVKOBJAUT) และคำสั่ง Grant Object Authority (GRTOBJAUT) สำหรับข้อมูลเพิ่มเติมเกี่ยวกับการเปลี่ยนการให้สิทธิ์อ็อบเจกต์

คำสั่ง: Change Class (CHGCLS)

ตัวอย่าง: ตัวอย่างนี้จะเปลี่ยนคลาสที่ชื่อว่า CLASS1 ในไลบรารี บนรายชื่อไลบรารีของงาน ระดับความสำคัญการรันสำหรับ คลาสจะถูกเปลี่ยน เป็น 60 และการแบ่งเวลาเป็น 900 มิลลิวินาที

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

หลักการที่เกี่ยวข้อง

“คลาสอ็อบเจกต์” ในหน้า 35

คลาสอ็อบเจกต์มีแอตทริบิวต์การรันที่ควบคุมสถานะแวดล้อมรันใหม่ ของงาน คลาสอ็อบเจกต์ที่ได้มาจาก IBM, หรือ คลาส, ใช้ได้กับทั้ง แอ็พพลิเคชันแบบโต้ตอบและแอ็พพลิเคชันแบบแบ็ตช์ตามปกติ คลาสที่ให้มาพร้อมกับระบบ (ตาม ชื่อ) มีดังนี้:

การจัดการ thread

คุณสามารถดำเนินการกิจต่างๆเมื่อจัดการ thread ได้

การดู thread ที่กำลังรันภายในงานเฉพาะ:

งานแอ็คทีฟทุกงานที่กำลังรันบนระบบมี thread ที่กำลังรันอยู่ภายในงานอย่างน้อยหนึ่ง thread thread คือหน่วยงานอิสระที่รัน อยู่ภายในงานที่ใช้รีซอร์สเดียวกันกับงาน เนื่องจากงานขึ้นอยู่กับงานที่ทำ โดย thread จึงมีความสำคัญที่ต้องทราบวิธีการค้นหา threads ที่กำลังรัน ภายในงานเฉพาะ

หลักการที่เกี่ยวข้อง

“thread” ในหน้า 38

คำว่า *thread* ย่อมาจาก “thread of control” thread คือพารามิเตอร์ที่โปรแกรมใช้ในขณะรัน, ขั้นตอนปฏิบัติ และลำดับ การปฏิบัติตามขั้นตอน thread รันได้จากตำแหน่งเริ่มต้น ในลำดับที่จัดลำดับ ที่กำหนดไว้ล่วงหน้าสำหรับชุดของอินพุตที่กำหนดให้

ข้อมูลที่เกี่ยวข้อง

ตัวอย่าง: จบ thread โดยใช้ Java

API การจัดการ thread

System i Navigator:

ในการดู thread ที่รันภายใต้งานเฉพาะ ให้ปฏิบัติตามข้อแนะนำต่อไปนี้:

1. จาก System i Navigator ให้ขยาย My Connections → *your.system* → Work Management → Active Jobs.
2. คลิกชื่องานซึ่งคุณต้องการทำงาน และคลิก Details → Threads.

อินเตอร์เฟซแบบอักษร:

คำสั่ง: Work With Job (WRKJOB)

ตัวอย่าง: ตัวอย่างต่อไปนี้จะแสดงจอภาพ Work With Threads สำหรับงาน Crtpfrdta

```
WRKJOB JOB(Crtpfrdta) OPTION(*THREAD)
```

คุณสามารถทำอะไรกับ thread:

เนื่องจาก thread ช่วยงานประมวลผลมากกว่าหนึ่งการดำเนินการในแต่ละ ครั้งในขณะรัน ดังนั้นการมอนิเตอร์ threads ที่กำลังรันอยู่ภายในงานหนึ่งอาจเป็น สิ่งจำเป็น การทำเช่นนี้ช่วยให้คุณสามารถรันงานได้อย่างมีประสิทธิภาพ คุณสามารถใช้ System i Navigator เพื่อหา thread ที่คุณต้องการจัดการ

หลังจากที่คุณระบุตำแหน่ง thread แล้ว คุณสามารถคลิกขวาที่ thread และ เลือก action อย่างใดอย่างหนึ่งต่อไปนี้:

รีเซ็ตสถิติ

อนุญาตให้คุณรีเซ็ตข้อมูลรายการที่คุณกำลังดูอยู่ และตั้งเวลาที่ผ่านไปให้เป็น 00:00:00

รายละเอียด

เนื่องจากฟังก์ชันของ thread คล้ายกับฟังก์ชันของงาน, ดังนั้นฟังก์ชันทั้งสองจึงมีการดำเนินการบางอย่างเหมือนกัน รายละเอียดมีข้อมูลรายละเอียดเกี่ยวกับ thread action ต่อไปนี้:

- Call stack
- รายชื่อไลบรารี
- อ็อบเจกต์ที่ถูกล็อก
- Transaction
- สถิติประสิทธิภาพการทำงานที่ผ่านไป

พักไว้ อนุญาตให้คุณพัก thread สามารถพัก thread ได้หลายครั้ง ระบบปฏิบัติการจะคอยตรวจสอบดูจำนวนครั้งที่ thread ถูกพัก

รีลีส รีลีส thread ที่ถูกพักไว้ ต้องรีลีส thread ในแต่ละครั้งที่ถูกพักไว้เพื่อให้ thread รัน

ลบบอก/จบ

อนุญาตให้คุณจบหนึ่งหรือหลาย thread ที่เลือก

Thread Properties

แสดงแอ็ททริบิวต์ต่างๆ ของ thread

สำหรับข้อมูลรายละเอียดเกี่ยวกับ action ที่คุณสามารถทำบน thread ให้ดูคำอธิบาย System i Navigator ทางออนไลน์

ข้อมูลที่เกี่ยวข้อง

ค่ากำหนดของระบบประสิทธิภาพ: Thread affinity

ค่ากำหนดของระบบประสิทธิภาพ: ปรับเปลี่ยนรีซอร์ส thread โดยอัตโนมัติ

การดูแลสมบัติของ thread:

thread จะช่วยให้งานสามารถทำงานหลายๆ อย่างได้ในเวลาเดียวกัน หาก thread หยุดการประมวลผล thread จะหยุดงานไม่ให้เห็น

หลักการที่เกี่ยวข้อง

“thread” ในหน้า 38

คำว่า *thread* ย่อมาจาก “thread of control” thread คือพาร์ทิโปรแกรมที่ใช้ในขณะที่รัน, ขั้นตอนปฏิบัติ และลำดับ การปฏิบัติตามขั้นตอน thread รันโค้ดจากตำแหน่งเริ่มต้น ในลำดับที่จัดลำดับ ที่กำหนดไว้ล่วงหน้าสำหรับชุดของอินพุตที่กำหนดให้

ข้อมูลที่เกี่ยวข้อง

ตัวอย่าง: จบ thread โดยใช้ Java

API การจัดการ thread

System i Navigator:

ในการดูแอ็ททริบิวต์ของ thread ให้ปฏิบัติตามข้อแนะนำต่อไปนี้:

1. จาก System i Navigator ให้ขยาย **My Connections** → *your system* → **Work Management** → **Active Jobs or Server Jobs**.
2. คลิกขวางานซึ่งคุณต้องการทำงาน 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 (หมายความว่าระดับความสำคัญสูงสุดจะต่างกัน) ระดับความสำคัญการรัน thread ไม่มีทางสูงกว่าระดับความสำคัญในการรันงานที่ thread กำลังรัน

คุณยังสามารถดูค่าประสิทธิภาพที่ถูกคำนวณไว้นับตั้งแต่ที่ 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)
```

การจบ หรือการลบ thread:

initial thread ซึ่งถูกสร้างเมื่องานเริ่มทำงาน ไม่สามารถถูกลบทิ้งหรือจบการทำงานได้ อย่างไรก็ตาม บางครั้งก็เป็นเรื่องจำเป็นที่ต้องจบการทำงานของ thread รองเพื่อให้งานสามารถรันต่อไปได้ให้ระวัง 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 เหล่านั้นอาจให้ผลลัพธ์ที่ไม่พึงปรารถนา

ในการลบหรือจบ thread รอง, คุณต้องมีสิทธิ์พิเศษ service (*SERVICE) หรือสิทธิ์ Thread Control

หลักการที่เกี่ยวข้อง

“thread” ในหน้า 38

คำว่า *thread* ย่อมาจาก “thread of control” thread คือพาทที่โปรแกรมใช้ในขณะที่รัน, ขั้นตอนปฏิบัติ และลำดับ การปฏิบัติตามขั้นตอน thread รันได้จากตำแหน่งเริ่มต้น ในลำดับที่จัดลำดับ ที่กำหนดไว้ล่วงหน้าสำหรับชุดของอินพุตที่กำหนดให้

ข้อมูลที่เกี่ยวข้อง

ตัวอย่าง: จบ thread โดยใช้ Java

API การจัดการ thread

System i Navigator:

ในการลบ หรือจบ thread ให้ใช้ข้อแนะนำดังต่อไปนี้:

1. จาก System i Navigator ให้ขยาย My Connections → *your system* → 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 อีอพชัน: 4=จบ

การจัดการการจ้ดตารางเวลางาน

คุณสามารถจ้ดตารางเวลางานเพื่อรันโดยใช้ Advanced Job Scheduler โดยการใช้น้้าต่าง System i Navigator Job Properties หรือโดยการเปล่ยน job schedule entry ผ่านทางอินเทอร์เน็ตเฟสแบบอ้กขระ

การจ้ดตารางเวลางานโดยใช้ System i Navigator

น้้าต่าง Job Properties – Job Queue นำเสนอวิธีการจ้ดตารางเวลางานแบ้ตซ์เพื่อรันทันที รันครั้งเดียวที่วันที่และเวลาที่ระบุ หรือรัน ในช่วงสม้เสมอ (เช่นวันที่หนึ่งของแต่ละเดือน)

ในการจ้ดตารางเวลางานโดยใช้ System i Navigator ให้ใช้ข้อแนะนำดังต่อไปนี้:

1. ขยาย My Connections → *your system* → Work Management → Job Queues → Active Job Queues หรือ All Job Queues → คิวงานที่มีงานของคุณ.
2. คลิกขวาที่งานและคลิก Properties.
3. บนน้้าต่าง Job Properties ให้คลิกแท็บ Job Queues
4. เพื่อจ้ดตารางเวลางาน ให้ใช้อ้อพชั่นที่อยู่ ภายใต้ When to make job available to run.

สำหรัข้อมูลเก่ยกับวิธีการใช้น้้าต่างนี้ ให้ดูคำอธิบาย System i Navigator

การจ้ดตารางเวลางานโดยใช้ Management Central Scheduler

ถ้าคุณไม่ได้ติดตั้ง plug-in Advanced Job Scheduler คุณสามารถใช้ Management Central Scheduler เพื่อจ้ดตารางเวลางานได้

คุณสามารถเริ่ม Management Central scheduler โดยคลิกบนปุ่ม Schedule ที่ปรากฏบนน้้าต่าง System i Navigator หลาย น้้าต่าง ตัวอย่างเช่น สมมติว่าคุณต้องการใช้น้้าต่าง System i Navigator Run Command เพื่อส่งงานล่างข้อมูล แต่คุณไม่ ต้องการให้งานรันจนกว่าจะผ่านพ้นชั่วโมงที่มีการใช้งานสูงสุด

1. จาก System i Navigator คลิกขวาที่เซิร์ฟเวอร์ที่คุณต้องการจะรันงานล่างข้อมูล และคลิก Run Command.
2. จากน้้าต่าง Run Command, ให้พิมพ์ไวยากรณ์แบบอ้กขระสำหรัการรันงานของคุณ ถ้าคุณต้องการความช่วยเหลือ ให้พิมพ์คำสั่งแรกและ คลิก Prompt.
3. เมื่อคุณทำคำสั่งเสร็จสมบูรณ์แล้ว ให้คลิก Schedule. น้้าต่าง Management Central Scheduler จะแสดงขึ้น โดยในน้้า ต่างนี้ คุณสามารถจ้ดตารางเวลางานนี้ให้รันครั้งเดียว หรือให้รันเป็นงานที่ต่อเนื่อง

คุณสามารถจ้ดตารางเวลาภารกิจที่รันครั้งเดียว โดยภารกิจที่รันครั้งเดียวจะเริ่มในวันที่ และเวลาที่ระบุไว้ ภารกิจที่รัน เพียงครั้งเดียวถูกลบออกจาก Scheduled Tasks container เมื่อรันภารกิจ จากนั้น ปรากฏใน Task Activity container

สำคัญ: ห้ามใช้ Work with Job Schedule Entries (WRKJOBSCDE) เพื่อเปล่ยน หรือลบงานที่จ้ดตารางเวลาโดยใช้ Management Central Scheduler หรือ Advanced Job Scheduler ถ้างานถูกเปล่ยน หรือลบออกโดยใช้ WRKJOBSCDE, Management Central จะไม่ได้รับการแจ้งถึงการเปล่ยนเปล่ง ภารกิจไม่สามารถรันได้ตามที่ คาด และข้อความแสดงความผิดพลาดอาจปรากฏขึ้นในบันทึกการใช้งานเซิร์ฟเวอร์ Management Central

ถ้าจำเป็นต้องทำการเปล่ยนเปล่งงานที่จ้ดตารางเวลาโดยใช้ Management Central Scheduler หรือ Advanced Job Scheduler ให้ใช้ System i Navigator อินเทอร์เน็ตเฟส

หลักการที่เกี่ยวข้อง

“Management Central Scheduler” ในหน้า 62

System i Navigator นำเสนอ integrated scheduler, Management Central scheduler เพื่อจ้ดตารางเวลาที่คุณต้องการให้งาน

ของคุณประมวลผล คุณมีอ็อปชันการเลือกที่จะปฏิบัติ ภารกิจในทันทีหรือในภายหลัง คุณสามารถใช้ Management Central scheduler เพื่อจัดตารางเวลาภารกิจส่วนใหญ่ใน Management Central

Advanced Job Scheduler

ตัว IBM Advanced Job Scheduler for i5/OS (5761-JS1) โลเซ็นส์โปรแกรมเป็นตัวจัดลำดับงานที่ยืดหยุ่นที่ช่วยให้งานแบบ unattended ถูกประมวลผลได้ตลอด 24 ชั่วโมงต่อวัน, 7 วันต่อสัปดาห์ เครื่องมือการจัดตารางเวลานี้แนะนำคุณลักษณะปฏิทินเพิ่มเติม และนำเสนอการควบคุมเหตุการณ์ที่จัดตารางเวลาในระดับที่สูงขึ้นกว่าตัวจัดตารางเวลา Management Central นอกจากนี้ คุณยังสามารถดูประวัติความสมบูรณ์ของงานและบริหารการแจ้งเตือนสถานะของงานได้อีกด้วย

หากคุณต้องการจัดตารางเวลางานบนหลายระบบในเน็ตเวิร์กของคุณ ต้องติดตั้งผลิตภัณฑ์บนแต่ละระบบของคุณ หากคุณต้องการใช้ Advanced Job Scheduler ใน System i Navigator (และใน Management Central) จากนั้น คุณต้องติดตั้ง client plug-in จากระบบที่ติดตั้ง Advanced Job Scheduler ไว้

ทั้งนี้ไม่จำเป็นต้องติดตั้ง Advanced Job Scheduler โลเซ็นส์โปรแกรมบนระบบจุดปลายทุกระบบในเน็ตเวิร์ก Management Central ของคุณ เมื่อคุณติดตั้ง Advanced Job Scheduler บนระบบศูนย์กลาง งาน หรือภารกิจที่คุณกำหนดบนระบบจุดปลาย จะรวบรวมข้อมูล, ที่ต้องการจากระบบศูนย์กลาง คุณต้องตั้งค่าข้อมูล job definition ทั้งหมดบนระบบศูนย์กลาง

ถ้าระบบในเน็ตเวิร์กของคุณมี Advanced Job Scheduler ติดตั้งไว้แบบโลคัล คุณสามารถจัดตารางเวลาภารกิจภายนอกเน็ตเวิร์ก Management Central ได้ ภายใต้ My Connections ใน System i Navigator, คุณมีสิทธิเข้าใช้งาน Advanced Job Scheduler บนระบบโลคัลเมื่อคุณขยาย Work Management.

หมายเหตุ: สำหรับข้อมูลในการสั่งซื้อ ให้ดูที่ Job Scheduler สำหรับ i5/OS  Web site.

Advanced Job Scheduler for Wireless:

Advanced Job Scheduler for Wireless เป็นแอปพลิเคชันที่ช่วยให้คุณเข้าใช้งาน Advanced Job Scheduler บนอุปกรณ์เชื่อมต่ออินเทอร์เน็ตหลายประเภท เช่น โทรศัพท์ที่พร้อมเชื่อมต่อกับอินเทอร์เน็ต, PDA Web browser หรือ PC Web browser

คุณลักษณะพิเศษไร้สายของ Advanced Job Scheduler มีอยู่บนระบบของคุณ ซึ่ง Advanced Job Scheduler การติดตั้ง และอนุญาตให้คุณเข้าถึงงานของคุณ และ activity ตลอดจนสามารถส่งข้อความให้ผู้รับบนระบบของคุณ และหยุด แล้วเริ่ม Advanced Job Scheduler การมอนิเตอร์ Advanced Job Scheduler for Wireless อนุญาตให้ผู้ใช้ทุกรายปรับการตั้งค่า และความชอบตามความต้องการในการใช้การบราวซ์ของตนเอง ตัวอย่าง ผู้ใช้คนหนึ่งสามารถแสดง activity, แสดงงานต่างๆ และปรับงานที่แสดงตามความต้องการ

Advanced Job Scheduler for Wireless อนุญาตให้คุณเข้าถึงงานของคุณ เมื่อคุณไม่สามารถเข้าใช้งาน System i เทอร์มินัล หรือ อีเมลเตอร์ เชื่อมต่อกับอินเทอร์เน็ตด้วยอุปกรณ์เคลื่อนที่ของคุณ และป้อน URL สำหรับ Advanced Job Scheduler for Wireless servlet การทำเช่นนี้จะเรียกใช้เมนูที่คุณสามารถเข้าใช้งานแบบเรียลไทม์กับ Advanced Job Scheduler.

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

การจัดตารางเวลางานด้วย Advanced Job Scheduler:

ในการจัดการ Advanced Job Scheduler, คุณต้องติดตั้งไลเซนส์โปรแกรมก่อน แล้วจึงดำเนินการทำให้เสร็จสิ้น เพื่อปรับตามความต้องการ Advanced Job Scheduler. ท้ายที่สุด ภารกิจส่วนที่เหลือช่วยให้คุณสามารถทำงาน และบริหารตัวจัดตารางเวลานี้ได้

การติดตั้งตัวจัดลำดับงานระดับสูง:

ในครั้งแรกที่คุณเชื่อมต่อกับเซิร์ฟเวอร์ Management Central ของคุณ System i Navigator ให้ถามว่าคุณต้องการติดตั้ง Advanced Job Scheduler. ถ้าคุณเลือกไม่ติดตั้งในขณะนี้ แต่ต้องการติดตั้งภายหลัง คุณสามารถทำได้โดยใช้คุณลักษณะ Install Plug-ins feature ของ System i Navigator.

1. จากหน้าต่าง System i Navigator ของคุณ ให้คลิก File จากเมนูบาร์
2. เลือก Install Options → ติดตั้ง Plug-ins.
3. คลิกระบบซอร์สซึ่งได้รับ Advanced Job Scheduler การติดตั้ง แล้วคลิก OK. ตรวจสอบกับผู้ดูแลระบบ ถ้าคุณไม่แน่ใจว่าจะใช้ระบบซอร์สอะไร
4. ใส่ i5/OS User ID และ Password แล้วคลิก OK.
5. คลิก Advanced Job Scheduler จากรายการเลือก Plug-in
6. คลิก Next แล้วคลิก Next อีกครั้ง
7. คลิก Finish เพื่อให้สมบูรณ์และออกจากหน้าต่าง

ขณะนี้คุณสามารถติดตั้ง Advanced Job Scheduler.

การระบุตำแหน่งตารางเวลา:

ถ้าต้องการระบุตำแหน่งตารางเวลาให้ทำตามขั้นตอนต่อไปนี้:

1. ให้ขยาย Management Central.
2. คลิก Scan Now เมื่อเห็นข้อความว่า System i Navigator ตรวจพบส่วนประกอบใหม่ คุณอาจเห็นข้อความนี้อีกครั้งเมื่อคุณเข้าใช้งาน ระบบจาก My Connections container
3. ขยาย My Connections เลือกระบบที่มี Advanced Job Scheduler ไลเซนส์โปรแกรมที่ได้รับการติดตั้ง แล้วจึงเลือก Work Management → Advanced Job Scheduler.

หลังจากที่คุณทำงานเบื้องต้นเกี่ยวกับ Advanced Job Scheduler คุณก็พร้อมที่จะตั้งค่า Advanced Job Scheduler.

การตั้งค่า Advanced Job Scheduler:

ก่อนเริ่มต้นงานตามกำหนดการ คุณต้องกำหนด Advanced Job Scheduler.

การกำหนดคุณสมบัติทั่วไป:

คุณสามารถปฏิบัติตามคำสั่งเหล่านี้เพื่อกำหนดคุณสมบัติทั่วไปที่ใช้โดย Advanced Job Scheduler. คุณสามารถระบุระยะเวลาการเก็บรักษา activity และ log entry สำหรับ Advanced Job Scheduler รวมทั้งช่วงเวลาทำงานจะไม่ได้รับอนุญาตให้รัน

คุณสามารถระบุวันทำงานที่จะให้งานทำการประมวลผล และต้องมีแอ็พพลิเคชันสำหรับงานที่จัดตารางไว้แต่ละงานหรือไม่ ถ้าคุณสามารถติดตั้งผลิตภัณฑ์ที่ได้ประกาศแล้ว คุณยังสามารถตั้งค่าคำสั่งที่จะใช้เพื่อส่งการแจ้งเตือนงานเสร็จสิ้น หรือล้มเหลว หรือคุณสามารถใช้คำสั่ง Send Distribution using Job Scheduler (SNDDSTJS) เพื่อแจ้งเตือนผู้รับ

คุณสามารถระบุระยะเวลาการเก็บรักษาบันทึก activity ของงาน รวมทั้งช่วงเวลาที่งานจะไม่ได้รับอนุญาตให้รัน คุณสามารถระบุวันที่จะให้งานทำการประมวลผล และต้องมีแอ็พพลิเคชันสำหรับงานที่จะส่งแต่ละงานหรือไม่

คุณสามารถมีผลิตภัณฑ์ที่ได้แจ้งไว้ล่วงหน้าติดต่อกันซึ่งจะทำให้คุณได้รับการเตือน (ข้อความ) เมื่องานจบลง คุณสามารถกำหนดคำสั่งการแจ้งเตือนซึ่งจะส่งข้อความเตือนเมื่องานเสร็จสมบูรณ์ หรือล้มเหลว หรือคุณสามารถใช้คำสั่ง Send Distribution using Job Scheduler (SNDDSTJS) เพื่อแจ้งเตือนผู้รับ

ถ้าต้องการตั้งค่าคุณสมบัติทั่วไปสำหรับ Advanced Job Scheduler ให้ทำตามขั้นตอนเหล่านี้:

1. ขยาย **Work Management** จาก System i Navigator หน้าต่างของคุณ
2. คลิกขวา **Advanced Job Scheduler** แล้วคลิก **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 ระบุจำนวนวัน ระยะเวลาที่คุณต้องการเก็บรักษา Advanced Job Scheduler log entries
5. คุณสามารถระบุ **Reserved period**. งานจะไม่รันในระหว่างช่วงเวลานี้
6. ให้ระบุวันที่ทำงานจากในรายการ ถ้ามีการเลือกวัน วันจะถูกกำหนดให้เป็นวันทำงานและสามารถใช้อ้างอิงเมื่อจัดตารางเวลางาน
7. คลิก **Application required for scheduled job** เพื่อกำหนดว่า จำเป็นต้องใช้แอ็พพลิเคชันสำหรับแต่ละงานที่จัดตารางเวลาหรือไม่ แอ็พพลิเคชัน คืองานที่ถูกจัดกลุ่มเข้าด้วยกันสำหรับการประมวลผล จะไม่สามารถเลือกได้ถ้างานที่มีอยู่ไม่ประกอบด้วยแอ็พพลิเคชัน ถ้าคุณเลือกที่จะมี แอ็พพลิเคชันที่ต้องการสำหรับงาน ให้ไปที่การทำงานกับแอ็พพลิเคชัน
8. คลิก **Calendars** เพื่อตั้งค่าการจัดตารางเวลา, วันหยุด, และปีบัญชีที่จะใช้, ตั้งค่าปฏิทินวันหยุด, และตั้งค่า ปีบัญชี
9. คลิก **Base periodic frequency on start time** เพื่อกำหนดพื้นฐานของรันใหม่บนเวลาเริ่มต้นสำหรับงานที่จะถูกจัดตารางเวลาให้รันเป็นช่วงเวลา ตัวอย่างเช่น งานจะรันทุก 30 นาที เริ่มทำงานที่เวลา 8:00 น. (สำหรับงานที่รันตามรอบนาฬิกา ให้ระบุ 7:59 น. เป็นเวลาสิ้นสุดงาน) งานที่รันเป็นเวลาทั้งหมด 20 นาที ด้วยการเลือกฟิลด์นี้ งานจะรันที่ 8:00 น., 8:30 น., 9:00 น. และต่อไป หากไม่เลือกฟิลด์นี้ งานจะรันที่ 8:00 น., 8:50 น., 9:40 น., 10:30 น. และต่อไป
10. คลิก **Reset held jobs** เพื่อคำนวณใหม่ และแสดงวันที่และเวลาที่ไปทำงานที่พักไว้จะรัน
11. ระบุ **Start time of day**. นี่เป็นเวลาของวันที่คุณพิจารณาว่าเป็นการเริ่มต้นวันใหม่ งานทั้งหมดที่ถูกระบุให้ใช้เวลาของวันจะมีวันที่เปลี่ยนเป็นวันก่อนหน้า ถ้าเวลาที่งานเริ่มต้นเป็นเวลาก่อนหน้าฟิลด์ **Start time of day**
12. ระบุ **Job monitor user**. ฟิลด์นี้ระบุชื่อของโปรไฟล์ผู้ใช้ที่ใช้เป็นเจ้าของของงานการมอนิเตอร์งานทั้งหมดที่มีการระบุ **Current user** จะใช้โปรไฟล์ผู้ใช้ของงานการมอนิเตอร์โปรไฟล์ผู้ใช้ที่ฟิลด์ของงานการมอนิเตอร์คือ QIUS
13. ในฟิลด์ **Notification command** คุณสามารถ ระบุคำสั่งได้ ใช้คำสั่ง Send Distribution using Job Scheduler Notification (SNDDSTJS) ที่ให้มาพร้อมกับระบบ หรือคำสั่งที่ระบุโดยซอฟต์แวร์ การแจ้งเตือนของคุณ คำสั่ง SNDDSTJS จะใช้ Advanced Job Scheduler ฟังก์ชันการเตือน ผู้รับที่กำหนดสามารถได้รับข้อความ job schedule entry ที่จับแบบปกติและแบบไม่ปกติ

การระบุค่าระดับการอนุญาต:

ข้อมูลนี้อธิบายวิธีการระบุระดับการอนุญาตสำหรับงาน, ฟังก์ชันของผลิตภัณฑ์ และแสดงการอนุญาตดีฟิลด์สำหรับงานใหม่

คุณสามารถระบุระดับการอนุญาตสำหรับงาน, ฟังก์ชันของผลิตภัณฑ์ และแสดงการอนุญาตดีฟอลต์สำหรับงานใหม่ที่จะใช้กับแต่ละ Job Control/Application การอนุญาตสำหรับงานช่วยให้คุณช่วยให้หรือปฏิเสธการเข้าถึงการปฏิบัติการต่อไปนี้: การส่ง, การจัดการ, การอนุญาต, การแสดงผล, การทำสำเนา, การอัปเดต หรือการลบ คุณยังสามารถให้หรือปฏิเสธการเข้าใช้งานแต่ละฟังก์ชันของผลิตภัณฑ์นี้ด้วย เช่น Work with Schedule Calendars, Send Reports และ Add Job

ค่าดีฟอลต์ระดับการอนุญาตถูกถ่ายโอนไปยังงานใหม่เมื่อมีการเพิ่มงานใหม่เข้าไปในแต่ละกรณี ระบบจะถ่ายโอนการอนุญาตสำหรับงานใหม่ตามที่ระบุในแอ็พพลิเคชันภายใน job definition ถ้าไม่มีการใช้แอ็พพลิเคชัน ระบบจะถ่ายโอนค่าการอนุญาตงานใหม่เป็น *SYSTEM

การระบุระดับการอนุญาตสำหรับฟังก์ชันของผลิตภัณฑ์:

ถ้าต้องการระบุระดับการอนุญาตสำหรับฟังก์ชันของผลิตภัณฑ์ให้ทำตามขั้นตอนเหล่านี้:

1. จาก System i Navigator ให้ขยาย **Work Management**.
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. จาก System i Navigator ให้ขยาย **Work Management**.
2. คลิกขวา **Advanced Job Scheduler** และคลิก **Scheduled Jobs** เพื่อแสดงรายการงาน
3. คลิกขวางานที่จัดตารางเวลา และคลิก **Permissions**.
4. บนหน้าต่าง Permissions Properties ให้แก้ไขระดับการอนุญาต ตามความจำเป็น คุณสามารถให้หรือปฏิเสธการเข้าใช้งานของผู้ใช้ปกติหรือผู้ใช้เฉพาะได้ ยิ่งไปกว่านั้น คุณสามารถระบุการส่ง, การจัดการ, การอนุญาต, การแสดงผล, การทำสำเนา, การอัปเดต หรือการลบการอนุญาตต่างๆ ได้

การระบุระดับการอนุญาตดีฟอลต์:

ถ้าต้องการระบุระดับการอนุญาตดีฟอลต์สำหรับงานใหม่ที่เกี่ยวข้องกับ Job Control/Application ให้ทำตามขั้นตอนเหล่านี้:

1. จาก System i Navigator ให้ขยาย **Work Management**.
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 คุณสามารถให้หรือปฏิเสธการเข้าใช้งานของผู้ใช้ปกติหรือผู้ใช้เฉพาะได้ ยิ่งไปกว่านั้น คุณสามารถระบุการส่ง, การจัดการ, การอนุญาต, การแสดงผล, การทำสำเนา, การอัปเดต หรือการลบการอนุญาตต่างๆ ได้

การตั้งค่าปฏิทินตารางเวลา:

ข้อแนะนำเหล่านี้แสดงวิธีตั้งค่าปฏิทินของวันที่เลือกสำหรับการจัดตารางเวลางาน หรือกลุ่มงาน ปฏิทินนี้สามารถระบุวันที่ที่จะนำมาใช้สำหรับการจัดตารางเวลางานหนึ่ง หรือสามารถนำไปใช้ร่วมกับตารางเวลาอื่นๆ

ปฏิทินตารางเวลา คือปฏิทินของวันที่คุณเลือกซึ่งอาจใช้สำหรับการจัดตารางเวลางานหรือกลุ่มงาน คุณอาจแสดงผลปฏิทินตารางเวลา, ใส่เพิ่มปฏิทินตารางเวลาใหม่, ใส่เพิ่มปฏิทินตารางเวลาใหม่โดยใช้ฐานเดิมจากปฏิทินตารางเวลาที่มีอยู่แล้ว หรือลบปฏิทินที่มีอยู่แล้วออก ถ้าปฏิทินไม่ได้ถูกใช้โดยงานที่จัดตารางเวลาในปัจจุบัน

คุณสามารถเลือกปฏิทิน และแสดงผลคุณสมบัติของมันเพื่อทำการเปลี่ยนแปลงได้ เมื่อคุณเลือกปฏิทิน รายละเอียดของปฏิทินจะถูกแสดงขึ้นภายใต้ Details

ถ้าต้อง การตั้งค่าปฏิทินการจัดตารางเวลา ให้ทำตามขั้นตอนเหล่านี้:

1. ขยาย **Work Management** จาก System i Navigator หน้าต่างของคุณ
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 นี้คือ ปฏิทินที่ได้ตั้งค่าไว้ก่อนหน้าแล้ว และคุณสมบัติของมันจะถูกนำมาประยุกต์ใช้กับปฏิทินใหม่ราวกับว่าคุณผสมรวมสองปฏิทินเข้าด้วยกัน คุณจะไม่มีปฏิทินอ้างอิงถ้าครั้งนี้เป็นการเรียกใช้ตัวจัดลำดับงานระดับสูงครั้งแรกของคุณ
8. เลือกวันที่ที่คุณต้องการเพิ่มลงในปฏิทินของคุณ คุณต้องระบุว่าแต่ละวันที่ที่คุณได้เลือกจะใช้สำหรับปีปัจจุบัน หรือทุกๆปีในฟิลด์ **Selected date** ก่อนที่คุณจะสามารถใส่วันที่อื่นถัดไปเพิ่มในปฏิทิน มิฉะนั้น วันที่ใดๆที่คุณเลือกจะถูกยกเลิกเมื่อคุณคลิกวันที่อื่น
9. ให้ระบุ ถ้าคุณต้องการวันแน่นอนของสัปดาห์ เพิ่มสอดแทรกบนปฏิทิน

การตั้งค่าปฏิทินวันหยุด:

ข้อแนะนำเหล่านี้แสดงวิธีตั้งค่าปฏิทินสำหรับวันที่คุณไม่ต้องการให้มีการประมวลผลงานที่กำหนดตารางเวลาไว้สามารถระบุวันที่ทางเลือกสำหรับวันยกเว้นแต่ละวัน หรือสามารถข้ามการประมวลผลสำหรับวันนั้น

ปฏิทินวันหยุด เป็นปฏิทิน exception สำหรับวันในปฏิทินที่คุณไม่ต้องการประมวลผลงาน Advanced Job Scheduler สามารถระบุวันที่ทางเลือกสำหรับวันยกเว้นแต่ละวัน ซึ่งคุณระบุในปฏิทิน วันหยุด คุณสามารถแสดงผลปฏิทินวันหยุด, ใส่เพิ่มปฏิทินวันหยุดใหม่, ใส่เพิ่มปฏิทินวันหยุดใหม่โดยใช้ฐานเดิมจากปฏิทินที่มีอยู่แล้ว หรือลบปฏิทินที่มีอยู่แล้วออก ถ้าปฏิทินไม่ได้ถูกใช้โดยงานที่จัดตารางเวลาในปัจจุบัน

เราสามารถใชัตารางเวลาที่กำหนดไว้ล่วงหน้าในปฏิทินวันหยุด คุณสามารถสร้างตารางเวลา THIRDFRI ที่มีความถี่ในทุกๆวันศุกร์ที่สามของแต่ละเดือน เมื่อคุณใช้ THIRDFRI ใน ปฏิทินวันหยุด นั้นหมายความว่ากำหนดให้งานทั้งหมดที่ใช้ปฏิทินวันหยุดนี้ ไม่ว่านในวันศุกร์ที่สามของทุกเดือน เราสามารถใชัตารางเวลามากกว่าหนึ่งตารางในปฏิทินวันหยุดหนึ่ง วันที่ที่ถูกสร้างขึ้นมาจากตารางเวลาจะถูกแสดงอยู่ในปฏิทินโดยมีกรอบสีดำล้อมรอบ

คุณสามารถเลือกปฏิทิน และแสดงผลคุณสมบัติของมันเพื่อทำการเปลี่ยนแปลงได้ เมื่อคุณเลือกปฏิทิน รายละเอียดของปฏิทินจะถูกแสดงขึ้นภายใต้ Details

การตั้งค่าปฏิทินวันหยุด:

ถ้าต้องการตั้งค่าปฏิทินวันหยุด ให้ทำตามขั้นตอนต่อไปนี้:

1. ขยาย **Work Management** จากหน้าต่าง System i Navigator ของคุณ
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 นี่คือ ปฏิทินที่ได้ตั้งค่าไว้ก่อนหน้านี้แล้ว และคุณสมบัติของมันจะถูกนำมาประยุกต์ใช้กับปฏิทินใหม่ราวกับว่าคุณผสมรวมสองปฏิทินเข้าด้วยกัน คุณจะไม่มีปฏิทินอ้างอิงถ้าครั้งนี้เป็นการเรียกใช้ตัวจัดลำดับงานระดับสูงครั้งแรกของคุณ
8. เลือกวันที่ที่คุณต้องการเพิ่มลงในปฏิทินของคุณ คุณต้องระบุว่าแต่ละวันที่คุณเลือกจะใช้สำหรับปีปัจจุบัน หรือทุกๆ ปีในฟิลด์ **Selected date** ก่อนที่คุณจะสามารถใส่วันอื่นถัดไปเพิ่มในปฏิทิน มิฉะนั้น วันที่ใดๆที่คุณเลือกจะถูกยกเลิกเมื่อคุณคลิกวันที่อื่น
9. ให้เลือกวันทางเลือกสำหรับวันงาน คุณสามารถเลือกวันทำงานก่อนหน้านี้, วันทำงานถัดไป, วันที่เฉพาะ หรือไม่เลือกเลย ถ้าต้องการเลือกวันที่เฉพาะ ให้คลิก **Specific alternate date** และพิมพ์วันที่ลงไป
10. เลือกเจาะจงวันในสัปดาห์ที่จะแทรกเพิ่มเติมลงไป ในปฏิทิน

การใส่เพิ่มตารางเวลาในปฏิทินวันหยุด:

ถ้าต้องการใส่เพิ่มปฏิทินวันหยุดลงในงานที่จัดตารางเวลา ให้ทำตามขั้นตอนเหล่านี้:

1. ขยาย **Work Management** จากหน้าต่าง System i Navigator ของคุณ
2. คลิกขวา **Advanced Job Scheduler** และคลิก **Properties**.
3. บนหน้า **General** คลิก **Calendars**.
4. บนหน้า **Holiday calendar** เลือกปฏิทินวันหยุดและคลิก **Properties**.
5. จากมุมมองซ้ายล่างของแท็บ, คลิก **Schedules**.
6. เลือกตารางเวลาที่เหมาะสม และคลิก **Add**.
7. เพื่อเปลี่ยน **Alternate day** ให้คลิกขวาที่ ตารางเวลาจากรายการ **Selected Schedules** และคลิก **Alternate Day** ที่ถูกต้อง

การตั้งค่าปฏิทินปีบัญชี:

หากคุณต้องการแบ่งปีบัญชีเป็นช่วงๆที่ไม่ใช่เป็นเดือน ให้ปฏิบัติตามขั้นตอนเหล่านี้เพื่อตั้งค่าปีบัญชีของวันที่เลือกสำหรับการจัดตารางเวลางาน หรือกลุ่มงาน

ปฏิทินปีบัญชีคือปฏิทินของวันที่คุณเลือก ซึ่งคุณสามารถใช้สำหรับการจัดตารางเวลางานหรือกลุ่มงาน ใช้ปฏิทินปีบัญชีเพื่อกำหนดปฏิทินปีบัญชีเฉพาะสำหรับธุรกิจของคุณ คุณสามารถระบุวันที่เริ่มต้นและวันที่สิ้นสุดสำหรับแต่ละรอบระยะเวลาในปีบัญชีได้

ถ้าต้องการตั้งค่า ปฏิทินปีบัญชี ให้ทำตามขั้นตอนเหล่านี้:

1. ขยาย **Work Management** จากหน้าต่าง System i Navigator ของคุณ.
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 ชั่วโมง
9. คลิก OK เพื่อบันทึก fiscal calendar entry
10. ทำซ้ำขั้นตอนที่ 7 ถึง 9 ตามความจำเป็น

การระบุเมลเซิร์ฟเวอร์เพื่อใช้ในการแจ้งเตือน:

ต้องใช้เมลเซิร์ฟเวอร์ ถ้าคุณต้องการส่งการแจ้งเตือนทางอีเมล

ถ้าต้องการตั้งค่าคุณสมบัติการแจ้งเตือน ให้ทำตามขั้นตอนเหล่านี้:

1. ขยาย Work Management จากหน้าต่าง System i Navigator ของคุณ
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 โลบารรีข้อมูลต้นฉบับสามารถทำหน้าที่เป็นโลบารรีข้อมูลที่แอคทีฟ และโลบารรีข้อมูลที่ทำสำเนาสามารถใช้สำหรับการทดสอบ ดังนั้นคุณจึงมีสถานะแวดล้อมการจัดตารางเวลา สองสถานะ สถานะหนึ่งสำหรับการทดสอบและอีกสถานะหนึ่งคือสถานะจริง นอกจากนี้ โลบารรีข้อมูลที่ใช้ทดสอบสามารถทำหน้าที่เป็นตัวสำรองข้อมูลได้ ถ้าระบบดั้งเดิมล้มเหลว คุณลักษณะนี้ช่วยให้คุณมีความปลอดภัยมากขึ้น ถ้าคุณสร้างข้อผิดพลาดในโลบารรีข้อมูลต้นฉบับ เพราะว่าคุณมีสำเนาสำรองของโลบารรีข้อมูล

มีหลายเหตุผลว่าเพราะเหตุใดคุณอาจต้องการตั้งค่าสถานะแวดล้อม การจัดตารางเวลาหลายสถานะ คุณอาจต้องการมีผลิตภัณฑ์เวอร์ชันการผลิตและ เวอร์ชันการทดสอบที่รันอยู่พร้อมกัน สถานะแวดล้อมแบบนี้ช่วยให้คุณทดสอบตารางเวลา งานต่างๆ ก่อนที่จะใช้ตารางจริงในโลบารรีข้อมูลบนระบบการผลิต หรือคุณอาจมีระบบหนึ่งที่เป็นตัวสำรองสำหรับระบบหนึ่งหรือหลายระบบซึ่งคุณสามารถใช้ผลิตภัณฑ์พวก data-mirroring เพื่อจำลองโลบารรีข้อมูล Advanced Job Scheduler (QUSRIJS) จากระบบต้นทางไปยังโลบารรีที่มีชื่อแตกต่างกัน ในกรณีนี้ โลบารรีข้อมูลยังคงแอคทีฟอยู่จนกว่าจะมีปัญหาเกิดขึ้นกับระบบต้นทาง

สถานะแวดล้อมการจัดตารางเวลาคือภาพจำลองของโลบารรี QUSRIJS เพียงแต่มีข้อมูลที่แตกต่างกัน ตัวอย่างเช่น คุณสามารถมีโลบารรีข้อมูลอื่นที่ชื่อว่า QUSRIJSTST โดยมีอ็อบเจกต์ทั้งหมดเหมือนกับที่มีใน QUSRIJS แต่ละโลบารรีถือว่าเป็นโลบารรีข้อมูลเช่นกัน

ถ้าต้องการ ตั้งค่าสภาวะแวดล้อมการจัดตารางเวลาหลายสภาวะ ให้ทำตามขั้นตอนเหล่านี้:

1. จัดหาไลบรารีข้อมูลจากระบบ

เพื่อสร้างไลบรารีข้อมูล คุณต้องจัดหาไลบรารีข้อมูลหนึ่งไลบรารีจากระบบ คุณสามารถจัดหาไลบรารีข้อมูลจากระบบได้สามวิธีต่อไปนี้:

- บันทึกไลบรารีข้อมูลจากระบบและจัดเก็บไลบรารีบนระบบ การผลิต
- ทำสำเนาไลบรารีข้อมูลบนระบบปัจจุบันโดยใช้คำสั่ง Copy Library (CPYLIB).
- มิเรอร์ไลบรารีข้อมูลบนระบบทดสอบ ระบบเหล่านี้ควรจะรันที่ระดับซีลิสเวอร์ชันเดียวกัน

หมายเหตุ: ไลบรารีข้อมูลที่ทำสำเนา, คัดลอก หรือมิเรอร์จะใช้ชื่อที่แตกต่างจาก ระบบดั้งเดิม

2. กำหนดไลบรารีข้อมูลให้ผู้ใช้

หลังจากที่คุณจัดหาไลบรารีข้อมูลสำหรับทดสอบแล้ว ให้ใส่เพิ่มไลบรารีข้อมูลไปยังคุณสมบัติของ Advanced Job Scheduler และกำหนดผู้ใช้งานให้กับไลบรารีข้อมูล ดังนั้น เมื่อมีผู้ใช้งานคนหนึ่งใช้งาน Advanced Job Scheduler การเปลี่ยนแปลงที่เกิดขึ้นจากผู้ใช้งานจะถูกเก็บอยู่ในไลบรารีข้อมูลที่กำหนดให้กับผู้ใช้งานคนนั้น

3. ทำสำเนางานจากไลบรารีข้อมูลทดสอบไปที่ไลบรารีข้อมูลจริง (เป็นอ็อปชัน)

ถ้าคุณใช้ไลบรารีข้อมูลเพื่อการทดสอบ คุณอาจต้องการทำสำเนางานจากไลบรารีข้อมูลทดสอบไปยังไลบรารีข้อมูลจริงที่จะใช้ คุณต้องทำอย่างนี้ เฉพาะถ้าคุณได้คัดลอกหรือทำสำเนาไลบรารีข้อมูลในขั้นตอนที่ 1 และมีงานที่คุณต้องการย้ายไปไว้ที่ไลบรารีข้อมูลจริงที่จะใช้ คุณไม่จำเป็นต้องทำเช่นนี้ ถ้าคุณมิเรอร์ไลบรารีข้อมูลจากระบบจริงไปยังระบบทดสอบเพื่อทำสำเนางานจากไลบรารีข้อมูลของระบบหนึ่งไปยังระบบอื่น ให้ใช้คำสั่ง Copy Job using Job Scheduler (CPYJOBS) สำหรับข้อมูลเพิ่มเติมเกี่ยวกับพารามิเตอร์เฉพาะสำหรับคำสั่งนี้ ให้ดูคำอธิบาย ทางออนไลน์

การกำหนดไลบรารีข้อมูลให้ผู้ใช้:

ไลบรารีข้อมูลจะจัดเก็บการเปลี่ยนแปลงใดๆที่ผู้ใช้ทำโดยใช้ Advanced Job Scheduler ไลบรารีข้อมูลประกอบด้วยอ็อบเจกต์ทั้งหมดที่พบอยู่ในไลบรารี QUSRJIS คุณสามารถมิเรอร์ไลบรารีข้อมูลได้ไม่จำกัดจำนวน

ถ้าต้องการกำหนดไลบรารีข้อมูลให้แก่ผู้ใช้ให้ทำตามขั้นตอนเหล่านี้:

1. ขยาย **Work Management** จากหน้าต่าง System i Navigator ของคุณ
2. คลิกขวา **Advanced Job Scheduler** และคลิก **Properties**.
3. บนหน้าต่าง Data Libraries, คลิก **Add** เพื่อ ระบุไลบรารีข้อมูล ไลบรารีข้อมูลที่อยู่ในรายการเป็นไลบรารีที่พร้อมใช้งานสำหรับผู้ใช้ทั้งหมดบนระบบ
4. บนหน้าต่าง Users คลิก **Add** เพื่อใส่เพิ่มผู้ใช้ใหม่
5. ระบุชื่อ
6. เลือกไลบรารีข้อมูล
7. คลิก **OK** เพื่อใส่เพิ่มผู้ใช้
8. คลิก **Properties** เพื่อเปลี่ยนไลบรารีข้อมูลที่ได้กำหนดให้แก่ผู้ใช้

ด้วยการใช้ไลบรารีข้อมูล คุณสามารถตั้งค่าสภาวะแวดล้อมการจัดตารางเวลาได้หลายสภาวะ

การบริหาร Advanced Job Scheduler:

ข้อมูลนี้แสดงวิธีจัดตารางเวลางานโดยใช้ Advanced Job Scheduler

การสร้างและจัดตารางเวลางาน:

คุณสามารถจัดตารางเวลางาน และระบุคำสั่งที่เกี่ยวข้องกับงานได้ คุณยังสามารถระบุคำสั่งการเริ่มต้นและสิ้นสุดเพื่อวันเวอร์ชันพิเศษของงานที่จัดตารางเวลาได้ด้วย

ถ้าต้องการสร้างและจัดตารางเวลางานที่จัดตารางเวลาใหม่ ให้ทำตามขั้นตอนเหล่านี้:

1. ขยาย **Work Management** จากหน้าต่าง System i Navigator ของคุณ
2. คลิกขวาที่ **Advanced Job Scheduler**.
3. คลิกขวา **Scheduled Jobs** และคลิก **New Scheduled Job**.

การสร้างและจัดตารางเวลากลุ่มงาน:

คุณสามารถตั้งค่า และจัดตารางเวลาลำดับของงานที่รันอย่างต่อเนื่องตามลำดับที่ระบุ งานที่อยู่ภายในกลุ่มงานหนึ่งจำเป็นต้องเสร็จสิ้นลงก่อนมีการส่งงานถัดไปเข้ามาประมวลผล

กลุ่มงานคือ งานต่างๆ ที่ถูกจัดกลุ่มเข้าด้วยกันเพื่อรันอย่างต่อเนื่องตามลำดับที่ระบุ การจบงานลงอย่างปกติโดยสมบูรณ์เป็นสิ่งจำเป็นสำหรับงานแต่ละงานในกลุ่ม ก่อนที่งานถัดไปในกลุ่มจะถูกส่งเข้ามาประมวลผล ถ้างานใดๆ ในกลุ่มไม่จบลงแบบสมบูรณ์ตามปกติ การประมวลผลจะหยุดลงสำหรับกลุ่มนั้น

ถ้าต้องการสร้างและจัดตารางเวลากลุ่มงานใหม่ ให้ทำตามขั้นตอนเหล่านี้:

1. ขยาย **Work Management** จากหน้าต่าง System i Navigator ของคุณ
2. คลิก **Advanced Job Scheduler**.
3. คลิกขวา **Job Groups** และคลิก **New Job Group**.

อ้างอิงคำอธิบายทางออนไลน์สำหรับข้อมูลเพิ่มเติม เมื่อคุณกรอกรายละเอียด สำหรับกลุ่มงานใหม่

ตารางเวลางานที่ได้ถูกกำหนดไว้ก่อน:

คุณสามารถสร้างตารางเวลาที่มีข้อมูลที่เป็นต่อการจัดตารางเวลางาน หรือการคำนวณวันที่ยกเว้นภายในปฏิทินวันหยุด

ตัวอย่างเช่น คุณสามารถสร้างตารางเวลา ENDOFWEEK ซึ่งมีข้อมูลของวันในสัปดาห์ที่จะทำงาน พร้อมด้วยข้อมูลของปฏิทินเพิ่มเติม ตารางเวลา ENDOFWEEK สามารถใช้ได้โดยงานทั้งหมดที่สอดคล้องกับความถี่ของการจัดตารางเวลานั้น คุณสามารถเข้าใช้งานคุณลักษณะนี้ผ่านทาง System i Navigator.

คุณสามารถใช้ตารางเวลาที่ได้ถูกกำหนดไว้ก่อนอันเดียวกันกับที่ใช้ในงานเกี่ยวกับปฏิทินวันหยุดของคุณ คุณสามารถสร้างตารางเวลา THIRDFRI ที่มีความถี่ในทุกๆ วันศุกร์ที่สามของแต่ละเดือน เมื่อคุณใช้ THIRDFRI ใน ปฏิทินวันหยุด นั้นหมายความว่า คุณกำหนดให้งานทั้งหมดที่ใช้ปฏิทินวันหยุดนี้ ไม่ว่านในวันศุกร์ที่สามของแต่ละเดือน คุณสามารถใช้หนึ่งหรือหลายตารางเวลาในหนึ่งปฏิทินวันหยุด วันที่ที่ถูกรสร้างขึ้นมาจากตารางเวลาจะถูกแสดงอยู่ในปฏิทินโดยมีกรอบสีดำล้อมรอบ

การตั้งค่าตารางเวลาที่ถูกกำหนดไว้ก่อน:

ถ้าต้องการตั้งค่าตารางเวลาที่ถูกกำหนดไว้ก่อน ให้ทำตามขั้นตอนเหล่านี้

1. ขยาย **Work Management** จากหน้าต่าง System i Navigator ของคุณ
2. คลิกขวา **Advanced Job Scheduler** และคลิก **Properties**.

3. คลิก **Schedules** แท็บ
4. คลิก **New** และพิมพ์ชื่อสำหรับตารางเวลา
5. พิมพ์คำอธิบายสำหรับตารางเวลา
6. เลือกความถี่และวันที่ที่คุณต้องการรวมในตารางเวลาของคุณ รวมไปถึงปฏิทินเพิ่มเติมใดๆ

อ้างอิงคำอธิบายทางออนไลน์สำหรับข้อมูลเพิ่มเติมในขณะที่คุณกรอกรายละเอียดสำหรับตารางเวลาใหม่

การใส่เพิ่มตารางเวลาที่งานที่จัดตารางเวลา:

ถ้าต้องการใส่เพิ่มตารางเวลาที่งานที่จัดตารางเวลาให้ทำตามขั้นตอนเหล่านี้:

1. ขยาย **Work Management** จากหน้าต่าง System i Navigator ของคุณ
2. คลิกขวาที่ **Advanced Job Scheduler**.
3. คลิก **Scheduled Jobs** เพื่อแสดงรายการของงาน
4. คลิกขวางานที่จัดตารางเวลาและคลิก **Properties**.
5. คลิก **Schedule** แท็บ
6. จากมุมมองของแท็บ เลือกชื่อพจนานุกรม Schedule ที่เหมาะสม

การใส่เพิ่มตารางเวลาในปฏิทินวันหยุด:

ปฏิทินวันหยุดเป็นปฏิทินยกเว้นสำหรับวันที่คุณไม่ต้องการ ประมวลผลงาน Advanced Job Scheduler สามารถระบุ วันทางเลือกสำหรับวันยกเว้นแต่ละวัน ซึ่งคุณระบุในปฏิทิน วันหยุด

ถ้าต้องการใส่เพิ่มตารางเวลาในปฏิทินวันหยุด ให้ทำตามขั้นตอนเหล่านี้:

1. ขยาย **Work Management** จากหน้าต่าง System i Navigator ของคุณ
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:

หลายครั้ง อาจจำเป็นต้องรันงานที่จัดตารางเวลาในขณะนี้ หรือในอนาคตเพิ่มเติมจากตารางเวลาปกติ ใช้คำสั่ง **Submit Job using Job Scheduler (SBMJOBJS)**, อีพจนานุกรม 7 จากจอแสดงผล **Work with Jobs** หรือ **Run** อีพจนานุกรมจาก System i Navigator. นอกจากนี้ ยังจำเป็นต้องประมวลผลเพียงบางส่วนของคำสั่งที่อยู่ในรายการคำสั่ง เมื่อตั้งค่าการรันพิเศษนี้

คำสั่ง **SBMJOBJS** ช่วยให้คุณสามารถระบุลำดับของคำสั่งการเริ่มต้น และการสิ้นสุดได้ ตัวอย่างเช่น **JOBA** มี 5 คำสั่ง, ลำดับที่ 10 ถึง 50 คุณสามารถระบุคำสั่ง **SBMJOBJS** เพื่อให้เริ่มต้นที่ลำดับ 20 และสิ้นสุดที่ลำดับ 40 การระบุเช่นนี้จะทำให้ข้ามลำดับ 10 และ 50

System i Navigator อนุญาตให้คุณเลือกคำสั่งเริ่มต้น และคำสั่งสิ้นสุดจากรายการคำสั่ง

ในการรันเวอร์ชันพิเศษของงานที่จัดตารางเวลาไว้ด้วย System i Navigator ให้ปฏิบัติตามขั้นตอนเหล่านี้:

1. ขยาย **Work Management** จากหน้าต่าง System i Navigator ของคุณ
2. คลิกขวาที่ **Advanced Job Scheduler**.
3. คลิก **Scheduled Jobs** เพื่อแสดงรายการของงาน
4. คลิกขวาที่งานที่จัดตารางเวลา และคลิก **Run**.
5. ระบุว่าจะรันงานในขณะนี้หรือในอนาคต
6. เลือกคำสั่งเริ่มต้นและคำสั่งสิ้นสุด

อ้างอิงคำอธิบายทางออนไลน์สำหรับข้อมูลเพิ่มเติม เมื่อคุณกรอกรายละเอียด สำหรับงานใหม่

การจัดตารางเวลา job dependencies:

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 ทุกงาน 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 รายชื่อของงานที่ไม่สามารถแก้ไขเมื่องานที่เลือกถูกส่ง ถ้างานใดๆแก้ไขที่พออยู่ Advanced Job Scheduler จะไม่อนุญาตให้งานที่ระบุไว้ งานที่ถูกเลือกจะถูกหวัง จนกว่างานทั้งหมดในรายชื่อจะไม่แก้ไขที่พอ

- **Resource dependencies**

Resource dependencies มีพื้นฐานอยู่บนสองสามสิ่ง แต่ละชนิดที่ตามมาจะอธิบายถึงพื้นที่ที่ถูกตรวจสอบ ต่อไปนี้เป็นชนิดของ resource dependencies:

ไฟล์ งานขึ้นอยู่กับว่ามีอยู่หรือการไม่มีอยู่ของไฟล์ และขึ้นอยู่กับว่างานมีระดับการจัดสรรตรงกับระดับการจัดสรรที่ระบุในการประมวลผลหรือไม่ นอกจากนี้ยังสามารถตรวจสอบด้วยว่า มีการแสดงเร็กคอร์ดก่อนที่งานจะถูกประมวลผลหรือไม่ ตัวอย่างเช่น สามารถตั้งค่า JOBA เพื่อให้รันเฉพาะเมื่อมีไฟล์ ABC อยู่เท่านั้น และสามารถจัดไฟล์เฉพาะถ้ามีเร็กคอร์ดอยู่ในไฟล์

อ็อบเจกต์

งานขึ้นอยู่กับว่ามีอยู่หรือการไม่มีอยู่ของอ็อบเจกต์ชนิด QSYS และขึ้นอยู่กับว่างานมีระดับการจัดสรรตรงกับระดับการจัดสรรที่ระบุในการประมวลผลหรือไม่ ตัวอย่างเช่น สามารถตั้งค่า JOBA เพื่อให้รันเฉพาะเมื่อมีพื้นที่ข้อมูล XYZ อยู่เท่านั้น นอกจากนี้ งานยังอาจขึ้นอยู่กับว่ามีอยู่หรือการไม่มีอยู่ของอ็อบเจกต์ที่พบใน ระบบไฟล์รวม ถ้างานขึ้นอยู่กับ อ็อบเจกต์ใดๆ ในพาธ ให้จบพาธระบบไฟล์รวมด้วย forward slash '/'.

Hardware configuration

งานขึ้นอยู่กับมีการมีอยู่หรือการไม่มีอยู่ของ hardware configuration และสถานะของมันที่จะประมวลผล ตัวอย่างเช่น สามารถตั้งค่า JOBA เพื่อให้รันเฉพาะเมื่อมีอุปกรณ์ TAP01 อยู่ และมีสถานะเป็น Available

ไฟล์เน็ตเวิร์ก

งานขึ้นอยู่กับสถานะของไฟล์เน็ตเวิร์กที่จะประมวลผล

ระบบย่อย

งานขึ้นอยู่กับสถานะของระบบย่อยที่จะประมวลผล

ถ้าต้องการทำงานกับ job dependencies ให้ทำตามขั้นตอนเหล่านี้:

1. ขยาย **Work Management** จากหน้าต่าง System i Navigator ของคุณ
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:

Work Flow Manager อนุญาตให้คุณกำหนดหน่วยของงานที่ประกอบด้วยขั้นตอนแบบอัตโนมัติ หรือแบบทำเอง จากนั้นสามารถจัดตาราง เวลาและรันหน่วยของงานเหล่านี้แบบโต้ตอบได้ Work Flow Manager อยู่ใน Advanced Job Scheduler container ใน System i Navigator อินเทอร์เน็ต

แต่ละขั้นตอนภายในเวิร์กโฟลว์สามารถมี predecessor Advanced Job Scheduler job หนึ่งหรือหลายงาน และ successor Advanced Job Scheduler job หนึ่งหรือหลายงาน เมื่อเวิร์กโฟลว์เริ่มขึ้น ขั้นตอนแรกจะถูกแฟล็กเพื่อรัน เมื่อขั้นตอนแรกเสร็จสมบูรณ์แล้ว ขั้นตอนถัดไปจะถูกแฟล็กเพื่อรัน และต่อไป

ข้างล่างนี้เป็นข้อควรพิจารณาเพิ่มเติมบางอย่างเมื่อใช้ Work Flow Manager:

- คุณสามารถเริ่มเวิร์กโฟลว์ด้วยตนเองที่ขั้นตอนใดๆ เมื่อทำเช่นนั้น หมายความว่า คุณจะข้าม ขั้นตอนก่อนหน้านี้อย่างสมบูรณ์ในเวิร์กโฟลว์
- ขั้นตอนแบบอัตโนมัติจะเสร็จสมบูรณ์หลังจากที่ขั้นตอนก่อนหน้านี้อย่างสมบูรณ์ได้เสร็จสมบูรณ์แล้ว นี้รวม ถึง predecessor Advanced Job Scheduler jobs ทั้งหมดด้วย
- หลังจากที่ขั้นตอนเสร็จสมบูรณ์ successor Advanced Job Scheduler jobs จะ ถูกแฟล็กเพื่อรัน
- ขั้นตอนแบบทำเองสามารถเสร็จสมบูรณ์ในลำดับใดๆ ก็ได้ トラバเท่าที่ predecessor jobs ของขั้นตอนเสร็จสิ้นแล้ว
- คุณสามารถทำเครื่องหมายขั้นตอนแบบทำเองที่สมบูรณ์แล้วให้เป็นไม่สมบูรณ์ และรันขั้นตอนนั้นอีกครั้งได้ トラバเท่าที่ไม่มีขั้นตอนแบบอัตโนมัติที่ไม่สมบูรณ์อยู่ต่อจากนั้น
- คุณสามารถกำหนดให้ขั้นตอนรองกว่างานจะเสร็จสมบูรณ์ ก่อนที่จะแจ้ง ความสมบูรณ์ของงานได้ โดยการระบุ predecessor jobs ที่เหมือนกับ successor jobs ของขั้นตอนก่อนหน้านี
- คุณสามารถแจ้งให้ผู้ใช้ทราบเมื่อขั้นตอนเริ่มต้น, หยุด หรือเมื่อขั้นตอน ไม่ได้เริ่มตามเวลาที่ระบุ หรือเมื่อขั้นตอนให้เวลานานเกินไป ตัวอย่างเช่น คุณสามารถแจ้งให้ ผู้ใช้ที่รับผิดชอบขั้นตอนแบบทำเองหนึ่งทราบ ว่า ขั้นตอนแบบอัตโนมัติก่อนหน้านี เสร็จสมบูรณ์แล้ว

เมื่อคุณใช้เวิร์กโฟลว์ไฟล์บันทึก activity จะแสดงเวลาที่เวิร์กโฟลว์เริ่มต้น, ขั้นตอนที่รัน, สถานะของขั้นตอนแบบอัตโนมัติ (สำเร็จหรือล้มเหลว), เวลาที่เวิร์กโฟลว์สิ้นสุด และสถานะสุดท้ายของเวิร์กโฟลว์

ตารางที่ 8. ตัวอย่าง Work Flow

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 ขั้นตอนจะแพลิก successor job ที่ initialize ไฟล์ค่าจ้างเพื่อรัน แล้วทำให้เสร็จสมบูรณ์ขั้นตอน 2 มี successor job ของขั้นตอน 1 เป็น predecessor ดังนั้นขั้นตอน 2 จะคอยให้งานที่ initialize ไฟล์ค่าจ้างเสร็จสมบูรณ์ หลังจากเสร็จสมบูรณ์แล้ว ขั้นตอน 2 จะแจ้งให้ Clerk ทราบว่าเขาสามารถป้อนบัตรลงเวลาได้ ไม่มี successor jobs ที่จะแพลิกเพื่อรัน

Clerk จะทำขั้นตอน 3 ให้เสร็จสมบูรณ์ด้วยตนเองหลังจากที่ป้อนบัตรลงเวลาทั้งหมดแล้ว successor job ที่ประมวลผลไฟล์บัตรลงเวลาและพิมพ์รายงานบัตรลงเวลาจะถูกแพลิกเพื่อรัน ข้อควรระวังคือ Supervisor จะได้รับการแจ้งเตือน ถ้าขั้นตอนไม่เสร็จ

สิ้นภายใน 120 นาที เนื่องจาก predecessor job ของขั้นตอน 4 คือ successor ของขั้นตอน 3 ดังนั้นขั้นตอน 4 จะรอจนกว่างานที่ประมวลผลไฟล์บัตรลงเวลาและพิมพ์รายงานบัตรลงเวลาจะเสร็จสมบูรณ์

หลังจากที่งานเสร็จสมบูรณ์ Clerk จะได้รับแจ้งว่าสามารถตรวจสอบรายงาน บัตรลงเวลาได้แล้ว ไม่มี successor jobs ที่จะแฟล็กเพื่อรัน หลังจากการตรวจสอบรายงาน บัตรลงเวลา, Clerk จะทำขั้นตอน 5 ให้เสร็จสิ้นด้วยตนเอง successor job ที่ประมวลผลค่าจ้างและจัดทำเช็คจะถูกแฟล็กเพื่อรัน

เนื่องจาก predecessor job ของขั้นตอน 6 คือ successor ของขั้นตอน 5 ดังนั้น ขั้นตอน 6 จะรอจนกว่างานที่ประมวลผลค่าจ้างและจัดทำเช็คจะเสร็จ สมบูรณ์ หลังจากที่ยานเสร็จสมบูรณ์ จะมีการแจ้งให้ Clerk และ Supervisor ทราบว่า Payroll ได้เสร็จสมบูรณ์แล้ว ขณะนี้สามารถพิมพ์และแจกจ่ายเช็คได้

สำหรับข้อมูลเพิ่มเติมเกี่ยวกับ Work Flow Manager ให้ดูคำอธิบายทางออนไลน์

การสร้างเวิร์กโฟลว์ใหม่:

เมื่อคุณสร้างเวิร์กโฟลว์ใหม่ คุณจำเป็นต้องระบุวิธีการเริ่มเวิร์กโฟลว์, เวลาประมวลผลสูงสุด, ขั้นตอนภารกิจ และลำดับการรัน, การจัดตารางเวลา, การแจ้งเตือน และรายละเอียดของเอกสาร

ในการสร้างเวิร์กโฟลว์ใหม่ คุณต้องดำเนินขั้นตอนต่อไปนี้:

1. จาก System i Navigator ให้ขยาย **My Connections** → *your system* → **Work Management** → **Advanced Job Scheduler**
2. คลิกขวาที่ **Work Flow Manager** และเลือก **New Work Flow** หน้าต่าง The New Work Flow จะปรากฏขึ้น

สำหรับข้อมูลเพิ่มเติมเกี่ยวกับวิธีการทำหน้าต่าง New Work Flow ให้เสร็จสมบูรณ์ให้ดูคำอธิบายทางออนไลน์

หลังจากที่คุณตั้งค่าเวิร์กโฟลว์ของคุณแล้ว, คุณสามารถจัดการ เวิร์กโฟลว์ได้โดยการคลิกขวาบนชื่อเวิร์กโฟลส์ และคลิก **Work Flow Status**.

การเริ่มเวิร์กโฟลว์:

เมื่อคุณเริ่มเวิร์กโฟลว์ คุณสามารถเลือกได้ว่าคุณต้องการให้เวิร์กโฟลว์ เริ่มต้นบนลำดับแรกหรือบนลำดับที่ระบุ

ถ้าต้องการเริ่มเวิร์กโฟลว์ให้ทำตามขั้นตอนต่อไปนี้:

1. จาก System i Navigator, ให้ขยาย **Work Management** → **Advanced Job Scheduler** → **Work Flow Manager**คลิกขวาที่เวิร์กโฟลว์ และเลือก **Start**. หน้าต่าง Start Work Flow window จะปรากฏขึ้น
2. เลือกว่าคุณต้องการให้เวิร์กโฟลว์ เริ่มต้นบนลำดับแรกหรือบนลำดับที่ระบุ ถ้าคุณเลือกที่จะเริ่มที่ลำดับอื่นที่ไม่ใช่ลำดับแรก ขั้นตอนก่อนหน้านี้ทั้งหมดจะถูกทำเครื่องหมายว่าเสร็จสิ้นแล้ว

สำหรับข้อมูลเพิ่มเติมเกี่ยวกับหน้าต่าง Start Work Flow ให้ดูคำอธิบายทางออนไลน์.

การทำงานกับเวิร์กโฟลว์:

คุณสามารถควบคุมและมอนิเตอร์เวิร์กโฟลว์เมื่อเวิร์กโฟลว์รันได้โดยใช้ หน้าต่าง Work Flow Status

คุณสามารถเข้าใช้งานหน้าต่าง Work Flow Status window โดยขยาย **My Connections** → **ระบบของคุณ** → **Work Management** → **Advanced Job Scheduler** → **Work Flow Manager**. คลิกขวาที่เวิร์กโฟลว์ และเลือก **Status**.

- หน้าต่าง General แสดงสถานะปัจจุบันของเวิร์กโฟลว์
- หน้าต่าง Steps แสดงรายการของขั้นตอนทั้งหมดที่ได้กำหนดที่เวิร์กโฟลว์ ในปัจจุบัน
คุณสามารถดูได้ว่าขั้นตอนได้ถูกกำหนดเป็นแบบอัตโนมัติหรือแบบทำเอง ใดอย่างหนึ่ง และดูเวลาเริ่มต้นและสิ้นสุดขั้นตอน
 - เพื่อทำเครื่องหมายว่าขั้นตอนแบบทำเองได้เสร็จสมบูรณ์แล้ว ให้เลือกขั้นตอนที่ถูกต้องและทำเครื่องหมายเลือกกล่อง **Complete** .
 - ขั้นตอนแบบทำเองสามารถทำเครื่องหมายว่าเสร็จสมบูรณ์แล้วในลำดับใดๆ ก็ได้ ถ้า predecessor Advanced Job Scheduler jobs ทั้งหมดสำหรับขั้นตอนได้เสร็จสมบูรณ์แล้ว
 - ขั้นตอนแบบทำเองสามารถถูกทำเครื่องหมายว่า ไม่เสร็จสมบูรณ์ ถ้าไม่มีขั้นตอน แบบอัตโนมัติที่เสร็จสมบูรณ์อยู่ถัดไปในรายการ
 - สามารถเริ่มเวิร์กโฟลว์ด้วยตนเองได้ในทุกขั้นตอน การทำเช่นนี้จะข้ามขั้นตอนก่อนหน้านี้ ทั้งหมด ถ้าต้องการรีเฟรชรายการ ให้คลิก **Refresh**.
- หน้าต่าง Documentation แสดงข้อความเอกสารคู่มือ สำหรับเวิร์กโฟลว์

การมอนิเตอร์ activity งานด้วย Advanced Job Scheduler:

คุณสามารถใช้ Advanced Job Scheduler เพื่อดูประวัติ หรือสถานะของงาน หรือกลุ่มงาน คุณยังสามารถตั้งค่าการเก็บรักษา activity retention ซึ่งเป็นระยะเวลาที่คุณต้องการเก็บรักษาเรกคอร์ด activity สำหรับงาน

activity ของงานที่จัดตารางเวลาไว้:

activity ของงานที่จัดตารางเวลาไว้ช่วยให้คุณสามารถระบุว่าจะเก็บรักษาเรกคอร์ด 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:

ขั้นตอนเหล่านี้แสดงถึงวิธีระบุการเก็บรักษา activity

1. ขยาย **Work Management** จากหน้าต่าง System i Navigator ของคุณ
2. ขยาย **Advanced Job Scheduler**
3. คลิกขวาบน **Scheduled Job Activity** และคลิก **Properties**

การดูรายละเอียดของ activity งานที่จัดตารางเวลาไว้:

ขั้นตอนเหล่านี้แสดงถึงวิธีดูรายละเอียด activity งานที่จัดตารางเวลาไว้

1. ขยาย **Work Management** จากหน้าต่าง System i Navigator ของคุณ
2. ขยาย **Advanced Job Scheduler**.
3. ดับเบิลคลิกที่ **Scheduled Job Activity**.

การดู activity งานที่จัดตารางเวลาไว้สำหรับงานเฉพาะ:

ขั้นตอนเหล่านี้แสดงถึงวิธีดู activity งานที่จัดตารางเวลาไว้สำหรับงานเฉพาะ

1. ขยาย **Work Management** จากหน้าต่าง System i Navigator ของคุณ
2. ขยาย **Advanced Job Scheduler**.
3. คลิก **Scheduled jobs**.
4. คลิกขวาที่ **Job Name** ที่มี activity ที่คุณต้องการจะแสดง และคลิก **Activity**.

การดูรายละเอียดของไฟล์บันทึก activity:

ไฟล์บันทึก activity แสดง activity ที่เกิดขึ้นภายในตัวจัดตารางเวลา เช่น การเพิ่มงานในตาราง, การเปลี่ยนงานในตาราง, หรือ การส่ง การละเมิดความปลอดภัย, ลำดับการประมวลผลโดยงานที่ถูกจัดตารางเวลาไว้ และความผิดพลาดต่างๆ ที่เกิดขึ้นจะถูกแสดงให้เห็น วันที่และเวลาสำหรับ activity ก่อนหน้านี้จะถูกแสดงไว้ด้วย

เพื่อดูรายละเอียดของข้อความ ให้ดับเบิลคลิกวันที่และเวลา ถ้าต้องการดูรายละเอียดไฟล์บันทึก activity ให้ทำตามขั้นตอนเหล่านี้:

1. ขยาย **Work Management** จากหน้าต่าง System i Navigator ของคุณ
2. ขยาย **Advanced Job Scheduler**.
3. คลิก **Activity Log**. entry ของวันที่ปัจจุบันจะแสดงขึ้น เพื่อเปลี่ยนเกณฑ์การเลือก ให้เลือก **Include** จากเมนู Options

การดูไฟล์บันทึก activity สำหรับงานเฉพาะ:

ขั้นตอนเหล่านี้แสดงวิธีดูไฟล์บันทึก activity สำหรับงานเฉพาะ

1. ขยาย **Work Management** จากหน้าต่าง System i Navigator ของคุณ
2. ขยาย **Advanced Job Scheduler**.
3. คลิก **Scheduled jobs**.
4. คลิกขวา **Job Name** ที่มีไฟล์บันทึก activity ที่คุณต้องการแสดง และคลิก **Activity log**.

คุณยังสามารถใช้หน้า **Last Run** ของ คุณสมบัติของงานเพื่อดูความก้าวหน้าของงานได้ด้วย ระบุ Set Step ด้วยการใช้คำสั่ง Job Scheduler (SETSTPJS) ก่อนหน้า หรือหลังจากขั้นตอนในโปรแกรม CL พร้อมกับ description ที่ระบุความก้าวหน้าของงาน เมื่องานดำเนินไปถึงคำสั่ง SETSTPJS ในโปรแกรม description ที่เกี่ยวข้องจะถูกแสดงขึ้นในหน้า Last Run และบนอุปกรณ์ไร้สาย

การมอนิเตอร์ข้อความด้วย Advanced Job Scheduler:

แต่ละคำสั่งในรายการคำสั่งของงานงานหนึ่งสามารถมี identifier ข้อความเพื่อการมอนิเตอร์ได้หลายตัว เมื่องานรันและมีการออกข้อความแสดงความผิดพลาดที่ตรงกับข้อความใดข้อความหนึ่งที่ป้อนสำหรับคำสั่งที่เลือก งานจะลือกข้อผิดพลาด แต่ยังคงประมวลผลต่อไปด้วยคำสั่งถัดไปใน รายการ

ถ้าเลขศูนย์ถูกระบุไว้ในสองตำแหน่งขวาสุดหรือทั้งสี่ตำแหน่งขวาสุด อย่างใดอย่างหนึ่ง, เช่น ppmm00, นั่นคือการระบุ identifier ข้อความแบบทั่วไป. ตัวอย่าง ถ้า CPF0000 ถูกระบุไว้ หมายความว่าข้อความที่ขึ้นต้นด้วย CPF ทั้งหมดจะถูกมอนิเตอร์

ถ้าต้องการใส่เพิ่ม identifier ข้อความลงในคำสั่ง ให้ทำตามขั้นตอนเหล่านี้:

1. ขยาย **Work Management** จากหน้าต่าง System i Navigator ของคุณ
2. คลิกขวาที่ **Advanced Job Scheduler**.
3. คลิก **Scheduled Jobs** เพื่อแสดงรายการของงาน
4. คลิกขวาบนงานที่จัดตารางเวลาและคลิก **Properties**.
5. เลือกคำสั่งจากรายการ และคลิก **Properties**.
6. คลิก **Messages**.
7. ใส่ identifier ข้อความที่จะมอนิเตอร์และคลิก **Add**.

การสร้าง และทำงานกับพื้นที่ข้อมูลแบบโลคัล:

พื้นที่ข้อมูลแบบโลคัลคือส่วนของพื้นที่ว่างที่ถูกจัดสรรสำหรับ งาน ไม่ทุกงานที่ใช้พื้นที่ข้อมูลแบบโลคัลของตน แต่มีเพียงบางงานเท่านั้น แต่ละคำสั่งภายใน งานมีสิทธิเข้าใช้งานพื้นที่ข้อมูลแบบโลคัลของงาน คุณอาจต้องการใช้พื้นที่ข้อมูลแบบโลคัล ถ้าคุณกำลังจัดตารางเวลางาน ซึ่งก่อนหน้านี้กำหนดให้คุณต้องระบุ พารามิเตอร์เพิ่มเติมด้วยตนเอง ใช้พื้นที่ข้อมูลแบบโลคัลเพื่อระบุพารามิเตอร์เพิ่มเติม เพื่อให้คุณไม่ต้องระบุพารามิเตอร์ด้วยตนเองในทุกครั้งที่งานเริ่มขึ้น

ถ้าต้องการระบุข้อมูลพื้นที่ข้อมูลแบบโลคัลสำหรับงานที่จัดตารางเวลา ให้ทำ ตามขั้นตอนเหล่านี้:

1. ขยาย **Work Management** จากหน้าต่าง System i Navigator ของคุณ
2. ขยาย **Advanced Job Scheduler** → **Scheduled Jobs**.
3. คลิกขวาบนงานและคลิก **Properties**.
4. แก้ไขหน้าต่าง Local Data Area ตามความจำเป็น

อ้างอิงคำอธิบายทางออนไลน์สำหรับข้อมูลเพิ่มเติม เมื่อคุณกรอกรายละเอียด สำหรับพื้นที่ข้อมูลแบบโลคัล

การสร้าง และใช้งานควบคุมแอ็พพลิเคชัน และการควบคุมงาน:

แอ็พพลิเคชันคืองานหลายๆ งานที่ถูกรวมเป็นกลุ่มไว้เพื่อการประมวลผล มีความหมายกว้างกว่ากลุ่มของงานที่ไม่มีความจำเป็นต้องประมวลผลตามลำดับ งานหลายงานในแอ็พพลิเคชันต่างๆ สามารถประมวลผลไปพร้อมกันได้และงานแต่ละงานไม่ต้องรอให้งานอื่นประมวลผลเสร็จก่อน เราสามารถทำงานกับทุกๆ งานภายในแอ็พพลิเคชันและแต่ละงานสามารถมีชุดของค่าดีฟอลต์ของตัวเอง ตัวควบคุมงานทั้งหลายคือค่าดีฟอลต์ที่ถูกกำหนดลงไปในงานในขณะที่คุณเพิ่มงานลงไปในตารางเวลางาน เช่นเดียวกับค่าดีฟอลต์ที่ถูกใช้เมื่อมีการส่งงาน

แอ็พพลิเคชัน คืองานที่ถูกจัดกลุ่มเข้าด้วยกันสำหรับการประมวลผล ตัวอย่างเช่น คุณอาจมีชุดของงานที่คุณใช้สำหรับระบบเงินเดือนที่คุณต้องการจัดกลุ่มเข้าด้วยกันสำหรับประมวลผลทางการบัญชี

ตัวควบคุมงาน คือค่าดีฟอลต์ที่ถูกกำหนดให้กับงานหนึ่งในขณะที่คุณเพิ่มมันลงไปในการตารางงาน เช่นเดียวกับการใช้ค่าดีฟอลต์เมื่อมีการส่งงานให้เข้าไปประมวลผล ตัวควบคุมงาน ดีฟอลต์ได้แก่ ปฏิทิน, ปฏิทินวันหยุด, คิวงาน, รายละเอียดของงาน และอื่นๆ

คุณสามารถแสดงแอ็พพลิเคชันที่มีอยู่ หรือตัวควบคุมงานบนระบบของคุณ คุณสามารถเพิ่มตัวควบคุมแอ็พพลิเคชัน หรือตัวควบคุมงานใหม่, เพิ่มตัวควบคุมแอ็พพลิเคชัน หรือตัวควบคุมงานใหม่ โดยใช้ฐานเดิมที่มีอยู่ หรือลบตัวควบคุมแอ็พพลิเคชัน หรือตัวควบคุมงาน คุณยังสามารถเลือกแอ็พพลิเคชัน หรือตัวควบคุมงาน แล้วแสดงผลคุณสมบัติของมันเพื่อทำการเปลี่ยนแปลงได้

ถ้าต้องการสร้างแอ็พพลิเคชัน/ตัวควบคุมงานใหม่ให้ทำตามขั้นตอนดังต่อไปนี้:

1. ขยาย **Work Management** จาก System i Navigator หน้าต่างของคุณ
2. คลิกขวา **Advanced Job Scheduler** และคลิก **Properties**.
3. คลิกแท็บ **Applications/ Job Controls**.
4. คลิก **New** และป้อนชื่อแอ็พพลิเคชัน
5. พิมพ์คำอธิบายสำหรับแอ็พพลิเคชัน
6. ให้เลือกคนติดต่อสำหรับแอ็พพลิเคชัน คนติดต่อคือ ชื่อของผู้ใช้ที่คุณติดต่อกับถ้าคุณมีปัญหาเกี่ยวกับงานที่อยู่ในแอ็พพลิเคชัน คุณสามารถระบุคนติดต่อได้สูงสุดถึง 5 คนต่อแอ็พพลิเคชัน คุณยังสามารถเลือกที่จะใส่เพิ่มหรือลบคนติดต่อออกจากรายชื่อคนติดต่อ
7. คุณอาจพิมพ์ข้อมูลเพิ่มเติมเพื่อช่วยให้คุณระบุถึงแอ็พพลิเคชันได้ ข้อมูลนั้นจะเชื่อมโยงกับ แอ็พพลิเคชันใหม่ ข้อมูลนี้อาจเป็นประโยชน์หากเกิดปัญหาใดๆ ขึ้น

การทำงานกับการแจ้งเตือน:

ภายในการแจ้งเตือน คุณสามารถทำภารกิจต่างๆ ได้ การแจ้งเตือนอนุญาตให้คุณระบุคุณสมบัติของผู้รับและคุณสมบัติรายการการแจกจ่ายรายงาน ยิ่งกว่านั้น คุณสามารถส่งอีเมล และจัดลำดับการส่งต่อข้อความ ในกรณีที่ผู้รับไม่มีการตอบกลับภายในเวลาที่กำหนด

ก่อนที่คุณจะสามารถส่งข้อความอีเมลได้ คุณต้องระบุเมลเซิร์ฟเวอร์ที่จะใช้สำหรับการแจ้งเตือน

ต่อไปนี้เป็นฟังก์ชันการแจ้งเตือนที่สำคัญของ **Advanced Job Scheduler**:

ผู้รับ เมื่อจัดตารางเวลางาน คุณสามารถระบุว่าการให้ส่งข้อความการแจ้งเตือน ให้แก่ผู้รับที่ระบุหรือไม่ คุณสามารถส่งข้อความแจ้งเตือนถ้าหากงานล้มเหลว, งานเสร็จสมบูรณ์ หรืองานนั้นไม่เริ่มทำงานภายในเวลาที่กำหนด สำหรับผู้รับแต่ละคน คุณต้องกำหนดคุณสมบัติของผู้รับ คุณสามารถเข้าถึงคุณสมบัติของผู้รับได้โดยการเลือก **Advanced Job Scheduler** → **Notification** → **Recipients** แล้วเลือกผู้รับจากรายชื่อของผู้รับ

รายการการแจกจ่ายรายงาน

ให้ใช้รายการการแจกจ่ายรายงานในการระบุรายการของสพูลไฟล์ที่อนุญาตให้แจกจ่ายได้ แต่ละสพูลไฟล์ที่เกิดจากงานจะถูกตรวจว่ามีรายชื่อที่ตรงกันอยู่ในรายการสพูลไฟล์หรือไม่ ถ้าตรงกัน ผู้รับที่เกี่ยวข้องกับสพูลไฟล์นั้นจะได้รับสำเนาของสพูลไฟล์ผ่านทาง อีเมล, สำเนาของสพูลไฟล์ในเอาต์พุตคิว หรือทั้งสองอย่าง คุณสามารถเข้าใช้งานรายการการแจกจ่ายรายงานโดยการเลือก **Advanced Job Scheduler** → **Notification** → **Report distribution list**.

อีเมล คุณสามารถส่งข้อความอีเมลไปยังผู้รับที่ถูกระบุอยู่ในรายการผู้รับหรืออาจส่งไปยังอีเมลแอดเดรสที่เฉพาะ

เจาะจงก็ได้ ในคุณสมบัติของผู้รับ ต้องมีอีเมลแอดเดรสที่ต้องการส่งข้อความไป ตอนที่ส่งข้อความอีเมล คุณสามารถแนบสพูลไฟล์ไปด้วย คุณสามารถส่งสพูลไฟล์ที่มีรูปแบบเป็น PDF ยิ่งไปกว่านั้น คุณสามารถระบุรายการการส่งต่อที่จะใช้ ถ้าผู้รับ ไม่ตอบสนองภายในช่วงเวลาที่จะระบุ

การระบุสพูลไฟล์ที่จะแนบไปกับอีเมล:

ถ้าต้องการระบุสพูลไฟล์ที่จะแนบไปกับอีเมล ให้ทำดังต่อไปนี้:

1. ขยาย **Basic Operations** จากหน้าต่าง System i Navigator ของคุณ
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 System i Navigator ของคุณ
2. คลิก **Advanced Job Scheduler** → **Notification** → **E-mail** → **Sent**.
3. คลิกขวาที่ข้อความการส่งต่อ และคลิก **Stop**.

หมายเหตุ: ถ้าต้องการดูเฉพาะข้อความการส่งต่อ ให้เลือก **View** → **Customize this view** → **Include** จาก System i Navigator หน้าต่าง จากนั้น ในฟิลด์ **Type** เลือก **Escalating**.

การทำงานกับรายชื่อไลบรารี:

รายชื่อไลบรารีคือรายชื่อของไลบรารีที่ผู้ใช้เป็นผู้กำหนดเพื่อให้ตัวจัดลำดับงานใช้เมื่อมีการประมวลผลงาน

รายชื่อไลบรารี เป็นรายชื่อของไลบรารีที่ผู้ใช้กำหนดเองที่ถูกใช้งานโดยตัวจัดลำดับงานระดับสูง เพื่อค้นหาข้อมูลที่ต้องการในขณะกำลังประมวลผล คุณสามารถแสดงรายชื่อไลบรารี, ใส่เพิ่มรายชื่อไลบรารีใหม่, ใส่เพิ่มรายชื่อไลบรารีใหม่โดยใช้ฐานเดิมจากไลบรารีที่มีอยู่แล้ว หรือลบรายชื่อไลบรารีออก, ถ้ารายชื่อไลบรารีไม่ได้ถูกใช้อยู่โดยงานที่จัดตารางเวลาในปัจจุบัน

คุณสามารถเลือกรายชื่อ และแสดงผลคุณสมบัติของมันเพื่อทำการเปลี่ยนแปลง คุณสามารถใส่ไลบรารีได้สูงสุดถึง 250 ไลบรารีบนรายชื่อไลบรารี

ถ้าต้องการใส่เพิ่มรายชื่อไลบรารีใหม่ ให้ทำตามขั้นตอนเหล่านี้:

1. ขยาย **Work Management** จากหน้าต่าง System i Navigator ของคุณ
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 งานที่ถูกส่งผ่านตัวจัดลำดับงาน ตัวอย่างของตัวแปรคำสั่งได้แก่ การเริ่มต้นแต่ละเดือน, หมายเลขฝ่าย, รหัสบริษัท, และอื่นๆ

ตัวแปรคำสั่ง (ที่เดิมเรียกกันว่า พารามิเตอร์) คือตัวแปรที่คุณเก็บอยู่ในตัวจัดลำดับงาน และใช้ในงานที่ถูกส่งผ่านไปยังตัวจัดลำดับงานระดับสูง ตัวแปรคำสั่งประกอบด้วยข้อมูลที่จะถูกแทนที่ภายในสตริงคำสั่งของงานที่กำหนดตารางเวลาไว้ ตัวอย่างของตัวแปรคำสั่งได้แก่ การเริ่มต้นของแต่ละเดือน, หมายเลขแผนกของบริษัท, หมายเลขบริษัท และอื่นๆ คุณสามารถแสดงผลตัวแปรคำสั่ง, ใส่เพิ่มตัวแปรคำสั่งใหม่, ใส่เพิ่มตัวแปรคำสั่งใหม่โดยใช้ฐานเดิมจากตัวแปรคำสั่งที่มีอยู่แล้ว, หรือลบตัวแปรคำสั่งออก, ถ้าตัวแปรไม่ได้ถูกใช้งานในปัจจุบันโดยงานที่จัดตารางเวลา

คุณสามารถเลือกตัวแปรคำสั่งที่มีอยู่แล้วและแสดงผลคุณสมบัติของมันเพื่อทำการเปลี่ยนแปลง

ถ้าต้องการใส่เพิ่มตัวแปรคำสั่ง ให้ทำตามขั้นตอนเหล่านี้:

1. ขยาย **Work Management** จากหน้าต่าง System i Navigator ของคุณ
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. ให้เลือกว่าคุณต้องการดึงข้อมูลค่าทดแทนออกมาจากผู้ควบคุมระบบตอนรันไทม์หรือไม่

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:

ก่อนรัน Advanced Job Scheduler for Wireless ต้องมั่นใจว่าคุณปฏิบัติตามข้อกำหนดฮาร์ดแวร์ และซอฟต์แวร์แล้ว

องค์ประกอบต่อไปนี้เป็นสิ่งจำเป็นสำหรับการรัน Advanced Job Scheduler for Wireless:

- ตัว Advanced Job Scheduler (5761-JS1) โลเซ็นส์โปรแกรม : ตัว Advanced Job Scheduler ผลิตภัณฑ์ที่มี Advanced Job Scheduler for Wireless
- อุปกรณ์ที่ใช้รันฟังก์ชัน
 - โทรศัพท์ที่พร้อมเชื่อมต่ออินเทอร์เน็ตที่มีการบริการอินเทอร์เน็ตไร้สาย
 - PDA พร้อมเว็บเบราว์เซอร์, โมเด็มไร้สาย และบริการอินเทอร์เน็ตไร้สาย
 - เวิร์กสเตชันที่มีเว็บเบราว์เซอร์แบบตั้งเดิม
- ระบบที่รัน i5/OS V5R3 หรือหลังจากนั้นในเน็ตเวิร์ก TCP/IP
- เว็บแอปพลิเคชันเซิร์ฟเวอร์ที่รันบนระบบศูนย์กลาง ซึ่งได้แก่สิ่งต่อไปนี้:
 - ASF Jakarta Tomcat Application server
 - แอปพลิเคชันเซิร์ฟเวอร์อื่นที่รันบนระบบศูนย์กลาง ที่มีความสามารถ เป็น host servlets
- HTTP Server installed on the system
- ระบุเซิร์ฟเวอร์ HTTP ของคุณพร้อมด้วยคุณลักษณะพิเศษไร้สาย Advanced Job Scheduler ในการทำเช่นนี้ให้เชื่อมต่อระบบที่มี Advanced Job Scheduler ติดตั้งอยู่โดยการใช้อินเทอร์เน็ตเฟสแบบอักขระ จากนั้น ระบุคำสั่งต่อไปนี้:

CALL QIJS/QIJSINT

การเลือกอุปกรณ์:

โทรศัพท์ที่พร้อมเชื่อมต่ออินเทอร์เน็ต และ PDAs ไร้สายเป็นเทคโนโลยีที่เปลี่ยนแปลงอย่างรวดเร็วมาก เทคโนโลยีพวกนี้แตกต่างกันในด้านขนาดจอภาพ, ในด้านอินเทอร์เน็ตเฟสกับผู้ใช้ และในลักษณะที่สำคัญอื่นๆอีกมากมาย ข้อมูลในหัวข้อนี้ช่วยให้คุณเลือกอุปกรณ์ที่เข้ากันได้กับ Advanced Job Scheduler for Wireless อุปกรณ์ไร้สายอื่นๆอาจจะทำงานเข้ากันได้ด้วย ถ้าอุปกรณ์นั้นสนับสนุนการบราวซิ่งอินเทอร์เน็ตแบบไร้สาย แต่การโต้ตอบอาจแตกต่างกัน

โทรศัพท์ที่พร้อมเชื่อมต่ออินเทอร์เน็ต: เลือกโทรศัพท์ที่พร้อมเชื่อมต่ออินเทอร์เน็ตเพื่อใช้กับ Advanced Job Scheduler for Wireless

PDAs: เลือก PDA เพื่อใช้กับ Advanced Job Scheduler for Wireless

PCs: คุณยังสามารถใช้เว็บเบราว์เซอร์แบบตั้งเดิมกับ Advanced Job Scheduler for Wireless ได้ด้วย

การตั้งค่าสถานะแวดล้อมไร้สายของคุณ:

เพื่อให้มั่นใจว่า Advanced Job Scheduler for Wireless รันอย่างเหมาะสม คุณจำเป็นต้องดัดแปลง web application server ของคุณ และการตั้งค่าไฟล်วอลล์

ก่อนที่คุณจะเริ่มใช้ Advanced Job Scheduler for Wireless ตรวจสอบให้แน่ใจว่าคุณได้ตั้งค่าอย่างถูกต้องหรือตั้งค่าไอเท็มดังต่อไปนี้แล้ว:

1. ตั้งค่า Web application server ของคุณ แล้วกำหนดให้ Advanced Job Scheduler for Wireless รันด้วยการใช้ ASF Jakarta Tomcat servlet engine คำสั่งเครื่องเหล่านี้จะระบุวิธีการสร้าง และเริ่มเว็บแอปพลิเคชันเซิร์ฟเวอร์ของคุณ นอกจากนี้ คำสั่งยังระบุโปรแกรมที่คุณต้องรันก่อนการทำงานกับฟังก์ชันไร้สายของ Advanced Job Server ด้วย

2. ตั้งค่าไฟล်วอลล์ของคุณ: เมื่อคุณใช้ System i Navigator กับ Wireless คุณต้องเข้าสู่ระบบของคุณจากอินเทอร์เน็ต หากคุณมีไฟล်วอลล์ คุณอาจต้องดัดแปลงการเชื่อมต่อไฟล်วอลล์ของคุณเพื่อรัน System i Navigator สำหรับ Wireless.
3. เลือกภาษา: ภาษาดีฟอลต์ตั้งไว้ที่ภาษาอังกฤษ แต่คุณสามารถตั้งค่าอุปกรณ์ของคุณเพื่อแสดงภาษาที่คุณเลือก

หลังจากที่คุณทำขั้นตอนเหล่านี้เสร็จสิ้นแล้ว คุณก็พร้อมที่จะเชื่อมต่อกับ เซิร์ฟเวอร์ของคุณ และเริ่มต้นโดยใช้ Advanced Job Scheduler for Wireless

การตั้งค่าเว็บแอ็พพลิเคชันเซิร์ฟเวอร์ของคุณ:

ก่อนการทำงานกับ Advanced Job Scheduler for Wireless คุณต้องเริ่มต้นและตั้งค่าเว็บแอ็พพลิเคชันเซิร์ฟเวอร์ ขั้นตอนต่อไปนี้เป็น การตั้งค่า ASF Tomcat servlet engine สำหรับเซิร์ฟเวอร์ HTTP (สนับสนุนโดย Apache) เพื่อรัน Advanced Job Scheduler for Wireless

ข้อกำหนด

ก่อนคุณ initialize คุณต้องมีสิทธิ์ QSECOFR และ IBM HTTP Server for i5/OS (5761-DG1) ติดตั้งไลเซนส์โปรแกรมไว้แล้ว:

หมายเหตุ: วิธีการต่อไปนี้จะสร้าง instance ของเซิร์ฟเวอร์ HTTP อันใหม่ขึ้นมา คุณไม่สามารถใช้วิธีการต่อไปเพื่อตั้งค่า Advanced Job Scheduler บนเซิร์ฟเวอร์ HTTP ที่มีอยู่แล้ว

เริ่มการทำงานของ Advanced Job Scheduler for Wireless บนเซิร์ฟเวอร์ HTTP Server

การรันคำสั่งต่อไปนี้จะใส่เพิ่ม Advanced Job Scheduler for Wireless servlet ที่ Apache Software Foundation Jakarta Tomcat servlet engine และยังคงตั้งค่า IBM เซิร์ฟเวอร์ HTTP (สนับสนุนโดย Apache) ที่มีชื่อว่า Advanced Job SchedulerP ซึ่งคอยรอรับคำขอบนพอร์ต 8210

ก่อนการทำงานกับ Advanced Job Scheduler for Wireless คุณต้อง initialize Advanced Job Scheduler for Wireless บน HTTP server instance บนระบบของคุณ เพื่อทำเช่นนี้, ให้ระบุคำสั่งต่อไปนี้จากอินเทอร์เน็ตเฟสแบบอักษร

CALL QIJS/QIJSINT

คำสั่งนี้จะรันโปรแกรมที่ให้มาพร้อมกับระบบของคุณ

หลังจากที่คุณตั้งค่าเว็บแอ็พพลิเคชันเซิร์ฟเวอร์และ initialize Advanced Job Scheduler instance บนเว็บแอ็พพลิเคชันเซิร์ฟเวอร์แล้ว คุณสามารถตั้งค่า สภาวะแวดล้อม Advanced Job Scheduler wireless ต่อไปได้

การเลือกภาษา:

เมื่อคุณเชื่อมต่อกับ Advanced Job Scheduler for wireless คุณสามารถ ระบุภาษาที่จะใช้ได้ หากคุณไม่ต้องการระบุภาษา เฉพาะ คุณสามารถเชื่อมต่อกับระบบของคุณต่อไปได้

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

การเชื่อมต่อไปยัง i5/OS ระบบปฏิบัติการ:

คุณสามารถใช้อุปกรณ์ไร้สายของคุณเพื่อเชื่อมโยงไปยังระบบที่บรรจุผลิตภัณฑ์ Advanced Job Scheduler product อยู่

เพื่อเริ่มโดยใช้ Advanced Job Scheduler for Wireless ให้ระบุ URL ของระบบของคุณลงในอุปกรณ์ไร้สายของคุณ เมื่อใช้อุปกรณ์ของคุณไปที่ URL บนระบบของคุณ ให้ใช้รูปแบบดังต่อไปนี้ ตรวจสอบให้แน่ใจว่าการพิมพ์จุดสิ้นสุดของ URL (/servlet/AdvancedJobSchedulerPervasive) อย่างแม่นยำดังที่แสดงต่อไปนี้:

โฮสต์. โดเมน: พอร์ต/servlet/AdvancedJobSchedulerPervasive

โฮสต์: System i ชื่อโฮสต์ โดเมน: โดเมนที่โฮสต์ตั้งอยู่ พอร์ต: พอร์ตที่ instance ของเว็บเซิร์ฟเวอร์กำลังเฝ้าฟังอยู่ ค่าดีพอลต์คือ 8210

ถ้าต้องการระบุภาษาเฉพาะที่ใช้ให้ดู การเลือกภาษา.

โทรศัพท์ที่พร้อมเชื่อมต่ออินเทอร์เน็ต และโครงร่างบราวเซอร์ของ PDA

ถ้าคุณเชื่อมต่อกับคุณลักษณะ Advanced Job Scheduler for Wireless สำเร็จแล้วบนระบบ จอแสดงผลเริ่มต้นจะมีข้อมูลสรุปเกี่ยวกับโทรศัพท์ที่พร้อมเชื่อมต่ออินเทอร์เน็ต หรือ PDA ของคุณ ข้อมูลสรุประบุถึงข้อมูลที่เป็นอยู่ในปัจจุบัน, มีงานที่จัดตารางเวลาอยู่จำนวนเท่าไร, มี activity entry อยู่จำนวนเท่าไร และอัปเดตการตรวจสอบสถานะของการมอนิเตอร์งานหรือการส่งข้อความไปยังผู้รับ นอกจากนี้ ข้อสรุปยังมีสถานะโดยรวมทั้งหมดของ OK หรือ Attention ปรากฏบนส่วนบนสุดของจอแสดงผล ถ้ามีการระบุ Attention งานจะมีข้อความที่ต้องการความสนใจมากขึ้น งานที่ต้องการความสนใจจะประกอบด้วยเครื่องหมายอุทธาน


โครงร่างบราวเซอร์แบบดั้งเดิม

โครงร่างบราวเซอร์แบบดั้งเดิมจะเหมือนกันทุกประการกับจอแสดงผลของโทรศัพท์ที่พร้อมเชื่อมต่ออินเทอร์เน็ต และ PDA อย่างไรก็ตาม จำนวนของเนื้อหาจะน้อยกว่าขนาดของจอแสดงผล ดังนั้น คุณสามารถลดขนาดของเว็บบราวเซอร์ลงเพื่อจะได้เพิ่มเนื้อที่มากขึ้นในการทำงานกับแอปพลิเคชันอื่นๆ ในขณะที่ยังคงเปิด Advanced Job Scheduler for Wireless Web browser ได้อยู่ นอกจากนี้ ถ้าคุณกำลังใช้งานอินเทอร์เน็ตบราวเซอร์แบบเดิมบนเครื่องพีซีของคุณ คุณสามารถเลือกใช้ **Show all** จากเมนูหลักของ Advanced Job Scheduler แล้วคุณก็สามารถเรียกดูเนื้อหาที่มากขึ้นในหน้าเว็บเพจหน้าเดียว

หลังจากที่คุณได้เชื่อมต่อกับระบบของคุณเสร็จเรียบร้อยแล้ว คุณอาจต้องการปรับการเชื่อมต่อของคุณตามความต้องการ

การปรับการเชื่อมต่อตามความต้องการของคุณ:

โดยใช้อุปกรณ์ไร้สายของคุณ, คุณสามารถปรับอินเทอร์เน็ตเฟสได้ตามความต้องการที่เฉพาะเจาะจงของคุณ ตัวอย่างเช่น คุณอาจต้องการดูงานเพียงบางงาน และระบุว่าไม่ต้องการดูชื่อกลุ่มของงาน และคุณอาจไม่ต้องการเข้าถึงรายการของ activity ที่จัดตารางเวลา เพจการปรับตามความต้องการบนอุปกรณ์ไร้สายของคุณอนุญาตให้คุณกรองงานต่างๆรวมทั้งการเปลี่ยนจอแสดงผลตามที่ชื่นชอบ

คุณสามารถปรับการเชื่อมต่อของคุณตามความต้องการได้หลายวิธี ไม่ว่าคุณกำลังใช้ PC, PDA หรือโทรศัพท์ที่พร้อมเชื่อมต่ออินเทอร์เน็ต เพื่อใช้ประโยชน์จากคุณลักษณะพิเศษเหล่านี้ ให้ดูที่ Job Scheduler for i5/OS  Web site.

การจัดการ Advanced Job Scheduler for Wireless:

คุณสามารถใช้อุปกรณ์ไร้สายของคุณในการทำงานกับ Advanced Job Scheduler

ในการใช้อุปกรณ์ไร้สาย คุณลักษณะพิเศษที่พร้อมใช้งานมีดังต่อไปนี้:

การดูงานที่แอ็คทีฟ, พักไว้ และงานที่ค้างคั่ง

คุณสามารถดูรายการของงานปกติ (งาน Advanced Job Scheduler) หรืองาน Management Central ที่มีสถานะเป็นแอ็คทีฟ, พักไว้ หรือค้างคั่ง คุณยังสามารถปรับการแสดงรายการของงานโดยให้เรียงลำดับตามชนิดของงาน, ชื่อ หรือเวลา ยิ่งไปกว่านั้น คุณสามารถระบุได้ว่าจะให้ไลบรารีข้อมูลใดเก็บข้อมูลสำหรับ 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

การแสดงความ

ถ้างานมีข้อความที่รอกงานอยู่ คุณสามารถดูข้อความ และตอบกลับข้อความได้โดยใช้อุปกรณ์ไร้สายของคุณ

การเริ่มงาน

คุณสามารถใช้อุปกรณ์ไร้สายของคุณเพื่อส่งงาน อี้อพชั่นที่คุณสามารถใช้เมื่อ ส่งงานขึ้นอยู่กับอุปกรณ์ไร้สายที่คุณใช้

การทำงานกับ Advanced Job Scheduler activity

คุณสามารถโต้ตอบกับ Advanced Job Scheduler activity จากอุปกรณ์ไร้สายได้ แต่ละ activity มีอี้อพชั่นที่ต่างกััน ขึ้นอยู่กับสถานะของ activity entry

Internationalization

Advanced Job Scheduler for Wireless ใช้โค้ดประเทศ และภาษาที่เชื่อมโยงกับ System i^(TM) Java^(TM) Virtual Machine ของคุณเพื่อกำหนดภาษา แลรูปแบบของวันที่/เวลาที่ใช้นบนอุปกรณ์ไร้สายของคุณ ถ้าค่าดีฟอลต์ Java Virtual Machine ไม่ใช่โค้ดที่คุณต้องการ ใช้ คุณสามารถเปลี่ยนโค้ดได้โดยง่าย ให้ดูคำอธิบายทางออนไลน์สำหรับรายละเอียดเพิ่มเติม

ให้ดูคำอธิบายออนไลน์สำหรับรายละเอียดเพิ่มเติมเกี่ยวกับการทำภารกิจที่ต้องการ

การแก้ปัญหาเกี่ยวกับ Advanced Job Scheduler:

เมื่องานไม่รันตามเวลาที่จัดตารางเวลาไว้ วิธีแก้ปัญหาเหล่านี้สามารถช่วยคุณพบสิ่งที่คุณสามารถทำได้

เพื่อแก้ปัญหาเกี่ยวกับ Advanced Job Scheduler ชั้นแรกให้ดูเพจ Frequently Asked Questions ที่เว็บไซต์ Job Scheduler

สำหรับ i5/OS  อ่านคำถามที่ถามเป็นประจำเกี่ยวกับวิธีการใช้งานฟังก์ชันเฉพาะด้วย Advanced Job Scheduler

นอกจากนั้น ต่อไปนี้คือรายการของไอเท็มที่คุณสามารถตรวจสอบได้ เมื่อมีงานที่ไม่รันตามเวลาที่ได้ถูกจัดตารางเวลาไว้:

ระดับการฟิซปัจจุบัน

สิ่งแรกที่คุณควรตรวจสอบคือโปรแกรมฟิซของคุณควรจะเป็นตัวล่าสุด เมื่อคุณทำการร้องขอโปรแกรมฟิซ ขอให้มั่นใจว่าคุณได้ร้องขอรายชื่อของโปรแกรมฟิซทั้งหมด ฟิซไม่ทั้งหมดถูกรวมอยู่ใน Cumulative PTF packages.

ตรวจสอบการมอนิเตอร์งาน

- งาน QIJSSCD ควรจะแอนด์ที่พอย์ในระบบย่อย QSYSWRK ถ้าไม่อยู่ในระบบย่อย ให้ใช้คำสั่ง Start Job Scheduler (STRJS)
- การมอนิเตอร์งานอาจวิ่งวนซ้ำถ้าสถานะของงานเป็น RUN นานเกินสิบนาที ถ้าการมอนิเตอร์วนซ้ำ ให้สั่งจบงานด้วย *IMMED แล้วเริ่มการมอนิเตอร์งานอีกครั้ง (STRJS)
- ถ้ามีข้อความที่จะต้องตอบ ให้ตอบกลับด้วย C (Cancel) การมอนิเตอร์งานจะหน่วงไว้ 90 วินาที แล้วจึงเริ่มการมอนิเตอร์อีกครั้ง พิมพ์บันทึกการใช้งานของงานการมอนิเตอร์ บันทึกการใช้งานนี้จะมีข้อความแสดงความผิดพลาด

การตรวจสอบไฟล์บันทึก Advanced Job Scheduler

ใช้คำสั่ง Display Log for Job Scheduler (DSPLOGJS) สำหรับงาน กด F18 เพื่อไปที่จุดสิ้นสุดของรายการ จะมี entry ให้เห็นเพื่ออธิบายว่าเพราะเหตุใดงานจึงไม่รัน ตัวอย่างของ entry รวมไปถึงความล้มเหลวของรีซอร์ส การแอนด์ที่ฟ หรือสถานการณ์การขึ้นอยู่กั้งงานอื่น หรือข้อผิดพลาดในการส่ง

การขึ้นอยู่กั้งงานอื่น

ถ้างานเป็นงานที่ขึ้นอยู่กั้งงานอื่น ให้ใช้อ็อปชัน 10 จากจอแสดงผล Work with Jobs เพื่อที่จะแสดงการขึ้นอยู่กั้งงานอื่น กด F8 เพื่อแสดงรายชื่อของงาน predecessor ทั้งหมด งานที่ขึ้นอยู่กั้งงานอื่นไม่สามารถรันได้ยกเว้นว่างาน predecessor ทั้งหมดจะแสดง *YES ในคอลัมน์ Complete .

การติดตามความก้าวหน้าของงาน

ถ้างานทำงานได้ไม่ถูกต้อง คุณสามารถใช้คำสั่ง Set Step using Job Scheduler (SETSTPJS) ก่อนหน้า หรือหลังจากขั้นตอนในโปรแกรม CL ของคุณ เพื่อช่วยกำหนดว่าปัญหาคืออะไร ระบุคำสั่งพร้อมด้วยคำอธิบายในโปรแกรม CL ของคุณ ใช้คำสั่งนี้ได้มาครั้งตามความจำเป็น คำอธิบายที่เกี่ยวข้องกับคำสั่งปัจจุบัน แสดงอยู่ในฟิลด์ Command step บนหน้า Last Run ของคุณสมบัติงานที่ จัดตารางเวลา นอกจากนี้ คุณสามารถดูฟิลด์ Command step ได้บนหน้าต่าง Status ของงานแอนด์ที่ฟ ฟิลด์ Command step จะถูกอัปเดตโดยอัตโนมัติทุกครั้งที่งานพบคำสั่ง SETSTPJS ใช้คำสั่งนี้ เพื่อช่วยพิจารณาความก้าวหน้าของงาน

การเก็บตัวอย่างข้อมูลเหล่านี้จะช่วยคุณในการวิเคราะห์ปัญหา:

เงื่อนไขของข้อความแสดงความผิดพลาด

พิมพ์บันทึกการใช้งานสำหรับเซสชันแบบโต้ตอบ, งานที่มอนิเตอร์หรืองานที่ถูกจัดตารางเวลาไว้ขึ้นอยู่กั้งว่าข้อผิดพลาดเกิดขึ้นที่ใด

วันที่ของตารางเวลางานไม่ถูกต้อง

ใช้คำสั่ง DSPJOBS สำหรับงานพร้อมด้วย OUTPUT(*PRINT) พิมพ์รายงานปฏิทินถ้ามีการใช้ปฏิทินในงานนั้น พิมพ์รายงานปฏิทินวันหยุดถ้ามีการใช้ปฏิทินวันหยุดในงานนั้น กดปุ่ม Print เพื่อพิมพ์จอแสดงผลของแต่ละ fiscal calendar entry สำหรับปฏิทินปีบัญชีที่ใช้ภายในงาน

ไฟล์บันทึก Advanced Job Scheduler

พิมพ์ไฟล์บันทึก Advanced Job Scheduler สำหรับช่วงเวลาที่มีย่อส่งสัปดาห์

ไฟล์ QALJSMST และ QALJSHST

ไฟล์ QALJSMST และไฟล์ QALJSHST ในไลบรารี QUSRIJS อาจจำเป็นต้องทำเจอร์นัลไว้ก่อนที่จะพยายามจำลองปัญหาให้เกิดขึ้นอีก นอกจากนี้ ไลบรารี QUSRIJS อาจเป็นที่ต้องการของฝ่ายสนับสนุนของ IBM

การทำงานกับ job schedule entries

นอกเหนือจากหน้าต่าง System i Navigator Job Properties – Job Queue คุณยังสามารถเปลี่ยนแปลง job schedule entry ได้โดยตรง โดยใช้อินเตอร์เฟซแบบอักษร ข้างล่างนี้เป็นรายการของภารกิจอินเตอร์เฟซแบบอักษรทั่วไปที่คุณสามารถใช้เมื่อทำงานกับ job schedule entries

สำคัญ: ห้ามใช้ Work with Job Schedule Entries (WRKJOBSCDE) เพื่อเปลี่ยน หรือลบงานที่จัดตารางเวลา โดยใช้ Management Central Scheduler หรือ Advanced Job Scheduler ถ้างานถูกเปลี่ยน หรือลบออกโดยใช้ WRKJOBSCDE, Management Central จะไม่ได้รับการแจ้งถึงการเปลี่ยนแปลง ภารกิจไม่สามารถรันได้ตามที่คาด และข้อความแสดงความผิดพลาดอาจปรากฏขึ้นในบันทึกการใช้งานเซิร์ฟเวอร์ Management Central

หลักการที่เกี่ยวข้อง

“Job schedule entries” ในหน้า 63

ถ้าระบบของคุณไม่มี Management Central Scheduler หรือ Advanced Job Scheduler คุณยังคงสามารถจัดตารางเวลางานได้โดยใช้ job schedule entry ซึ่งเข้าใช้งานได้จากอินเตอร์เฟซแบบอักษร ด้วยการใช้วิธีนี้ คุณจึงสามารถจัดตารางเวลางานให้รันซ้ำหรือรันเพียงครั้งเดียวได้

การใส่เพิ่ม job schedule entry:

คำสั่ง Add Job Schedule Entry (ADDJOBSCDE) ช่วยคุณสามารถจัดตารางเวลางานแบ็ตซ์โดยใช้ entry เพิ่มที่ตารางเวลางาน คุณสามารถใช้คำสั่งนี้เพื่อจัดตารางเวลางานแบ็ตซ์ให้ส่งพร้อมกันในคราวเดียว หรือ เพื่อจัดตารางเวลางานแบ็ตซ์ให้ส่งในช่วงเวลาที่สม่ำเสมอ

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)
              JOB(CLNUPLIB/CLNUPJOB)
```

การเปลี่ยน job schedule entry:

คำสั่งนี้จะเปลี่ยน entry ในตารางเวลางาน แต่ไม่ส่งผลต่อ งานใดๆ ที่ได้ส่งเรียบร้อยแล้วโดยใช้ entry นี้ ถ้าต้องการเปลี่ยน job entry ให้ใช้อินเตอร์เฟซแบบอักษร

เพื่อเปลี่ยน job schedule entry คุณต้องมีสิทธิเหมือนกันกับสิทธิ ที่ต้องใช้ในการใส่เพิ่ม entry อย่างไรก็ตาม, สิทธิในการใช้งาน แต่ละอ็อบเจ็กต์ จะถูกตรวจสอบเฉพาะเมื่อคุณกำลังเปลี่ยนพารามิเตอร์นั้นสำหรับ entry เท่านั้น ยิ่งกว่านั้น ถ้าคุณไม่มีสิทธิ พิเศษ *JOBCTL คุณสามารถเปลี่ยน ได้เฉพาะ entries ที่โปรไฟล์ผู้ใช้ของคุณใส่เพิ่มที่อ็อบเจ็กต์ตารางเวลางาน

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 ของงานแบ็ตช์ให้รันโปรแกรม A เมื่อเวลา 11 a.m. ของวันที่ 12/15/03 และในวัน เดียวกันของทุกสัปดาห์

```
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) อนุญาตให้คุณพัก entry หนึ่ง หรือทั้งหมด หรือเซตตารางเวลางานได้ ถ้า entry ถูกพัก จะไม่มีการส่งงานใดๆ ตามเวลาที่จัดตารางไว้ ถ้าต้องการพัก job schedule entry ให้ใช้อินเตอร์เฟซแบบอักษร

เพื่อพัก entries คุณต้องมีสิทธิพิเศษ job control (*JOBCTL); ถ้าไม่มี คุณสามารถพักได้เฉพาะ entries ที่คุณใส่เพิ่มเท่านั้น ถ้าคุณพัก job schedule entry:

- Entry จะถูกพักจนกว่าถูกรีลีสโดยใช้คำสั่ง Release Job Schedule Entry (RLSJOBSCDE) หรือ Work with Job Schedule Entries (WRKJOBSCDE)
- งานจะไม่ถูกส่งเมื่อถูกรีลีส แม้ว่าวันที่และเวลาที่งานถูก จัดตารางเวลาให้ส่งจะผ่านมาแล้วก็ตาม ในขณะที่ entry ถูกพักอยู่ แต่งานจะถูกส่งในวันที่ในอนาคตซึ่งจัดตารางเวลาไว้ให้ส่ง

คำสั่ง: Hold Job Schedule Entry (HDLJOBSCDE)

ตัวอย่าง: ตัวอย่างต่อไปนี้จะพัก job schedule entry CLEANUP

```
HLDJOBSCDE JOB(CLEANUP)
```

การพิมพ์รายการของ job scheduled entry:

ถ้าต้องการพิมพ์รายการของ job schedule entry ให้ใช้อินเตอร์เฟซแบบ อักษร

คำสั่ง: Work with Job Schedule Entries (WRKJOBSCDE)

ตัวอย่าง: ตัวอย่างต่อไปนี้จะพิมพ์รายการของ job schedule entries

```
WRKJOBSCDE OUTPUT(*PRINT)
```

ตัวอย่าง: ตัวอย่างต่อไปนี้จะพิมพ์ข้อมูลรายละเอียดเกี่ยวกับแต่ละ job schedule entry

การรีลีส job schedule entry:

คำสั่ง Release Job Schedule Entry (RLSJOBSCDE) ช่วยให้คุณสามารถรีลีส entry หนึ่ง, entry ทั้งหมด หรือชุดของ entry ในตารางเวลางาน ถ้าคุณรีลีส job schedule entry งานจะไม่ถูกส่งในทันที แม้ว่าวันที่และเวลาซึ่งถูกจัดตารางเวลาให้ส่งได้ผ่านไปแล้ว ในขณะที่ entry ถูกพักไว้ ถ้าเวลาที่จัดตารางเวลาผ่านไปแล้วในขณะที่ entry ถูกพัก จะมีการส่งข้อความเตือนเพื่อระบุว่างานหนึ่งหรือหลายงาน ขาดไป จากนั้นงานจะถูกส่งในวันที่ในอนาคตซึ่ง จัดตารางเวลาให้ส่ง ถ้าต้องการรีลีส job schedule entries ให้ ใช้อินเตอร์เฟซแบบอักษร

เพื่อรีลีส entries คุณต้องมีสิทธิ์พิเศษ 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, หลาย entry หรือ generic entry ในตารางเวลางานได้ แต่ละ job schedule entry เท่ากับงานแบ็คชิ่งหนึ่งงาน และมีข้อมูล ที่จำเป็นสำหรับการรันงานอย่างอัตโนมัติหนึ่งครั้ง หรือรันในช่วงเวลาสม่ำเสมอที่จัดตารางเวลาไว้ จะมีการส่งข้อความให้คุณและ message queue ถูกระบุใน job schedule entry เมื่อลบ entry เสร็จเรียบร้อยแล้ว ถ้าต้องการลบ job schedule entry ให้ใช้อินเตอร์เฟซแบบอักษร

เพื่อลบ entries คุณต้องกำลังรันอยู่ภายใต้โปรไฟล์ผู้ใช้ซึ่งมีสิทธิ์พิเศษ 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) command ข้อความระบบ CPC1239 จะถูกส่งไปที่ message queue ที่ระบุใน entry ถ้า single-submission entry ถูกพักไว้เมื่อถึงเวลาที่จัดตารางเวลาของ entry และ entry ระบุ *NO สำหรับแอ็ตทริบิวต์การบันทึก entry จะถูกลบออก เมื่อ entry ถูกรีลีสด้วยคำสั่ง Release Job Schedule Entry ในกรณีนี้ ข้อความ CPC1245 จะถูกส่งไปที่ message queue ที่ระบุใน entry

การจัดการระบบย่อย

เนื่องจากงานรันในระบบย่อย คุณอาจต้องมอนิเตอร์ activity ระบบย่อยสำหรับปัญหาที่อาจเกิดขึ้นได้ซึ่งอาจส่งผลกระทบต่อความสามารถในการรันของงาน

ระบบย่อย คือ สถานที่ทำงานสำหรับบนระบบ ชุดงานทั้งหมดของผู้ใช้ได้ทำผ่านทางชั้นงานต่างๆ ที่รันอยู่ในระบบย่อย จึงเป็นสิ่งสำคัญที่จะต้อง มอนิเตอร์พื้นที่เหล่านี้หากมีการทำงานที่ซ้ำเกิดขึ้น ใน System i Navigator คุณสามารถดูงาน และคิวงานที่เชื่อมโยงกับระบบย่อย นอกจากนี้ คุณยังมีฟังก์ชันเกี่ยวกับ งานและคิวงานที่เหมือนกันกับพื้นที่อื่นที่แสดงงานและคิวงานด้วย

ภารกิจระบบย่อยทั่วไป

ข้อมูลนี้อธิบายภารกิจทั่วไปที่คุณสามารถ ปฏิบัติบนระบบย่อย.

หลักการที่เกี่ยวข้อง

“ระบบย่อย” ในหน้า 11

ระบบย่อย คือ ที่ทำงานถูกประมวลผลบนเซิร์ฟเวอร์ ระบบย่อยเป็นสภาวะแวดล้อมการดำเนินการหนึ่งที่ได้กำหนดไว้ก่อน เพื่อให้ระบบประสานเวิร์กโฟลว์และการใช้รีซอร์สให้เหมาะสมกัน ระบบอาจประกอบด้วยระบบย่อยหลายระบบ โดยที่ระบบทั้งหมดดำเนินการ เป็นอิสระจากกัน ระบบย่อยจะจัดการรีซอร์สเอง

ข้อมูลที่เกี่ยวข้อง

รายงานจากประสบการณ์: การคอนฟิกูเรชันระบบย่อย

การดูแลทริบิวต์ระบบย่อย:

ระบบย่อยมีแอตทริบิวต์ต่างๆ แอตทริบิวต์เหล่านี้ให้ข้อมูลเกี่ยวกับ สถานะปัจจุบันของระบบย่อย หรือเกี่ยวกับค่าที่ระบุใน subsystem description

เมื่อคุณใช้ System i Navigator สามารถดูแอตทริบิวต์ต่อไปนี้ของระบบย่อยที่แอคทีฟ:

- **ระบบย่อย:** ชื่อของระบบย่อย และไลบรารีที่มี subsystem description
- **Description:** description ของระบบย่อย
- **สถานะ:** สถานะปัจจุบันของระบบย่อย คำอธิบายจะมีรายละเอียดของสถานะต่างๆ ที่ใช้ได้
- **งานแอคทีฟ:** จำนวนของงานที่แอคทีฟในปัจจุบัน กำลังรันหรือกำลังรอที่จะรัน อย่างไรก็ตามหนึ่งในระบบย่อย จำนวนงานดังกล่าวไม่รวมงานระบบย่อย
- **งานแอคทีฟสูงสุด:** จำนวนสูงสุดของงานที่สามารถแอคทีฟ กำลังรันหรือกำลังรอที่จะรัน อย่างไรก็ตามหนึ่งในระบบย่อย
- **งานระบบย่อย:** ชื่อของงานระบบย่อย รวมถึงผู้ใช้และ หมายเลข

System i Navigator:

ถ้าต้องการดูแอตทริบิวต์ของระบบย่อยให้ทำตามขั้นตอนเหล่านี้:

1. จาก System i Navigator ให้ขยาย My Connections → connection → Work Management → Subsystems → Active Subsystems.
2. คลิกขารระบบย่อยที่คุณต้องการดูแล้วคลิก Properties.

อินเตอร์เฟซแบบอักษร:

ถ้าต้องการใช้อินเตอร์เฟซแบบอักษร ให้พิมพ์คำสั่งต่อไปนี้:

คำสั่ง: Display Subsystem Description (DSPSBSD)

ตัวอย่าง: คำสั่งนี้จะแสดง subsystem description menu สำหรับระบบย่อย QBATCH

DSPSBSD QBATCH

การหยุดระบบย่อย:

คุณสามารถใช้ System i Navigator หรืออินเทอร์เฟซแบบอักษรเพื่อหยุดหนึ่ง หรือหลายระบบย่อย และระบุสิ่งที่จะเกิดขึ้นกับงานแอคทีฟที่กำลังจะประมวลผล ไม่มีงานใหม่ หรือขั้นตอนการเรดใหม่ถูกเริ่มในระบบย่อยหลังจากที่ระบบย่อยถูกหยุด

เมื่อระบบย่อยถูกหยุด คุณสามารถระบุสิ่งที่จะเกิดขึ้นกับงานแอคทีฟที่กำลัง ถูกระบบประมวลผลอยู่ได้ เช่น คุณสามารถระบุให้งานทุกงานในระบบย่อยหยุดการทำงานทันที (Immediate) หรือระบุให้งานได้รับอนุญาตให้ทำต่อไปจนเสร็จสิ้นการประมวลผลก่อนที่ระบบย่อยจะจบลง (Controlled)

สำคัญ: ขอแนะนำให้จบการทำงานของระบบย่อยด้วยอ็อปชัน Controlled ในทุกเมื่อที่เป็นไปได้ การจบลักษณะนี้จะช่วยให้งานแอคทีฟจบการทำงานด้วยตัวของมันเอง ใช้อ็อปชันนี้เพื่อให้มั่นใจว่างานเสร็จสิ้นการทำงานก่อนที่ระบบย่อยจะจบการทำงาน สิ่งนี้จะช่วยให้โปรแกรมที่กำลังรันอยู่ทำการ cleanup (กระบวนการสิ้นสุดงาน) ให้เรียบร้อย การระบุค่า Immediate อาจก่อให้เกิดผลที่ไม่พึงปรารถนา เช่น ข้อมูลถูกอัปเดตเพียงบางส่วน

การหยุดมีอยู่สองชนิด

Controlled (แนะนำ)

จบระบบย่อยในลักษณะที่ควบคุม งานจะถูกจบในลักษณะที่ควบคุม ด้วย สิ่งนี้ช่วยให้โปรแกรมที่กำลังรันอยู่ทำการ cleanup (กระบวนการสิ้นสุดงาน) ให้เรียบร้อย เมื่องานที่กำลังจะจบมีโปรซีเดอร์ทิศทางการจัดการสัญญาณสำหรับสัญญาณอะซิงโครนัส SIGTERM, จะมีการสร้างสัญญาณ SIGTERM สำหรับงานนั้น แอ็พพลิเคชันมีระยะเวลาตามที่ระบุสำหรับพารามิเตอร์ DELAY ในการทำ cleanup ให้เสร็จสมบูรณ์ก่อนที่งานจะถูกจบ

Immediate

จบระบบย่อยในทันที งานจะจบในทันทีด้วย เมื่องานที่กำลังจะจบ มีโปรซีเดอร์ทิศทางการจัดการสัญญาณสำหรับสัญญาณอะซิงโครนัส SIGTERM จะมีการสร้างสัญญาณ SIGTERM สำหรับงานนั้น และค่ากำหนดของระบบ QENDJOBLMT ระบุขีดจำกัดเวลานอกเหนือจากการจัดการสัญญาณ SIGTERM โปรแกรมที่กำลังรันอยู่ไม่ได้รับอนุญาตให้ปฏิบัติการ cleanup ใดๆ

หลักการที่เกี่ยวข้อง

“การจบงาน: แบบควบคุม” ในหน้า 120

การจบงานในลักษณะที่ควบคุมช่วยให้โปรแกรมที่กำลังรันอยู่ในงานสามารถปฏิบัติการล้าง end-of-job ได้ สามารถระบุเวลาหน่วง เพื่ออนุญาตให้งานจบลงในลักษณะที่ควบคุม ถ้าเวลาหน่วงหมดลงก่อนที่งานจะจบงานจะถูกจบในทันที

งานที่เกี่ยวข้อง

“วิธีการแสดงบันทึกการใช้งาน” ในหน้า 220

คุณสามารถดูบันทึกการใช้งานจากที่ใดก็ได้ภายในการจัดการระบบงานที่คุณเข้าไปใช้งาน เช่น ผ่านทางพื้นที่ระบบย่อย หรือพื้นที่พูลหน่วยความจำ คุณสามารถใช้ System i Navigator หรืออินเทอร์เฟซแบบอักษรเพื่อแสดงบันทึกการใช้งานได้

ข้อมูลที่เกี่ยวข้อง

ค่ากำหนดของระบบงาน: เวลาสูงสุดสำหรับการจบในทันที

System i Navigator:

ในการใช้ System i Navigator ใช้ข้อแนะนำดังต่อไปนี้:

1. จาก System i Navigator ให้ขยาย My Connections → connection → Work Management → Subsystems → Active Subsystems.
2. คลิกขวาที่ระบบย่อยหนึ่งหรือหลายระบบที่คุณต้องการหยุด แล้วคลิก Stop.
3. ระบุอ็อปชันที่ต้องการใช้เมื่อระบบย่อยถูกหยุด
4. คลิก Stop.

อินเทอร์เน็ตเฟสแบบอักษร:

ถ้าต้องการใช้อินเทอร์เน็ตเฟสแบบอักษร ให้พิมพ์คำสั่งต่อไปนี้:

คำสั่ง: End Subsystem (ENDSBS)

ตัวอย่าง: คำสั่งนี้จะจบงานแอนด์ที่พีทั้งหมดในระบบย่อย QBATCH และจบระบบย่อย งานแอนด์ที่พีเป็นเวลา 60 วินาทีในการปฏิบัติขั้นตอน การจบงานที่แอนด์พีเคชันจัดให้

```
ENDSBS SBS(QBATCH) OPTION(*CNTRL) DELAY(60)
```

ใช้พารามิเตอร์ End Subsystem Option (ENDSBSOPT) เพื่อพัฒนาประสิทธิภาพสำหรับการจบระบบย่อย ถ้าคุณระบุ ENDSBSOPT(*NOJOBLOG) ระบบย่อยจะจบ แต่จะไม่มีการสร้างบันทึกการใช้งานสำหรับงานทุกงานที่อยู่ในระบบย่อย

ถ้ามีปัญหาเกิดขึ้นในงาน แต่คุณได้ระบุ *NOJOBLOG การวินิจฉัยปัญหาอาจเป็นเรื่องยากหรือไม่สามารถทำได้ เนื่องจากปัญหาไม่ได้ ถูกบันทึกในบันทึกการใช้งาน ถ้าคุณใช้แอนด์พีเวิร์กชัน LOGOUTPUT(*PND) บันทึกการใช้งานจะถูกวางในสถานะคงค้าง แต่ไม่ถูกบันทึก อยากรู้ดี บันทึกการใช้งานยังคงพร้อมใช้งานถ้ามีความต้องการ ดูหัวข้อที่เกี่ยวข้องกับบันทึกการใช้งาน สำหรับข้อมูลเพิ่มเติมเกี่ยวกับการคงค้างบันทึกการใช้งาน

ถ้าคุณระบุ ENDSBSOPT(*CHGPTY *CHGTSL) ระดับความสำคัญการรัน และการแบ่งเวลาจะเปลี่ยนสำหรับงานทั้งหมดที่จบในระบบย่อยนี้ งานจะสำเร็จช้าลงสำหรับ processor cycle และจะจบโดยมีผลกระทบต่องานที่กำลังรันอยู่ในระบบย่อยอื่น น้อยลง

คุณสามารถระบุอ็อปชันทั้งหมดสามอ็อปชัน (*NOJOBLOG, *CHGPTY และ *CHGTSL) บนพารามิเตอร์ ENDSBSOPT parameter ตัวอย่างเช่น:

```
ENDSBSOPT(*NOJOBLOG *CHGPTY *CHGTSL)
```

หมายเหตุ: ถ้า คุณระบุ *ALL สำหรับชื่อระบบย่อยและมีงานใดๆ ที่กำลังรันอยู่ภายใต้ QSYSWRK คุณควรจะใช้ *CNTRLD เพื่อป้องกันไม่ทำให้ระบบย่อยจบอย่างผิดปกติ

การเริ่มระบบย่อย:

คำสั่ง Start Subsystem (STRSBS) จะเริ่มระบบย่อย โดยใช้ subsystem description ที่ระบุในคำสั่ง เมื่อระบบย่อย เริ่มต้น ระบบจะจัดสรรรีซอร์สที่จำเป็นและพร้อมใช้งาน (หน่วยเก็บ, เวิร์กสเตชัน และคิวงาน) ที่ถูกระบุใน subsystem description คุณสามารถเริ่มระบบย่อย โดยใช้ System i Navigator อินเทอร์เน็ตเฟส หรืออินเทอร์เน็ตเฟสแบบอักษร

หลักการที่เกี่ยวข้อง

“วิธีการเริ่มระบบย่อย” ในหน้า 21

เมื่อระบบย่อยเริ่มต้น ระบบจะจัดสรรสิ่งต่างๆ ให้ระบบย่อยและเริ่ม งานแบบ autostart และงานแบบ prestart ก่อนที่ระบบย่อยจะพร้อมสำหรับงาน

System i Navigator:

ในการเริ่มระบบย่อย โดยใช้ System i Navigator ให้ใช้ข้อแนะนำต่อไปนี้:

1. ขยาย My Connections → connection → Work Management.
2. คลิกขวา Subsystems และคลิก Start Subsystem.
3. ระบุ Name และ Library ของ ระบบย่อยที่จะเริ่มและคลิก OK.

อินเทอร์เฟซแบบอักษร:

คำสั่ง: Start Subsystem (STRSBS)

ตัวอย่าง: คำสั่งนี้จะเริ่มระบบย่อยของผู้ใช้ที่เชื่อมโยงกับ TELLER subsystem description ในไลบรารี QGPL ชื่อระบบย่อย คือ TELLER

STRSBS SBSD(QGPL/TELLER)

การสร้างรายละเอียดของระบบย่อย

คุณสามารถสร้าง subsystem description ได้สองวิธี คุณสามารถก๊อปปี้ subsystem description ที่มีอยู่และเปลี่ยนแปลง หรือคุณสามารถสร้าง description ใหม่ทั้งหมด

สองแนวทางที่คุณสามารถใช้มีดังต่อไปนี้:

1. ถ้าต้องการก๊อปปี้ subsystem description ที่มีอยู่โดยใช้อินเทอร์เฟซแบบ อักษร 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))
 - b. เปลี่ยนก๊อปปี้ของ subsystem description เพื่อให้ก๊อปปี้ทำงาน ในลักษณะที่คุณต้องการ ตัวอย่างเช่น คุณจะต้องลบ job queue entry ออกเนื่องจาก entry ระบุคิวงานที่ระบบย่อยดั้งเดิมใช้อยู่จากนั้น คุณต้องสร้าง job queue entry ใหม่ที่ระบุพารามิเตอร์ที่ระบบย่อยใหม่ใช้อยู่
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 ให้ใช้อินเทอร์เฟซ แบบอักษรและปฏิบัติตามคำแนะนำต่อไปนี้:
 - 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) to the subsystem description

หลักการที่เกี่ยวข้อง

“ระบบย่อย” ในหน้า 11

ระบบย่อย คือ ที่ทำงานถูกประมวลผลบนเซิร์ฟเวอร์ ระบบย่อยเป็นสถานะแวดล้อมการดำเนินการหนึ่งที่ได้กำหนด ไว้ก่อน เพื่อให้ระบบประสานเวิร์กโฟลว์และการใช้รีซอร์สให้เหมาะสมกัน ระบบอาจประกอบด้วยระบบย่อยหลายระบบ โดยที่ระบบทั้งหมดดำเนินการ เป็นอิสระจากกัน ระบบย่อยจะจัดการรีซอร์สเอง

“Subsystem description” ในหน้า 13

subsystem description คืออ็อบเจกต์ระบบที่มีข้อมูล ซึ่งกำหนดลักษณะของสถานะแวดล้อมการปฏิบัติการที่ควบคุมโดย

ระบบ identifier ที่ระบบรับรู้สำหรับชนิดอ็อบเจ็กต์คือ *SBSD subsystem description กำหนดว่างานจะเข้าสู่ระบบย่อยอย่างไร, ที่ไหน และงานมีจำนวนเท่าใด, และระบบย่อยจะใช้รีซอร์สใดในการปฏิบัติงาน ระบบย่อยแ็คทีฟจะใช้ชื่อธรรมดาของ subsystem description

ข้อมูลที่เกี่ยวข้อง

รายงานจากประสบการณ์: การคอนฟิกูเรชันระบบย่อย

การใส่เพิ่ม autostart job entries:

คุณใช้อินเตอร์เฟซแบบอักษรเพื่อใส่เพิ่ม autostart job entry งานแบบ autostart จะเริ่มโดยอัตโนมัติเมื่อระบบย่อยที่เกี่ยวข้องเริ่มต้นขึ้น โดยทั่วไปงานเหล่านี้จะทำการกำหนดค่าเริ่มต้นที่เชื่อมโยงกับระบบย่อย งานแบบ autostart ยังสามารถทำงานที่ซ้ำๆ หรือนำเสนอฟังก์ชัน การให้บริการศูนย์กลางสำหรับงานอื่นที่อยู่ในระบบย่อยเดียวกัน

คำสั่ง: Add Autostart Job Entry (ADDAJE)

ตัวอย่าง: ตัวอย่างนี้จะใส่เพิ่ม autostart job entry ที่ description ของ ระบบย่อย ABC

```
ADDAJE SBSB(USERLIB/ABC) JOB(START)
      JOBD(USERLIB/STARTJD)
```

หมายเหตุ: เพื่อให้การเปลี่ยนแปลงมีผลบังคับใช้ ต้องจบระบบย่อยที่แ็คทีฟ แล้วรีสตาร์ท

หลักการที่เกี่ยวข้อง

“การป้อนข้อมูลงานแบบ Autostart” ในหน้า 14

การป้อนข้อมูลงานแบบ autostart ระบุให้งานแบบ autostart เริ่มขึ้นในทันทีที่ระบบย่อยเริ่มต้นขึ้น. เมื่อระบบย่อยเริ่มทำงาน, ระบบจะจัดสรรสิ่งต่างๆ ให้ระบบย่อย และเริ่มงานแบบ autostart และ prestart ก่อนที่ระบบย่อยจะพร้อมที่จะทำงาน

การใส่เพิ่ม communications entries:

แต่ละ communication entry อธิบายอุปกรณ์สื่อสารหนึ่ง หรือหลายอุปกรณ์, ชนิดของอุปกรณ์ หรือตำแหน่งรีโมตซึ่งระบบย่อยจะเริ่มงาน เมื่อได้รับคำขอการเริ่มโปรแกรม 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 SBSB(ALIB/SBS1) DEV(COMDEV)
```

หมายเหตุ: คุณต้องระบุพารามิเตอร์เป็น DEV หรือ RMTLOCNAME แต่ไม่ใช่ทั้งสองตัว

หลักการที่เกี่ยวข้อง

“การป้อนข้อมูลแบบ Communication” ในหน้า 15

การป้อนข้อมูลงานแบบ communications ระบุให้ระบบย่อยทราบถึงซอร์สสำหรับงานสื่อสารที่ระบบจะประมวลผล การประมวลผลงานเริ่มขึ้นเมื่อระบบย่อยได้รับ คำขอการเริ่มโปรแกรมการสื่อสารจากระบบรีโมตและพบ routing entry ที่เหมาะสมสำหรับคำขอ

การใส่เพิ่ม job queue entries:

job queue entry ระบุคิวงานซึ่งงานจะถูกเลือกสำหรับการรันในระบบย่อย งานที่เริ่มต้นจากคิวงานคืองานแบ็ตซ์ คุณสามารถใส่เพิ่ม job queue entry โดยใช้อินเตอร์เฟซแบบอักษร

คุณสามารถระบุไอเท็มดังต่อไปนี้ใน job queue entry

- ชื่อคิวงาน (JOBQ)
- จำนวนสูงสุดของงานที่สามารถแอคทีฟในเวลาเดียวกันจากคิวงาน (MAXACT)
- ลำดับของระบบย่อยในการเลือกคิวงานซึ่งมีงานที่สามารถถูกเริ่มต้นได้ (SEQNBR)
- จำนวนสูงสุดของงานที่สามารถแอคทีฟในเวลาเดียวกันสำหรับระดับความสำคัญของคิวงานที่ระบุ (MAXPTYn)

คำสั่ง: Add Job Queue Entry (ADDJOBQE)

ตัวอย่าง: คำสั่งนี้จะใส่เพิ่ม job queue entry สำหรับคิวงาน NIGHT (ในไลบรารี QGPL) ที่ NIGHTSBS subsystem description ซึ่งตั้งอยู่ใน ไลบรารี QGPL entry ระบุว่าสามารถมีงานแบ็ตซ์จากคิวงาน NIGHT ที่แอคทีฟในเวลาเดียวกันได้มากถึงสามงานในระบบย่อย สมมติว่า หมายเลขลำดับดีฟอลต์คือ 10

```
ADDJOBQE SBS(D:QGPL/NIGHTSBS) JOBQ(QGPL/NIGHT) MAXACT(3)
```

หลักการที่เกี่ยวข้อง

“Job queue entry” ในหน้า 71

job queue entry ระบุคิวงานซึ่งงานจะถูกเลือกสำหรับการรันในระบบย่อย พารามิเตอร์ใน job queue entry ที่ควบคุมวิธีการจัดการคิวงานมีอยู่ห้าพารามิเตอร์

“Job queue entry” ในหน้า 15

Job queue entry ใน subsystem description ระบุ คิวงานซึ่งระบบย่อยจะได้รับงานมา เมื่อระบบย่อยถูกเริ่มขึ้น ระบบย่อยจะพยายามจัดสรรคิวงานแต่ละคิวที่กำหนดใน subsystem job queue entry

การใส่เพิ่ม prestart job entries:

Prestart job entries ระบุงานแบบ prestart ที่สามารถเริ่มต้น เมื่อระบบย่อยเริ่มต้น หรือเมื่อป้อนคำสั่ง 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 SBS(D:USERLIB/ABC) PGM(START)
      JOBD(USERLIB/STARTPJ)
```

หลักการที่เกี่ยวข้อง

“Prestart job entry” ในหน้า 54

คุณกำหนดงานแบบ prestart โดยใช้ prestart job entry prestart job entry ไม่มีผลต่อการจัดสรรอุปกรณ์หรือการกำหนดค่าของการเริ่มโปรแกรม

“การสืบสวนงานแบบ prestart” ในหน้า 244

หัวข้อนี้แนะนำขั้นตอนที่ช่วยให้คุณในการตอบคำถาม “ฉันจะ ค้นหาผู้ใช้จริงของงานแบบ prestart และกำหนดรีซอร์สที่ใช้โดยงานแบบ prestart นั้นได้อย่างไร?”

ข้อมูลที่เกี่ยวข้อง

รายงานจากประสบการณ์: การปรับ prestart job entry

การใส่เพิ่ม routing entries:

แต่ละ routing entry ระบุพารามิเตอร์ที่ใช้ในการเริ่มขั้นตอนการเรอต์ สำหรับงาน Routing entries ระบุพูลระบบย่อยของแหล่งเก็บข้อมูลหลักที่จะ ใช้โปรแกรมการควบคุมที่จะรัน (โดยปกติคือ โปรแกรมที่ระบบจัดให้ QCMD), และข้อมูลรันไทม์เพิ่มเติม (จัดเก็บอยู่ในคลาสอ็อบเจกต์) ถ้าต้องการใส่เพิ่ม routing entry ที่ subsystem description, ให้ใช้อินเตอร์เฟซแบบอักขระ

คำสั่ง: Add Routing Entry (ADDRTGE)

ตัวอย่าง: คำสั่งนี้จะใส่เพิ่ม routing entry 46 ที่ subsystem description PERT ในไลบรารี ORDLIB เพื่อใช้ routing entry 46 ข้อมูลการเรอต์ ต้องเริ่มด้วยสตริงอักขระ WRKSTN2 ที่เริ่มต้นในตำแหน่ง 1 ขั้นตอนการเรอต์สามารถแอ็คทีฟได้ไม่จำกัดจำนวนผ่านทาง entry นี้ได้ตลอดเวลา โปรแกรม GRAPHIT ในไลบรารี ORDLIB จะรันในพูลของหน่วยความจำ 2 โดยใช้คลาส AZERO ในไลบรารี MYLIB

```
ADDRTGE  SBS(ORDLIB/PERT)  SEQNBR(46)  CMPVAL(WRKSTN2)
          PGM(ORDLIB/GRAPHIT)  CLS(MYLIB/AZERO)  MAXACT(*NOMAX)
          POOLID(2)
```

หลักการที่เกี่ยวข้อง

“routing entry” ในหน้า 17

routing entry ระบุพูลระบบย่อยของแหล่งเก็บข้อมูลหลักที่จะ ใช้, โปรแกรมการควบคุมที่จะรัน (โดยปกติคือ โปรแกรมที่ระบบจัดให้ QCMD), และข้อมูลรันไทม์เพิ่มเติม (จัดเก็บอยู่ในคลาสอ็อบเจกต์) Routing entries ถูกจัดเก็บใน subsystem description

การใส่เพิ่ม workstation entries:

workstation entry จะถูกใช้เมื่องานเริ่มต้น เมื่อผู้ใช้ sign on หรือถ่ายโอนงานแบบโต้ตอบจากระบบย่อยอื่น คุณสามารถระบุไอเท็มดังต่อไปนี้ใน workstation entry ชื่อพารามิเตอร์แสดงอยู่ในวงเล็บ ใช้อินเตอร์เฟซแบบอักขระเพื่อใส่เพิ่ม workstation entries

- ชื่อ หรือชนิดของเวิร์กสเตชัน (WRKSTN หรือ WRKSTNTYPE)
- ชื่อรายละเอียดของงาน (JOBID) หรือชื่อรายละเอียดของงานในโปรไฟล์ผู้ใช้
- จำนวนสูงสุดของงานที่สามารถแอ็คทีฟในเวลาเดียวกันผ่านทาง (MAXACT)
- เวลาที่เวิร์กสเตชันจะถูกจัดสรร นั่นคือ เมื่อระบบย่อยถูกเริ่ม หรือเมื่องานแบบโต้ตอบเข้าสู่ระบบย่อยผ่านทางคำสั่ง Transfer Job (TFRJOB) และพารามิเตอร์ AT

ถ้าต้องการใส่เพิ่ม workstation entry ที่ subsystem description ให้ใช้อินเตอร์เฟซแบบอักขระ

คำสั่ง: Add Workstation Entry (ADDWSE)

ตัวอย่าง: ตัวอย่างต่อไปนี้จะใส่เพิ่ม workstation entry DSP12 ที่ระบบย่อย ABC.

```
ADDWSE SBS(D(USRLIB/ABC) WRKSTN(DSP12)
      JOBD(USRLIB/WSE)
```

หลักการที่เกี่ยวข้อง

“Workstation entries” ในหน้า 17

งานแบบโต้ตอบคืองานที่เริ่มขึ้นเมื่อผู้ใช้ signs on ที่จอภาพ และสิ้นสุดลงเมื่อผู้ใช้ signs off สำหรับงานที่จะรัน ระบบย่อย จะค้นหารายละเอียดของงาน ซึ่งอาจจะอยู่ใน workstation entry หรือโปรไฟล์ผู้ใช้

การสร้างไฟล์แสดงผล sign-on:

ไฟล์แสดงผล sign-on ใช้ในการแสดงจอแสดงผล sign-on ที่เวิร์กสเตชันซึ่งถูกจัดสรรให้กับระบบย่อย สามารถเปลี่ยนไฟล์แสดงผล sign-on เมื่อระบบย่อยแอดที่พ้อยๆ อย่างไม่ใช้ไฟล์แสดงผล sign-on ใหม่จนกว่าระบบย่อยจะเริ่มในครั้งถัดไปในการสร้างไฟล์แสดงผล sign-on ให้ใช้อินเตอร์เฟซแบบอักษร

สามารถสร้างไฟล์แสดงผล sign-on ใหม่โดยใช้ไฟล์แสดงผล sign-on ที่จัดโดย IBM เป็นจุดเริ่มต้น ซอร์สสำหรับไฟล์แสดงผลนี้ตั้งอยู่ใน โลบรารี QGPL ใน source physical file QDDSSRC ขอแนะนำอย่าให้ท่านสร้าง source physical file ใหม่และก๊อปปี้ไฟล์แสดงผลที่ได้มาจาก IBM ไปที่ source physical file ใหม่ก่อนที่จะทำการเปลี่ยนแปลงใดๆ ด้วยวิธีนี้ ซอร์สที่ได้มาจาก IBM ยังคงพร้อมใช้งานได้

ข้อควรพิจารณา:

- ลำดับในการประกาศฟิลด์ในไฟล์แสดงผล sign-on ต้องไม่มีการเปลี่ยนแปลง แต่ตำแหน่งในการแสดงผลฟิลด์บนจอแสดงผล สามารถเปลี่ยนได้
- ห้ามเปลี่ยนขนาดทั้งหมดของอินพุตบัฟเฟอร์หรือเอาต์พุตบัฟเฟอร์ อาจเกิดปัญหาที่ร้ายแรงได้ ถ้ามีการเปลี่ยนลำดับหรือขนาดของบัฟเฟอร์
- ห้ามใช้ฟังก์ชันคำอธิบาย data descriptions specifications (DDS) ในไฟล์แสดงผล sign-on
- ระบุ 256 เสมอบนพารามิเตอร์ MAXDEV
- ไม่สามารถระบุคีย์เวิร์ด MENUBAR และ PULLDOWN ใน sign-on display file description
- ความยาวบัฟเฟอร์สำหรับไฟล์แสดงผลต้องเป็น 318 ถ้าความยาวน้อยกว่า 318 ระบบย่อยจะใช้จอแสดงผล sign-on ดีฟอลต์, QDSIGNON ในโลบรารี QSYS
- ไม่สามารถลบบรรทัดลิสทิง
- เมมเบอร์ QDSIGNON คือ ไฟล์แสดงผล sign-on ที่จัดให้โดย IBM ซึ่งใช้รหัสผ่านที่เป็นอักขระ 10 ตัว
- เมมเบอร์ QDSIGNON2 คือ ไฟล์แสดงผล sign-on ที่จัดให้โดย IBM ซึ่งใช้รหัสผ่านที่เป็นอักขระ 128 ตัว

คำสั่ง: Create Display File (CRTDSPF)

ฟิลด์แบบ hidden ในไฟล์แสดงผลที่มีชื่อว่า UBUFFER สามารถเปลี่ยนเป็น ฟิลด์จัดการที่เล็กลงได้ UBUFFER มีความยาว 128 ไบต์และถูกกำหนดเป็นฟิลด์สุดท้ายในไฟล์แสดงผล ฟิลด์นี้สามารถเปลี่ยนฟังก์ชันเป็นอินพุต/เอาต์พุตบัฟเฟอร์เพื่อให้ข้อมูลที่ระบุในฟิลด์นี้จะได้พร้อมใช้งานที่แอสพลีเคชันโปรแกรมเมื่อเริ่มงานแบบโต้ตอบ คุณสามารถเปลี่ยนฟิลด์ UBUFFER ให้มี ฟิลด์ขนาดเล็กได้มากตามที่คุณต้องการ ถ้าอยู่ภายใต้ข้อกำหนดดังต่อไปนี้:

- ฟิลด์ใหม่ต้องตามหลังฟิลด์อื่นทั้งหมดในไฟล์แสดงผล ตำแหน่งของฟิลด์ บนจอแสดงผลไม่ใช่สิ่งสำคัญ トラบเทาที่ลำดับที่ฟิลด์ถูกวาง ใน data description specifications (DDS) ยังคงเป็นไปตามข้อกำหนดนี้
- ความยาวทั้งหมดต้องเป็น 128 ถ้าความยาวของฟิลด์ต่างๆมากกว่า 128 บางข้อมูลจะไม่ถูกส่งผ่าน
- ฟิลด์ทั้งหมดต้องเป็นฟิลด์อินพุต/เอาต์พุต (ชนิด B ในซอร์ส DDS) หรือฟิลด์แบบ hidden (ชนิด H ในซอร์ส DDS)

ข้อมูลที่เกี่ยวข้อง

โลแคลเป็นส่วนของสภาวะแวดล้อมหลายภาษา

DDS สำหรับ display files

การระบุการแสดงผล sign-on ใหม่:

ระบบย่อยใช้ไฟล์แสดงผล sign-on ที่ถูกระบุในพารามิเตอร์ SGNDSPF ของ subsystem description ในการสร้างจอแสดงผล sign-on ที่เวิร์กสเตชันผู้ใช้ ถ้าต้องการเปลี่ยนไฟล์แสดงผล sign-on จากดีฟอลต์ (QDSIGNON) เป็นค่าที่คุณสร้างขึ้น ให้ใช้อินเตอร์เฟซแบบอักษร

หมายเหตุ: ใช้เวอร์ชันทดสอบของระบบย่อยในการตรวจสอบว่า จอแสดงผล ถูกต้อง ก่อนที่จะพยายามเปลี่ยนระบบย่อยการควบคุม

คำสั่ง: Change Subsystem Description (CHGSBSD)

ระบุไฟล์แสดงผลใหม่บนพารามิเตอร์ SGNDSPF

ตัวอย่าง: สิ่งต่อไปนี้จะเปลี่ยนไฟล์แสดงผล sign-on สำหรับระบบย่อย QBATCH จากค่าดีฟอลต์เป็นไฟล์ใหม่ที่มีชื่อว่า MYSIGNON

```
CHGSBSD SBSD(QSYS/QBATCH) SGNDSPF(MYSIGNON)
```

ข้อมูลที่เกี่ยวข้อง

โลแคลเป็นส่วนของสภาวะแวดล้อมหลายภาษา

DDS สำหรับ display files

การเปลี่ยนรายละเอียดของระบบย่อย

คำสั่ง Change Subsystem Description (CHGSBSD) เปลี่ยนแอตทริบิวต์การดำเนินการของ subsystem description ที่ระบุ คุณสามารถเปลี่ยน subsystem description ในขณะที่ระบบย่อยแอ็คทีฟ ถ้าต้องการเปลี่ยน subsystem description ให้ใช้อินเตอร์เฟซแบบอักษร

หมายเหตุ: คุณไม่สามารถระบุค่า *RMV บนพารามิเตอร์ POOLS ในขณะที่ระบบย่อยแอ็คทีฟ เนื่องจากงานอาจถูกหยุดชั่วคราว

คำสั่ง: Change Subsystem Description (CHGSBSD)

ตัวอย่าง: คำสั่งนี้จะเปลี่ยน definition ของพูลหน่วยความจำ 2 ที่ระบบย่อย PAYCTL ใช้เป็นหน่วยความจำขนาด 1500K และ activity level เป็น 3 ไฟล์แสดงผล sign-on จะถูกเปลี่ยนเป็นไฟล์แสดงผล COMPANYA และตั้งอยู่ในไลบรารี QGPL library ถ้าระบบย่อยแอ็คทีฟเมื่อใช้คำสั่งนี้ จะไม่มีการใช้ COMPANYA จนกว่าระบบย่อยจะเริ่มในครั้งถัดไป

```
CHGSBSD SBSD(QGPL/PAYCTL) POOLS((2 1500 3))
SGNDSPF(QGPL/COMPANYA)
```

หลักการที่เกี่ยวข้อง

“Subsystem description” ในหน้า 13

Subsystem description คืออ็อบเจกต์ระบบที่มีข้อมูล ซึ่งกำหนดลักษณะของสถานะแวดล้อมการปฏิบัติการที่ควบคุมโดยระบบ identifier ที่ระบบรับรู้สำหรับชนิดอ็อบเจกต์คือ *SBSD subsystem description กำหนดว่างานจะเข้าสู่ระบบย่อยอย่างไร, ที่ไหน และงานมีจำนวนเท่าใด, และระบบย่อยจะใช้รีซอร์สใดในการปฏิบัติงาน ระบบย่อยแอ็คทีฟจะใช้ชื่อธรรมดาของ subsystem description

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 SBSB(USERLIB/ABC) JOB(START)
      JOBD(USERLIB/NEWJD)
```

หมายเหตุ: เพื่อให้การเปลี่ยนแปลงมีผลบังคับใช้ ต้องจบระบบย่อยที่แอ็คทีฟ แล้วรีสตาร์ท

หลักการที่เกี่ยวข้อง

“การป้อนข้อมูลงานแบบ Autostart” ในหน้า 14

การป้อนข้อมูลงานแบบ autostart ระบุให้งานแบบ autostart เริ่มขึ้นในทันทีที่ระบบย่อยเริ่มต้นขึ้น. เมื่อระบบย่อยเริ่มทำงาน, ระบบจะจัดสรรสิ่งต่างๆ ให้ระบบย่อย และเริ่มงานแบบ autostart และ prestart ก่อนที่ระบบย่อยจะพร้อมที่จะทำงาน

การเปลี่ยน communication entries:

คุณสามารถเปลี่ยนแอ็คทีวิตีของ communications entry ที่มีอยู่ใน subsystem description ที่มีอยู่ได้โดยใช้อินเตอร์เฟซแบบอักษร

- เมื่อเปลี่ยนพารามิเตอร์ Job description (JOBD) หรือ Default user profile (DFTUSR) communications entry จะเปลี่ยนไปด้วย อย่างไรก็ตาม ค่าของพารามิเตอร์เหล่านี้จะไม่เปลี่ยนสำหรับงานใดๆที่แอ็คทีฟในขณะนั้น
- ถ้าค่าของพารามิเตอร์ Maximum active jobs (MAXACT) ลดลงเป็นจำนวนน้อยกว่าจำนวนทั้งหมดของงานที่แอ็คทีฟผ่านทาง communications entry จะไม่มีการประมวลผลคำขอการเริ่มโปรแกรมใหม่ งานแอ็คทีฟจะรันต่อไป แต่ไม่มีการประมวลผลคำขอการเริ่มโปรแกรมเพิ่มเติม จนกว่าจำนวนของงานแอ็คทีฟจะน้อยกว่าค่าที่ระบุไว้สำหรับพารามิเตอร์ MAXACT

คำสั่ง: Change Communications Entry (CHGCMNE)

ตัวอย่าง: ตัวอย่างนี้จะเปลี่ยน communications entry (ใน subsystem description QGPL/BAKER) สำหรับอุปกรณ์ A12 และโหมด *ANY activity level สูงสุดจะถูกเปลี่ยนเป็น *NOMAX ซึ่งหมายความว่า communications entry ไม่มีข้อจำกัดเกี่ยวกับจำนวนของคำขอการเริ่มโปรแกรมที่สามารถแอ็คทีฟได้พร้อมกัน อย่างไรก็ตาม ค่า MAXJOBS ใน subsystem description BAKER จะจำกัดจำนวนทั้งหมดของงานที่สามารถแอ็คทีฟได้ในระบบย่อย จำนวนนี้รวมถึงงานที่สร้างโดยคำขอการเริ่มโปรแกรม และยังมีขีดจำกัดที่ผู้ใช้สามารถระบุจำนวนของงานแอ็คทีฟ ซึ่งสามารถเรอต์ผ่านทาง routing entry (MAXACT)

ขีดจำกัดที่ระบุใน routing entry สามารถควบคุมจำนวนของงานที่ใช้พูล หรือ ระดับการเรียกซ้ำของโปรแกรม ในทุกกรณี งานที่เป็นผลมาจากการประมวลผลคำขอการ เริ่มโปรแกรมไม่สามารถมีมากเกินไปเกินจำนวนขีดจำกัดเหล่านี้

```
CHGCMNE SBSDB(QGPL/BAKER) DEV(A12) MAXACT(*NOMAX)
```

หลักการที่เกี่ยวข้อง

“การป้อนข้อมูลแบบ Communication” ในหน้า 15

การป้อนข้อมูลงานแบบ communications ระบุให้ระบบย่อยทราบถึงซอร์สสำหรับงานสื่อสารที่ระบบจะประมวลผล การประมวลผลงานเริ่มขึ้นเมื่อระบบย่อยได้รับ คำขอการเริ่มโปรแกรมการสื่อสารจากระบบริโมตและพบ routing entry ที่เหมาะสมสำหรับคำขอ

การเปลี่ยน job queue entry:

คุณสามารถเปลี่ยน job queue entry ที่มีอยู่ใน subsystem description ที่ระบุได้ สามารถใช้คำสั่งนี้ในขณะที่ระบบย่อยแอคทีฟหรือไม่แอคทีฟ ถ้าต้องการเปลี่ยน job queue entry ในระบบย่อย ให้ใช้อินเตอร์เฟซแบบอักษร

คำสั่ง: Change Job Queue Entry (CHGJOBQE)

ตัวอย่าง: คำสั่งนี้จะเปลี่ยนจำนวนสูงสุดของงานที่สามารถ แอคทีฟได้พร้อมกันจากคิวงาน QBATCH ในไลบรารี QGPL หมายเลขลำดับของ job queue entry จะไม่เปลี่ยนแปลง สามารถมีงานจากคิวงาน QBATCH ที่แอคทีฟพร้อมกันได้มากถึงสี่งาน งานจากระดับความสำคัญ 1 สามารถ แอคทีฟได้หนึ่งงาน. ไม่มีข้อจำกัดจำนวนสูงสุดของงานที่สามารถแอคทีฟ พร้อมกันจากระดับความสำคัญ 2. ระดับความสำคัญ 3 ถึง 9 ไม่เปลี่ยนแปลง

```
CHGJOBQE SBSDB(QGPL/QBATCH) JOBQ(QGPL/QBATCH) MAXACT(4)
MAXPTY1(1) MAXPTY2(*NOMAX)
```

หลักการที่เกี่ยวข้อง

“Job queue entry” ในหน้า 71

job queue entry ระบุคิวงานซึ่งงานจะถูกเลือกสำหรับการรันในระบบย่อย พารามิเตอร์ใน job queue entry ที่ควบคุมวิธีการจัดการคิวงานมีอยู่ที่พารามิเตอร์

“Job queue entry” ในหน้า 15

Job queue entry ใน subsystem description ระบุ คิวงานซึ่งระบบย่อยจะได้รับงานมา เมื่อระบบย่อยถูกเริ่มขึ้น ระบบย่อยจะพยายามจัดสรรคิวงานแต่ละคิวที่กำหนดใน subsystem job queue entry

การเปลี่ยน prestart entry:

คุณสามารถเปลี่ยน prestart job entry ใน subsystem description ที่ระบุได้ ระบบย่อยอาจแอคทีฟอยู่เมื่อ prestart job entry ถูกเปลี่ยน การทำการเปลี่ยนแปลงที่ entry เมื่อระบบย่อยแอคทีฟจะมีผลต่อการดำเนิน การในอนาคต งานแบบ prestart ใหม่ใดๆ ที่เริ่มขึ้นหลังจากใช้คำสั่ง จะใช้ ค่าที่เกี่ยวข้องกับงานใหม่ คำสั่งนี้ระบุงานแบบ prestart ที่เริ่มขึ้นเมื่อระบบย่อยเริ่มต้น 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 ที่มีอยู่ในไลบรารี QGPL งานแบบ prestart ที่เกี่ยวข้องกับ entry นี้จะไม่เริ่มขึ้น ในครั้งถัดไปที่ PJSBS subsystem description ในไลบรารี QGPL ถูกเริ่มขึ้น ต้องใช้คำสั่ง STRPJ ในการเริ่มงานแบบ prestart เมื่อต้องเริ่มงานมากขึ้น จะต้องเริ่มงานเพิ่มเติมอีกหนึ่งงาน

```
CHGPJE SBS(D(QGPL/PJSBS) PGM(QGPL/PGM1) STRJOBS(*NO)
THRESHOLD(1) ADLJOBS(1)
```

หลักการที่เกี่ยวข้อง

“Prestart job entry” ในหน้า 54

คุณกำหนดงานแบบ prestart โดยใช้ prestart job entry prestart job entry ไม่มีผลต่อการจัดสรรอุปกรณ์หรือการกำหนดคำขอการเริ่มโปรแกรม

“การสืบสวนงานแบบ prestart” ในหน้า 244

หัวข้อนี้นำเสนอขั้นตอนที่ช่วยให้คุณในการตอบคำถาม “ฉันจะ ค้นหาผู้ใช้จริงของงานแบบ prestart และกำหนดรีซอร์สที่ใช้โดยงานแบบ prestart นั้นได้อย่างไร?”

ข้อมูลที่เกี่ยวข้อง

รายงานจากประสบการณ์: การปรับ prestart job entry

การเปลี่ยน routing entry:

คุณสามารถเปลี่ยน routing entry ใน subsystem description ที่ระบุได้โดยใช้อินเตอร์เฟซแบบอักษร routing entry ระบุพารามิเตอร์ที่จะใช้ในการเริ่มขั้นตอนการเราต์สำหรับงานหนึ่ง 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 SBS(D(LIB5/ORDER) SEQNBR(1478) CLS(LIB6/SOFAST) POOLID(3)
```

หลักการที่เกี่ยวข้อง

“routing entry” ในหน้า 17

routing entry ระบุพารามิเตอร์ของแหล่งเก็บข้อมูลหลักที่จะใช้, โปรแกรมการควบคุมที่จะรัน (โดยปกติคือ โปรแกรมที่ระบบจัดให้ QCMD), และข้อมูลรันใหม่เพิ่มเติม (จัดเก็บอยู่ในคลาสอ็อบเจกต์) Routing entries ถูกจัดเก็บใน subsystem description

การเปลี่ยน workstation entry:

คุณสามารถระบุรายละเอียดของงานอื่นสำหรับ workstation entry ที่ได้กำหนดไว้ก่อนหน้านี้ได้โดยใช้อินเตอร์เฟซแบบอักษร

- เมื่อมีการระบุพารามิเตอร์ Job description (JOBID) workstation entry จะถูกเปลี่ยนแปลง อย่างไรก็ตาม ค่าของพารามิเตอร์นี้จะไม่เปลี่ยนสำหรับงานใดๆที่เริ่มต้นผ่านทาง entry นี้ซึ่งแอ็คทีฟอยู่ในขณะนี้
- ถ้าค่าของพารามิเตอร์ Maximum active jobs (MAXACT) ลดลงเป็นจำนวนน้อยกว่าจำนวนทั้งหมดของเวิร์กสเตชันที่แอ็คทีฟผ่านทาง workstation entry จะไม่มีเวิร์กสเตชันเพิ่มเติมใดได้รับอนุญาตให้ sign on Workstation ที่แอ็คทีฟอยู่ถูก signed off สามารถสร้างงานเพิ่มเติมสำหรับเวิร์กสเตชันที่แอ็คทีฟด้วยคำสั่ง Transfer Secondary Job (TFRSECJOB) หรือคำสั่ง Transfer to Group Job (TFRGRPJOB) เวิร์กสเตชันอื่นจะไม่ได้รันอนุญาตให้ sign on ได้จนกว่าจำนวนของเวิร์กสเตชันจะน้อยกว่าค่าที่ระบุสำหรับพารามิเตอร์ MAXACT

คำสั่ง: Change Workstation Entry (CHGWSE)

ตัวอย่าง: คำสั่งนี้จะเปลี่ยน workstation entry สำหรับเวิร์กสเตชัน A12 ในระบบย่อย BAKER ที่พบในไลบรารีเนกประสงค์ งานจะถูกสร้างขึ้นสำหรับเวิร์กสเตชัน A12 เมื่อมีการป้อนรหัสผ่านของผู้ใช้บนหน้าจอ sign-on และกดปุ่ม Enter

```
CHGWSE  SBSDB(QGPL/BAKER)  WRKSTN(A12)  AT(*SIGNON)
```

หลักการที่เกี่ยวข้อง

“Workstation entries” ในหน้า 17

งานแบบโต้ตอบคืองานที่เริ่มขึ้นเมื่อผู้ใช้ signs on ที่จอภาพ และสิ้นสุดลงเมื่อผู้ใช้ signs off สำหรับงานที่จะรัน ระบบย่อย จะค้นหารายละเอียดของงาน ซึ่งอาจจะอยู่ใน workstation entry หรือโปรไฟล์ผู้ใช้

การเปลี่ยนการแสดงผล sign-on:

ระบบของคุณถูกจัดส่งมาพร้อมกับไฟล์แสดงผล sign-on ตามค่าดีฟอลต์ของ QDSIGNON ซึ่งตั้งอยู่ในไลบรารี QSYS ในสภาพที่คุณมีสภาพแวดล้อมหลายภาษา คุณอาจต้องการเปลี่ยนสิ่งที่แสดงบนจอภาพ sign-on หรือ บางทีคุณอาจต้องการใส่เพิ่มข้อมูลบริษัทของคุณลงบนจอภาพ sign-on ในสถานการณ์ดังกล่าว สิ่งแรกที่ต้องทำคือคุณต้องสร้างไฟล์แสดงผลใหม่ ถ้าต้องการทำเช่นนั้น ให้ใช้อินเตอร์เฟซแบบอักษร

แอ็ดทริบิวต์ SGNDSPF ใน subsystem description ซึ่ไปที่ไฟล์แสดงผล sign-on ที่ผู้ใช้จะเห็นเมื่อ sign on เข้าระบบย่อย

ขั้นตอนที่ใช้ในการเปลี่ยนจอแสดงผล sign-on สรุปได้ดังต่อไปนี้ :

1. สร้างไฟล์แสดงผล sign-on ใหม่
2. เปลี่ยน subsystem description เพื่อใช้ไฟล์แสดงผลที่เปลี่ยนแปลง แทนค่าดีฟอลต์ของระบบ
3. ทดสอบการเปลี่ยนแปลง

งานที่เกี่ยวข้อง

“การสร้างไฟล์แสดงผล sign-on” ในหน้า 185

ไฟล์แสดงผล sign-on ใช้ในการแสดงจอแสดงผล sign-on ที่เวิร์กสเตชันซึ่งถูกจัดสรรให้กับระบบย่อย สามารถเปลี่ยนไฟล์แสดงผล sign-on เมื่อระบบย่อยแอ็ดทริบิวต์อย่างไรก็ตาม จะไม่ใช่ไฟล์แสดงผล sign-on ใหม่จนกว่าระบบย่อยจะเริ่มในครั้งถัดไปในการสร้างไฟล์แสดงผล sign-on ให้ใช้อินเตอร์เฟซแบบอักษร

“การระบุการแสดงผล sign-on ใหม่” ในหน้า 186

ระบบย่อยใช้ไฟล์แสดงผล sign-on ที่ถูกระบุในพารามิเตอร์ SGNDSPF ของ subsystem description ในการสร้างจอแสดงผล sign-on ที่เวิร์กสเตชันผู้ใช้ ถ้าต้องการเปลี่ยนไฟล์แสดงผล sign-on จากดีฟอลต์ (QDSIGNON) เป็นค่าที่คุณสร้างขึ้น ให้ใช้อินเตอร์เฟซแบบอักษร

ข้อมูลที่เกี่ยวข้อง

โลแคลเป็นส่วนของสภาวะแวดล้อมหลายภาษา

DDS สำหรับ display files

การลบรายละเอียดของระบบย่อย

คำสั่ง Delete Subsystem Description (DLTSBSD) จะลบ subsystem descriptions (รวมถึง work entries หรือ routing entries ใดๆที่ใส่เพิ่ม) จากระบบ คิวงานที่ได้กำหนดที่ระบบย่อยนี้ โดยคำสั่ง Add Job Queue Entry (ADDJOBQE) จะไม่ถูกลบออกในข้อเท็จจริง 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 entry:

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 ต้องจบบระบบย่อยที่แฉีกทีฟ แล้วรีสตาร์ท

หลักการที่เกี่ยวข้อง

“การป้อนข้อมูลงานแบบ Autostart” ในหน้า 14

การป้อนข้อมูลงานแบบ autostart ระบุให้งานแบบ autostart เริ่มขึ้นในทันทีที่ระบบย่อยเริ่มต้นขึ้น. เมื่อระบบย่อยเริ่มทำงาน, ระบบจะจัดสรรสิ่งต่างๆ ให้ระบบย่อย และเริ่มงานแบบ autostart และ prestart ก่อนที่ระบบย่อยจะพร้อมที่จะทำงาน

การลบ communication entry:

คุณสามารถลบ communication entry ออกจาก subsystem description โดยใช้อินเตอร์เฟซแบบอักษร งานทั้งหมดที่แฉีกทีฟผ่านทาง communications entry ที่จะลบออกต้องถูกจบลงก่อน คำสั่งนี้จึงจะสามารถรันได้

คำสั่ง: Remove Communications Entry (RMVCMNE)

ตัวอย่าง: คำสั่งนี้จะลบ communications device entry สำหรับอุปกรณ์ COMDEV จาก subsystem description SBS1 ในไลบรารี LIB2

```
RMVCMNE  SBSDB(LIB2/SBS1)  DEV(COMDEV)
```

หลักการที่เกี่ยวข้อง

“การป้อนข้อมูลแบบ Communication” ในหน้า 15

การป้อนข้อมูลงานแบบ communications ระบุให้ระบบย่อยทราบถึงขอร์สสำหรับงานสื่อสารที่ระบบจะประมวลผล การประมวลผลงานเริ่มขึ้นเมื่อระบบย่อยได้รับ คำขอการเริ่มโปรแกรมการสื่อสารจากระบบรีโมตและพบ routing entry ที่เหมาะสมสำหรับคำขอ

การลบ job queue entry:

คุณสามารถลบ job queue entries ออกจาก subsystem description โดยใช้อินเตอร์เฟซแบบอักษร งานบนคิวงานยังคงอยู่บนคิว เมื่อ job queue entry ถูกลบออกจาก subsystem description จะไม่สามารถลบ job queue entry ถ้างานแฉีกทีฟในปัจจุบันใดๆ ถูกเริ่มจาก คิวงาน

คำสั่ง: Remove Job Queue Entry (RMVJOBQE)

ตัวอย่าง: คำสั่งนี้จะลบ job queue entry ที่อ้างอิงคิวงาน BATCH2 ใน MYLIB ออกจาก NIGHTRUN subsystem description ที่จัดเก็บในไลบรารี MYLIB

```
RMVJOBQE SBS(D:MYLIB/NIGHTRUN) JOBQ(MYLIB/BATCH2)
```

หลักการที่เกี่ยวข้อง

“Job queue entry” ในหน้า 71

job queue entry ระบุคิวงานซึ่งงานจะถูกเลือกสำหรับการรันในระบบย่อย พารามิเตอร์ใน job queue entry ที่ควบคุมวิธีการจัดการคิวงานมีอยู่ห้าพารามิเตอร์

“Job queue entry” ในหน้า 15

Job queue entry ใน subsystem description ระบุ คิวงานซึ่งระบบย่อยจะได้รับงานมา เมื่อระบบย่อยถูกเริ่มขึ้น ระบบย่อยจะพยายามจัดสรรคิวงานแต่ละคิวที่กำหนดใน subsystem job queue entry

งานที่เกี่ยวข้อง

“การกำหนดคิวงานที่ระบบย่อย” ในหน้า 207

ถ้าต้องการกำหนด job queue entry ที่ subsystem description ให้ใช้ อินเทอร์เน็ตแบบอักษระ

การลบ prestart job entry:

คุณสามารถลบ prestart job entries ออกจาก subsystem description โดยใช้ อินเทอร์เน็ตแบบอักษระ จะไม่สามารถลบ prestart job entry ถ้างานแ็คทีฟในปัจจุบันใดๆ ถูกเริ่มโดยใช้ entry

เมื่อลบ entry ที่มีการระบุ *LIBL เป็นชื่อไลบรารี โปรแกรมที่มีชื่อที่ระบุจะถูกค้นหาในรายชื่อไลบรารี ถ้าพบโปรแกรมใน รายชื่อไลบรารี 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 SBS(D:QGPL/PJE) PGM(QGPL/PGM1)
```

หลักการที่เกี่ยวข้อง

“Prestart job entry” ในหน้า 54

คุณกำหนดงานแบบ prestart โดยใช้ prestart job entry prestart job entry ไม่มีผลต่อการจัดสรรอุปกรณ์หรือการกำหนดค่าของการเริ่มโปรแกรม

“การสืบสวนงานแบบ prestart” ในหน้า 244

หัวข้อนี้นำเสนอขั้นตอนที่ช่วยคุณในการตอบคำถาม “ฉันจะ ค้นหาผู้ใช้จริงของงานแบบ prestart และกำหนดรีซอร์สที่ใช้โดยงานแบบ prestart นั้นได้อย่างไร?”

ข้อมูลที่เกี่ยวข้อง

รายงานจากประสบการณ์: การปรับ prestart job entry

การลบ routing entry:

คุณสามารถลบ routing entry ออกจาก subsystem description ที่ระบุได้โดยใช้ อินเทอร์เน็ตแบบอักษระ ระบบย่อยสามารถแ็คทีฟในเวลาที่ยังรัน คำสั่งได้ อย่างไรก็ดี จะไม่สามารถลบ routing entry ถ้ามีงานแ็คทีฟในปัจจุบันใดๆ ที่ถูกเริ่มต้นโดยใช้ entry

คำสั่ง: Remove Routing Entry (RMVRTGE)

ตัวอย่าง: คำสั่งนี้จะลบ routing entry 9912 ออกจาก subsystem description PERT ในไลบรารี OR

```
RMVRTGE SBSD(OR/PERT) SEQNBR(9912)
```

หลักการที่เกี่ยวข้อง

“routing entry” ในหน้า 17

routing entry ระบุพารามิเตอร์ของแหล่งเก็บข้อมูลหลักที่จะใช้, โปรแกรมการควบคุมที่จะรัน (โดยปกติคือโปรแกรมที่ระบบจัดให้ QCMD), และข้อมูลรันใหม่เพิ่มเติม (จัดเก็บอยู่ในคลาสอ็อบเจกต์) Routing entries ถูกจัดเก็บใน subsystem description

การลบ workstation entry:

คุณสามารถลบ workstation entry ออกจาก subsystem description โดยใช้อินเตอร์เฟซแบบอักษร ระบบย่อยสามารถแอ็คทีฟในเวลาที่ยัง คำสั่งได้ อย่างไรก็ตาม งานทั้งหมดที่แอ็คทีฟผ่านทาง workstation entry ต้องถูกจบก่อน entry จึงจะสามารถลบออกได้

คำสั่ง: Remove Work Station Entry (RMVWSE)

ตัวอย่าง: ตัวอย่างนี้จะลบ workstation entry สำหรับเวิร์กสเตชัน B53 ออกจาก subsystem description ที่มีชื่อว่า CHARLES ในไลบรารี LIB2

```
RMVWSE SBSD(LIB2/CHARLES) WRKSTN(B53)
```

หลักการที่เกี่ยวข้อง

“Workstation entries” ในหน้า 17

งานแบบโต้ตอบคืองานที่เริ่มขึ้นเมื่อผู้ใช้ signs on ที่จอภาพ และสิ้นสุดลงเมื่อผู้ใช้ signs off สำหรับงานที่จะรัน ระบบย่อยจะคำนวณรายละเอียดของงาน ซึ่งอาจจะอยู่ใน workstation entry หรือโปรไฟล์ผู้ใช้

การตั้งค่าระบบย่อยแบบโต้ตอบ

ข้อมูลในส่วนนี้อธิบายถึงวิธีการตั้งค่าระบบย่อยแบบโต้ตอบ

ขั้นตอนเหล่านี้ใช้กับวิธีการป้อนคำสั่งด้วยตนเอง อย่างไรก็ตาม คุณสามารถสร้างค่าติดตั้งของคุณใหม่เพื่อจุดประสงค์ในการกู้คืน โดยใช้โปรแกรม CL ในการสร้างระบบย่อยของคุณ

เมื่อคุณตั้งค่า ระบบย่อยแบบโต้ตอบใหม่ คุณควรพิจารณาถึงจำนวนอุปกรณ์ที่จะถูกจัดสรรให้ที่ระบบย่อยนั้น เนื่องจากระบบย่อยดำเนินการฟังก์ชันการจัดการอุปกรณ์ เช่น การแสดงจอแสดงผล sign-on และการจัดการกู้คืนความผิดพลาดของอุปกรณ์ คุณอาจต้องจำกัดจำนวนของอุปกรณ์ที่จัดสรรที่ระบบย่อยเดี่ยวแล้ว ให้ดูหัวข้อ Communications limits สำหรับรายละเอียดเพิ่มเติม

หมายเหตุ: หัวข้อนี้นำเสนอบทสรุปเกี่ยวกับสิ่งที่เกี่ยวข้องในการตั้งค่าระบบย่อยแบบโต้ตอบ รายงานจากประสบการณ์เกี่ยวกับระบบย่อยมีคำอธิบายอย่างละเอียดของแต่ละ ขั้นตอนและอ็อปชันเพิ่มเติมที่พร้อมใช้งานสำหรับแต่ละขั้นตอน

การสร้างไลบรารี:

ตัวอย่างนี้แสดงถึงวิธีการสร้างไลบรารีเพื่อจัดเก็บ configuration object ของระบบย่อยของคุณ

ตัวอย่างนี้ใช้ SBSLIB เป็นไลบรารี

```
CRTLIB SBSLIB TEXT('LIBRARY TO HOLD SUBSYSTEM CONFIGURATION OBJECTS')
```

การสร้างคลาส:

คลาสกำหนดลักษณะประสิทธิภาพสำหรับระบบย่อยแบบโต้ตอบของคุณ ปฏิบัติตามคำแนะนำนี้เพื่อสร้างคลาส

เพื่อสร้างคลาสที่เหมือนกับคลาส QINTER ให้ป้อน คำสั่งต่อไปนี้:

```
CRTCLS SBSLIB/INTER1 RUNPTY(20) TIMESLICE(2000) PURGE(*YES) DFTWAIT(30)  
TEXT('Custom Interactive Subsystem Class')
```

คุณสามารถใช้คลาส QINTER ใน QGPL สำหรับระบบย่อยแบบโต้ตอบที่ปรับตามความต้องการของคุณ หรือคุณสามารถสร้างคลาสเดียวเพื่อใช้สำหรับระบบย่อยแบบโต้ตอบทั้งหมดของคุณ หรือคุณสามารถสร้างหนึ่งคลาสสำหรับแต่ละระบบย่อยแบบโต้ตอบ

ตัวเลือกของคุณ ควรขึ้นอยู่กับว่า คุณต้องการจะปรับการตั้งค่าประสิทธิภาพสำหรับระบบย่อย ตามความต้องการของคุณหรือไม่ ระบบย่อยที่ได้มาจาก IBM ถูกจัดส่งมาพร้อมกับคลาส ที่สร้างขึ้นสำหรับแต่ละระบบย่อย โดยมีชื่อของคลาสเหมือนกับชื่อของระบบย่อย

ถ้าคุณไม่ต้องการสร้างคลาสสำหรับแต่ละระบบย่อยด้วยชื่อเดียวกับระบบย่อย คุณจะต้องระบุชื่อคลาสนำส่ง Add Routing Entry (ADDRTGE) ที่เป็นเช่นนี้เพราะดีฟอลต์สำหรับพารามิเตอร์ CLS คือ *SBSD ซึ่งหมายความว่า ชื่อคลาสมีชื่อเหมือนกับ subsystem description

การสร้าง subsystem description:

สำหรับแต่ละระบบย่อยที่คุณต้องการกำหนด ให้ทำขั้นตอนนี้ซ้ำเพื่อสร้าง subsystem description

ตัวอย่างต่อไปนี้จะสร้าง subsystem description ด้วยแอ็ททริบิวต์เฉพาะของ QINTER

```
CRTSBSD SBSD(SBSLIB/INTER1) POOLS((1 *BASE) (2 *INTERACT)) SGNDSPF(*QDSIGNON)
```

การสร้างคิวงาน:

คุณสามารถสร้างคิวงานสำหรับระบบย่อยโดยใช้ชื่อเดียวกับชื่อระบบย่อย และใส่เพิ่ม job queue entry ที่ subsystem description

ขั้นตอนนี้เป็นขั้นตอนบังคับ ถ้าคุณต้องการจะใช้คำสั่ง Transfer Job (TFRJOB) เพื่อถ่ายโอนงานเข้าในระบบย่อยที่ปรับตามความต้องการของคุณ

```
CRTJOBQ JOBQ(SBSLIB/INTER1)  
ADDJOBQE SBSD(SBSLIB/INTER1) JOBQ(SBSLIB/INTER1) MAXACT(*NOMAX)
```

การใส่เพิ่ม routing entry:

Routing entry ที่จัดให้มาพร้อมกับระบบสำหรับ QINTER มีฟังก์ชันเพิ่มเติมบางอย่าง ถ้าคุณต้องการฟังก์ชันเหล่านั้น ให้ใส่เพิ่ม routing entry ที่ subsystem description ที่ปรับตามความต้องการของคุณ

ปฏิบัติตามขั้นตอนนี้เพื่อใส่เพิ่ม routing entry:

ADDRTGE SBSDB(SBSLIB/INTER1) SEQNBR(9999) CMPVAL(*ANY) PGM(QSYS/QCMD) POOLID(2)

การใส่เพิ่ม workstation entry:

การใส่เพิ่ม workstation entry ใน subsystem description เป็นขั้นตอนหลักสำหรับการกำหนดว่าอุปกรณ์จะถูกจัดสรรที่ระบบย่อยใด

คุณต้องกำหนดว่าระบบย่อยใดควรจัดสรรอุปกรณ์ใด (AT(*SIGNON)) นอกจากนั้น 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

การปรับ QINTER:

เมื่อคุณเริ่มใช้ชุดของระบบย่อยของคุณเอง คุณอาจไม่ต้องการ ใช้ QINTER อย่างไรก็ดี 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 ไม่ให้ พยายามจัดสรรเวิร์กสเตชัน ทั้งหมด
2. ใส่เพิ่ม workstation entry สำหรับอุปกรณ์ที่มีชื่อว่า DSP* เพื่ออนุญาตให้อุปกรณ์แสดงผล twinax-attached ทั้งหมดถูกจัดสรรที่ QINTER ต่อไป

ในตัวอย่างนี้, อุปกรณ์แสดงผล twinax-attached จะรันใน QINTER ต่อไป; QINTER จะไม่พยายามจัดสรรอุปกรณ์อื่น

```
RMWSE SBSDB(QGPL/QINTER) WRKSTNTYPE(*ALL)
ADDWSE SBSDB(QGPL/QINTER) WRKSTN(DSP*)
```

วิธีที่สอง

ใส่เพิ่ม workstation entry เพื่อบอกให้ QINTER ไม่จัดสรรอุปกรณ์ ที่ถูกจัดสรรไปที่ระบบย่อยอื่น อย่างไรก็ดี อนุญาตให้ QINTER จัดสรรอุปกรณ์อื่น ที่ไม่ได้ถูกจัดสรรไปที่ระบบย่อยได้ต่อไป สิ่งนี้จะเก็บ workstation type entry ของ *ALL ในระบบย่อย QINTER และใส่เพิ่ม workstation name entry ที่มีพารามิเตอร์ AT ของอุปกรณ์เหล่านั้นที่ถูกจัดสรรไปที่ระบบย่อยอื่น

```
ADDWSE SBSDB(QGPL/QINTER) WRKSTN(PGMR*) AT(*ENTER)
ADDWSE SBSDB(QGPL/QINTER) WRKSTN(ORDERENT*) AT(*ENTER)
```

หมายเหตุ: คุณไม่สามารถใช้วิธีนี้ได้ ถ้าจำนวน device description บนระบบของคุณเกินกว่าจำนวนสูงสุดที่ระบบย่อยเดียวจะสามารถจัดการได้

การตั้งค่าคอนโซล:

ข้อควรพิจารณาสุดท้ายเกี่ยวกับ QINTER ซึ่งมีความสำคัญมากคือ workstation type entry ของ *CONS สำหรับคอนโซล ตรวจสอบให้แน่ใจว่าคุณไม่ได้ป้องกันไม่ให้บุคคลใด signing on ที่คอนโซลโดยไม่ได้ตั้งใจ คุณอาจป้องกันสิ่งนี้โดยการไม่ได้ใส่ workstation entries ใดๆ สำหรับคอนโซลที่ระบบย่อย แบบโต้ตอบที่ปรับตามความต้องการของคุณ

ระบบที่จัดส่งมาพร้อมกับระบบย่อยการควบคุมมี workstation entry เป็น AT(*SIGNON) สำหรับคอนโซล (*CONS workstation type entry) QINTER มี AT(*ENTER) workstation type entry สำหรับคอนโซล

วิธีปฏิบัติ ที่ดีคือ ควรรันคอนโซลในระบบย่อยการควบคุมเสมอ และไม่ถ่ายโอนงานคอนโซลเข้าในระบบย่อยแบบโต้ตอบอื่น การทำเช่นนี้ ช่วยป้องกันผู้ใช้ที่คอนโซลไม่ให้จบงานของตนโดยไม่ได้ตั้งใจ

ตัวอย่างเช่น ถ้าผู้ใช้ที่คอนโซลถ่ายโอนงานของตนไปที่ INTER1 แล้วลืม และในภายหลังเตรียมการสำรองข้อมูลโดยใช้คำสั่ง End System (ENDSYS) งานคอนโซลจะจบลงด้วย ซึ่งไม่ใช่สิ่งที่ผู้ควบคุมเครื่องตั้งใจ

การกำหนดผู้ใช้ที่ระบบย่อยเฉพาะ:

คุณสามารถใช้เทคนิคหลายอย่างในการกำหนดชื่ออุปกรณ์ จากนั้น จึงเชื่อมโยงชื่ออุปกรณ์เหล่านั้นกับผู้ใช้ หลังจากเสร็จสมบูรณ์แล้ว คุณสามารถใช้ workstation entry เพื่อนำผู้ใช้ไปที่ระบบย่อยที่ต้องการ

ระบบมีหลักการตั้งชื่อดีฟอลต์ที่จะใช้สำหรับ เซสชันการแสดงผล ในบางครั้งหลักการตั้งชื่อนี้อาจไม่เพียงพอสำหรับการเราต์ workstation entry ในหลายระบบย่อยโดยใช้โปรไฟล์ผู้ใช้

คุณสามารถทำการเปลี่ยนแปลงบนระบบของคุณเพื่อเพิ่มประสิทธิภาพ ดีฟอลต์ของระบบ โดยการกำหนดและจัดการหลักการตั้งชื่ออุปกรณ์ของคุณเอง โดยสามารถทำได้หลายวิธี แต่ละวิธีมีข้อดีและข้อเสียแตกต่างกันไป

หลักการที่เกี่ยวข้อง

“วิธีการจัดสรรอุปกรณ์เวิร์กสเตชัน” ในหน้า 22

ระบบย่อยพยายามจัดสรรอุปกรณ์เวิร์กสเตชันทั้งหมดใน subsystem description สำหรับ AT(*SIGNON) workstation entries

ข้อมูลที่เกี่ยวข้อง

รายงานจากประสบการณ์: การคอนฟิกูเรชันระบบย่อย

การใช้ Telnet exit point program

การกำหนดค่าเริ่มต้นอุปกรณ์เทอร์มินัลและ terminal exit points:

การกำหนดค่าเริ่มต้นอุปกรณ์ Telnet และ Terminal Exit Point exit point ช่วยให้สามารถกำหนดชื่ออุปกรณ์ในทันทีที่โคลเอ็นต์ signing on ที่ระบบ

exit point แสดง client IP address และ ชื่อโปรไฟล์ผู้ใช้ (พร้อมกับข้อมูลเพิ่มเติม) จากนั้นคุณสามารถทำการแมป โคลเอ็นต์ของคุณเองไปที่ device description ที่ควรจะใช้สำหรับโคลเอ็นต์

exit point การกำหนดค่าเริ่มต้นของอุปกรณ์ยังนำเสนอวิธีการข้าม sign-on panel ด้วย

ข้อดีของการใช้ exit point เหล่านี้ในการจัดการหลักการตั้งชื่ออุปกรณ์แบบเดิมของคุณ คือ คุณมีการควบคุมศูนย์กลางบนระบบของคุณสำหรับโคลเอ็นต์ทั้งหมดของคุณ

ข้อเสียคือต้องใช้ทักษะการทำโปรแกรมมิง

Exit point การเลือกอุปกรณ์:

exit point ช่วยให้คุณสามารถระบุหลักการตั้งชื่อที่ใช้สำหรับอุปกรณ์เสมือน และคอนโทรลเลอร์เสมือนที่สร้างขึ้นโดยอัตโนมัติ และสามารถระบุข้อจำกัดในการสร้างอย่างอัตโนมัติที่ใช้สำหรับคำขอพิเศษ

ด้วย exit point นี้ คุณสามารถระบุหลักการตั้งชื่อต่างๆ สำหรับอุปกรณ์ที่สร้างขึ้นโดยอัตโนมัติซึ่งใช้โดย Telnet, 5250 Display Station Pass-through, และ virtual terminal APIs

นอกจากนั้น คุณสามารถจัดการอุปกรณ์ Pass-through และกำหนดค่าของระบบ Telnet (QAUTOVRT) ได้อย่างถูกต้องมากขึ้น ตัวอย่างเช่น คุณอาจใช้ค่าหนึ่งสำหรับอุปกรณ์ที่สร้างขึ้นโดยอัตโนมัติสำหรับ Telnet และใช้อีกค่าหนึ่งสำหรับอุปกรณ์ 5250 Display Station Pass-through

Exit point นี้ช่วยให้คุณสามารถควบคุมหลักการตั้งชื่อดีฟอลต์ที่ใช้สำหรับอุปกรณ์ (เช่น QPADEV*) แต่ exit point เพียงอย่างเดียวไม่ทำให้คุณสามารถระบุอุปกรณ์เฉพาะสำหรับผู้ใช้งานเฉพาะได้ Exit point นี้มีประโยชน์มากที่สุด ถ้าคุณกำลังใช้วิธีการหลายอย่างในการเชื่อมต่อกับระบบ (Telnet, 5250 Display Station Pass-through, WebFacing, และอื่นๆ) เนื่องจาก exit point นี้อนุญาตให้คุณใช้หลักการตั้งชื่อ อุปกรณ์ที่แตกต่างกันได้และมีการจัดการ QAUTOVRT ที่แม่นยำสำหรับวิธีการใช้งานต่างๆ

การสนับสนุน PC5250 (System i Access) workstation ID:

คุณสามารถตั้งค่า System i Access เพื่อเชื่อมต่อกับชื่อเวิร์กสเตชันเฉพาะ ถ้าคุณคลิกปุ่มคำอธิบายจากหน้าต่งนี้ อี้อพชั่นหลากหลายสำหรับการระบุ workstation ID เช่น การร่างชื่อใหม่ ถ้าชื่อที่ระบุถูกใช้อยู่แล้ว 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 คุณสามารถสร้างเวิร์กสเตชันที่เลือกกำหนดคำสั่ง STRTCPTELN เพื่อให้แน่ใจถึง remote virtual terminal display value และเรียกใช้คำสั่งที่ IBM จัดให้

การสร้างคอนโทรลเลอร์และอุปกรณ์เสมือนด้วยตนเอง:

คุณสามารถสร้างคอนโทรลเลอร์และอุปกรณ์เสมือนด้วยตนเองได้

สำหรับข้อมูลเพิ่มเติมเกี่ยวกับการสร้างอุปกรณ์เสมือนสำหรับ Telnet ให้ดูหัวข้อ Configure the Telnet Server ใน i5/OS Information Center

ในวิธีนี้ คุณสามารถควบคุมชื่อของคอนโทรลเลอร์และอุปกรณ์ของคุณได้ แต่คุณไม่สามารถแม้อุปกรณ์เฉพาะกับ ผู้ใช้เฉพาะ

การสร้างระบบย่อยการควบคุม

IBM จัดเตรียม controlling subsystem configuration ที่สมบูรณ์สองรายการ: QBASE (ระบบย่อยการควบคุมดีฟอลต์) และ QCTL ในแต่ละครั้ง สามารถมีระบบย่อยการควบคุมที่แอ็คทีฟ บนระบบได้เพียงหนึ่งระบบย่อยเท่านั้น โดยปกติ คอนโซลที่ผู้ใช้ระบบย่อยที่จัดให้โดย IBM ควรเพียงพอสำหรับความต้องการทางธุรกิจส่วนใหญ่ 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 ประกอบไปด้วย:
 - *ANY หรือ QCMDI อย่างใดอย่างหนึ่ง เป็นข้อมูลการเรอต์
 - QSYS/QCMD เป็นโปรแกรมที่จะเรียกใช้
 - คลาส QSYS/QCTL หรือคลาสที่ผู้ใช้กำหนด (ที่เป็นเช่นนี้เนื่องจากผู้ใช้โดยทั่วไปคือ ผู้ควบคุมระบบ ต้องสามารถป้อนคำสั่งเพื่อทำบางสิ่ง เช่น ให้มีหน่วยความจำมากขึ้น ถ้ามีการใช้งานถึง threshold ของหน่วยความจำรอง)
- Workstation entry สำหรับคอนโซลที่มีชนิด *SIGNON (*SIGNON เป็นค่าสำหรับพารามิเตอร์ AT ที่ระบุบนคำสั่ง Add Work Station Entry (ADDWSE))

ค่า *SIGNON บ่งชี้ว่าหน้าจอ sign-on จะถูกแสดงขึ้นที่เวิร์กสเตชัน เมื่อระบบย่อยเริ่มต้นขึ้น ข้อกำหนดนี้ทำให้มั่นใจได้ว่าระบบย่อยมีอุปกรณ์โต้ตอบสำหรับการป้อนคำสั่งระดับระบบ และระบบย่อย คำสั่ง End System (ENDSYS) จะจบ i5/OS โลเซนส์โปรแกรมเป็น single session (หรือหน้าจอ sign-on) ที่คอนโซลในระบบย่อยการควบคุม subsystem description ที่ไม่มี workstation entry สำหรับคอนโซลไม่สามารถถูกเริ่มเป็นระบบย่อยการควบคุม

- entry สำหรับเวิร์กสเตชันอื่น:

สิ่งนี้นำเสนอซอร์สทางเลือก ของอินพุตการควบคุม หากตรวจพบปัญหาเกี่ยวกับคอนโซลในระหว่าง attended IPL และถ้าค่ากำหนดของระบบ 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 ให้เปลี่ยนค่ากำหนดของระบบ Controlling subsystem/library (QCTLSBSD) เป็นดังนี้ (สมมติให้ description มีชื่อว่า QGPL/QCTLA):

```
CHGSYSVAL SYSVAL(QCTLSBSD) VALUE('QCTLA QGPL')
```

การเปลี่ยนแปลง จะมีผลบังคับใช้ที่ IPL ถัดไป

หลักการที่เกี่ยวข้อง

“ระบบย่อยการควบคุม” ในหน้า 11

ระบบย่อยการควบคุมคือ ระบบย่อยแบบโต้ตอบที่เริ่มทำงาน โดยอัตโนมัติเมื่อระบบเริ่มขึ้น และเป็นระบบย่อยซึ่งผู้ควบคุมระบบ ใช้ในการควบคุมระบบผ่านทางคอนโซลของระบบ ระบบย่อยการควบคุมถูกระบุ ในคำกำหนดของระบบ Controlling subsystem/library (QCTLSBSD).

ข้อมูลที่เกี่ยวข้อง

รายงานจากประสบการณ์: สภาวะที่ถูกควบคุม

การวางระบบในสภาพที่ถูกจำกัด

ถ้าระบบย่อยทั้งหมด รวมถึงระบบย่อยการควบคุม จบลง ระบบจะอยู่ในสภาพถูกจำกัด คุณสามารถวางระบบในสภาพที่ถูกจำกัดได้ โดยการใช้หนึ่งหรือสองคำสั่งจากเวิร์กสเตชัน แบบโต้ตอบ

คำสั่ง: End Subsystem พร้อมกับพารามิเตอร์ *ALL (ENDSBS SBS(*ALL))

คำสั่ง: End System (ENDSYS)

สำคัญ: คำสั่ง ENDSBS หรือ ENDSYS ควรออกใช้จากงานแบบโต้ตอบในระบบย่อยการควบคุม และเฉพาะจากเวิร์กสเตชันที่มี entry ใน controlling subsystem description ระบบ AT(*SIGNON) งานแบบโต้ตอบซึ่งคำสั่งถูกออกใช้จากที่นั่น ยังคงแอคทีฟ เมื่อระบบย่อยการควบคุมเข้าสู่สภาพถูกจำกัด ถ้างานที่ออกคำสั่งเป็นหนึ่งในสองงานที่แอคทีฟที่เวิร์กสเตชัน (การใช้ปุ่ม System Request หรือคำสั่ง TFRSECJOB) งานทั้งสองจะไม่ถูก force ให้จบ อย่างไรก็ตาม ระบบย่อยการควบคุมจะไม่จบ สภาพที่ถูกจำกัด จนกว่าคุณจะทำงานใดงานหนึ่ง งานกลุ่มการหยุดชั่วคราว ยังป้องกันไม่ให้ระบบย่อยการควบคุมจบ (จนกว่างานกลุ่ม จะจบลง)

เมื่อระบบอยู่ในสภาพที่ถูกจำกัด, activity ส่วนใหญ่บนระบบจะจบลง, และมีเวิร์กสเตชันที่แอคทีฟเพียงเวิร์กสเตชันเดียวเท่านั้น ระบบต้องอยู่ในสภาพนี้สำหรับการรันคำสั่ง เช่น Save System (SAVSYS) หรือ Reclaim Storage (RCLSTG)

โปรแกรมการวินิจฉัยปัญหา เกี่ยวกับอุปกรณ์บางโปรแกรมอาจต้องการให้ระบบอยู่ในสภาพที่ถูกจำกัดด้วย เพื่อจบสภาพที่ถูกจำกัด คุณต้องเริ่มระบบย่อยการควบคุม อีกครั้ง

หลักการที่เกี่ยวข้อง

“ระบบย่อยการควบคุม” ในหน้า 11

ระบบย่อยการควบคุมคือ ระบบย่อยแบบโต้ตอบที่เริ่มทำงาน โดยอัตโนมัติเมื่อระบบเริ่มขึ้น และเป็นระบบย่อยซึ่งผู้ควบคุมระบบ ใช้ในการควบคุมระบบผ่านทางคอนโซลของระบบ ระบบย่อยการควบคุมถูกระบุ ในคำกำหนดของระบบ Controlling subsystem/library (QCTLSBSD).

ข้อมูลที่เกี่ยวข้อง

รายงานจากประสบการณ์: สภาวะที่ถูกควบคุม

การจัดการพูลหน่วยความจำ

การตรวจสอบให้แน่ใจว่างานมีหน่วยความจำเพียงพอที่จะดำเนินงานอย่างมีประสิทธิภาพถือเป็นสิ่งที่สำคัญ ถ้ามีการให้หน่วยความจำกับระบบย่อย A มากเกินไปและไม่เพียงพอที่ระบบย่อย B งานในระบบย่อย B อาจเริ่มรันได้ไม่ดี ข้อมูลต่อไปนี้ อธิบายงานหลากหลายที่เกี่ยวข้องกับการจัดการ พูลหน่วยความจำ

หลักการที่เกี่ยวข้อง

“พูลหน่วยความจำ” ในหน้า 24

พูลหน่วยความจำเป็นส่วนโลจิคัลของหน่วยความจำหลักหรือหน่วยเก็บข้อมูลที่สงวนไว้สำหรับการประมวลผลงานหรือกลุ่มของงาน บนระบบของคุณ หน่วยเก็บข้อมูลหลักสามารถจัดสรรแบบโลจิคัลที่เรียกว่าพูลหน่วยความจำได้โดยค่าเริ่มต้น คือ ระบบจะจัดการถ่ายโอนข้อมูลและโปรแกรมเข้าในพูลหน่วยความจำ .

ข้อมูลที่เกี่ยวข้อง

การจัดการประสิทธิภาพระบบ

การปรับประสิทธิภาพระดับต้น

แอปพลิเคชันสำหรับการจัดการประสิทธิภาพ

รายงานจากประสบการณ์: Performance Adjuster (QPFRADJ)

การดูข้อมูลพูลหน่วยความจำ

คุณสามารถดูข้อมูลเกี่ยวกับพูลหน่วยความจำที่มีอยู่บนระบบของคุณได้โดยใช้ System i Navigator หรืออินเตอร์เฟซแบบอักษร

หลักการที่เกี่ยวข้อง

“การจัดสรรพูลหน่วยความจำ” ในหน้า 29

เมื่อคุณเริ่มระบบย่อย ระบบจะพยายามจัดสรรพูลของหน่วยความจำที่ผู้ใช้กำหนด ซึ่งถูกกำหนดใน subsystem description ของ ระบบย่อยที่เริ่มต้น

“activity level ของพูลหน่วยความจำ” ในหน้า 30

activity level ของพูลหน่วยความจำ คือ จำนวนของ thread ที่สามารถใช้ CPU ในสถานะแอกทีฟได้ในเวลาเดียวกันในพูลหน่วยความจำ สิ่งนี้ช่วยให้ สามารถใช้รีซอร์สของระบบได้อย่างมีประสิทธิภาพ ระบบจัดการการควบคุมของ activity level

ข้อมูลที่เกี่ยวข้อง

การจัดการประสิทธิภาพระบบ

การปรับประสิทธิภาพระดับต้น

แอปพลิเคชันสำหรับการจัดการประสิทธิภาพ

รายงานจากประสบการณ์: Performance Adjuster (QPFRADJ)

System i Navigator:

จาก System i Navigator ให้ขยาย **My Connections** → **connection** → **Work Management** → **Memory Pools** → **Active Pools** หรือ **Shared Pools**.

Active Pools container แสดงทั้ง shared และ private pools トラバเท่าที่พูลยังแอกทีฟอยู่ Shared Pools container แสดง shared pools ทั้งหมดไม่ว่าสถานะปัจจุบันของพูลจะเป็นอะไรก็ตาม Inactive private pools ไม่ มีอยู่นอก pool definition จนกว่าพูลจะถูกเรียกทำงานโดยระบบย่อย ดังนั้น จึงไม่สามารถดูพูลโดยใช้ System i 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 หลังจากที่ทราบจำนวนของระบบย่อยที่กำลังส่งงาน ไปที่พูลและจำนวนของงานที่กำลังรันในพูล คุณอาจต้องการลด contention ริชอร์สโดยการปรับเปลี่ยนขนาดและ activity level ของพูล

หลักการที่เกี่ยวข้อง

“การจัดสรรพูลหน่วยความจำ” ในหน้า 29

เมื่อคุณเริ่มระบบย่อย ระบบจะพยายามจัดสรรพูลของหน่วยความจำที่ใช้กำหนด ซึ่งถูกกำหนดใน subsystem description ของ ระบบย่อยที่เริ่มต้น

“activity level ของพูลหน่วยความจำ” ในหน้า 30

activity level ของพูลหน่วยความจำ คือ จำนวนของ thread ที่สามารถใช้ CPU ในสถานะแอดคทีฟได้ในเวลาเดียวกันในพูลหน่วยความจำ สิ่งนี้ช่วยให้สามารถใช้ริชอร์สของระบบได้อย่างมีประสิทธิภาพ ระบบจัดการการควบคุมของ activity level

ข้อมูลที่เกี่ยวข้อง

การจัดการประสิทธิภาพระบบ

การปรับประสิทธิภาพระดับต้น

แอ็พพลิเคชันสำหรับการจัดการประสิทธิภาพ

รายงานจากประสบการณ์: Performance Adjuster (QFPRADJ)

System i Navigator:

ในการใช้ System i Navigator เพื่อมอนิเตอร์จำนวนของระบบย่อยที่กำลังใช้พูลหน่วยความจำ ให้ปฏิบัติตามคำแนะนำต่อไปนี้:

1. จาก System i Navigator ให้ขยาย My Connections → Connection → Work Management → Memory Pools → Active Pools or Shared Pools.
2. คลิกขวาพูลหน่วยความจำที่คุณต้องการทำงานด้วยและคลิก Subsystems.
จากหน้าต่างนี้ คุณสามารถกำหนดจำนวนของระบบย่อยที่กำลังใช้แต่ละหน่วยความจำเพื่อรันงานของตน

อินเตอร์เฟซแบบอักษร:

คำสั่ง: Work with Subsystems (WRKSBS)

คำสั่งนี้แสดงรายการของระบบย่อยทั้งหมดและพูลของระบบย่อย

การกำหนดจำนวนของงานในพูลหน่วยความจำ

System i Navigator นำเสนอวิธีการแสดงรายการของงานซึ่งอยู่ในพูลหน่วยความจำอย่างรวดเร็ว

ถ้าต้องการกำหนดจำนวนของงานในพูลหน่วยความจำ ให้ปฏิบัติตามคำแนะนำต่อไปนี้:

1. จาก System i Navigator ให้ขยาย My Connections → connection → 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 คุณสามารถใช้ฟังก์ชันเกี่ยวกับงานเดียวกันได้เสมือนคุณได้อยู่ในพื้นที่ Active jobs หรือ Server jobs
หลักการที่เกี่ยวข้อง

“การจัดสรรพูลหน่วยความจำ” ในหน้า 29

เมื่อคุณเริ่มระบบย่อย ระบบจะพยายามจัดสรรพูลของหน่วยความจำ ที่ผู้ใช้กำหนด ซึ่งถูกกำหนดใน subsystem description ของ ระบบย่อยที่เริ่มต้น

“activity level ของพูลหน่วยความจำ” ในหน้า 30

activity level ของพูลหน่วยความจำ คือ จำนวนของ thread ที่สามารถใช้ CPU ในสถานะแอดคทีฟได้ในเวลาเดียวกันในพูลหน่วยความจำ สิ่งนี้ช่วยให้สามารถใช้รีซอร์สของระบบได้อย่างมีประสิทธิภาพ ระบบจัดการการควบคุมของ activity level
ข้อมูลที่เกี่ยวข้อง

การจัดการประสิทธิภาพระบบ

การปรับประสิทธิภาพระดับต้น

แอ็พพลิเคชันสำหรับการจัดการประสิทธิภาพ

รายงานจากประสพการณ์: Performance Adjuster (QPFRADJ)

การกำหนดพูลที่งานเดี่ยวกำลังรันอยู่

ถ้าคุณมีงานที่ไม่ได้ปฏิบัติเนื่องจากคุณคาดว่าอาจต้อง ตรวจสอบพูลหน่วยความจำซึ่งงานกำลังรันอยู่ ในการกำหนดพูลที่งานเดี่ยวกำลังรันอยู่ให้ใช้ System i Navigator หรืออินเตอร์เฟซแบบอักษร

หลังจากที่คุณได้ระบุพูลซึ่งงานกำลังรันอยู่แล้ว คุณสามารถดูข้อมูลพูลหน่วยความจำและกำหนดว่าต้องทำการเปลี่ยนแปลงหรือไม่ ตัวอย่างเช่น ถ้าเกิดการเพจมากเกินไป พูลหน่วยความจำอาจต้องใหญ่ ขึ้น อีกสาเหตุหนึ่งที่ทำให้ประสิทธิภาพลดลงคือ อาจมีงานอื่นอยู่ใน พูลมากเกินไปและคุณต้องเราต์งานนี้ไปที่พูลอื่น

หลักการที่เกี่ยวข้อง

“การจัดสรรพูลหน่วยความจำ” ในหน้า 29

เมื่อคุณเริ่มระบบย่อย ระบบจะพยายามจัดสรรพูลของหน่วยความจำ ที่ผู้ใช้กำหนด ซึ่งถูกกำหนดใน subsystem description ของ ระบบย่อยที่เริ่มต้น

“activity level ของพูลหน่วยความจำ” ในหน้า 30

activity level ของพูลหน่วยความจำ คือ จำนวนของ thread ที่สามารถใช้ CPU ในสถานะแอดคทีฟได้ในเวลาเดียวกันในพูลหน่วยความจำ สิ่งนี้ช่วยให้สามารถใช้รีซอร์สของระบบได้อย่างมีประสิทธิภาพ ระบบจัดการการควบคุมของ activity level
ข้อมูลที่เกี่ยวข้อง

การจัดการประสิทธิภาพระบบ

การปรับประสิทธิภาพระดับต้น

แอ็พพลิเคชันสำหรับการจัดการประสิทธิภาพ

รายงานจากประสพการณ์: Performance Adjuster (QPFRADJ)

System i Navigator:

ในการใช้ System i Navigator ให้ใช้ข้อแนะนำต่อไปนี้:

1. จาก System i Navigator ให้ขยาย My Connections → connection → 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 ฟิลด์นี้จะเป็นพื้นที่เปล่าสำหรับงานที่ไม่แอ็ดที่พีในเวลาที่มีการร้องขอแสดงผล และยังเป็นพื้นที่เปล่าสำหรับงานระบบ (ชนิด SYS) งานการมอนิเตอร์ระบบย่อย (ชนิด SBS) ที่ไม่ได้อันภายในระบบย่อย และ batch immediate jobs (BCI) ที่กำลังรันอยู่ในพูลหน่วยความจำหลัก

คำสั่ง: Work with active job (WRKACTJOB)

คุณสามารถใช้คำสั่ง WRKACTJOB เพื่อดู system pool ID สำหรับงานแอ็ดที่พี

การจัดพารามิเตอร์การปรับสำหรับแชร์พูล

ถ้าต้องการจัดการพารามิเตอร์สำหรับแชร์พูล ให้ใช้ System i Navigator หรือคำสั่งอินเตอร์เฟซแบบอักษร

หลักการที่เกี่ยวข้อง

“แบบแผนการกำหนดหมายเลขพูล” ในหน้า 27

พูลมีแบบแผนการกำหนดหมายเลขอยู่สองชุด: ชุดหนึ่งใช้ภายใน ระบบย่อยและอีกชุดหนึ่งใช้ในระบบทั้งหมด ระบบย่อยจะใช้ชุดของหมายเลขที่ อ้างอิงถึงพูลที่ระบบย่อยใช้ ดังนั้น เมื่อคุณสร้างหรือเปลี่ยน subsystem description คุณสามารถกำหนดหนึ่งหรือหลายพูลและกำหนดหมายเลขพูลเป็น 1, 2, 3 และต่อไป นี่คือการกำหนดพูลระบบย่อย และไม่สอดคล้องกับหมายเลขพูลที่แสดงบนจอแสดงผล Work with System Status (WRKSYSSTS)

ข้อมูลที่เกี่ยวข้อง

การจัดการประสิทธิภาพระบบ

การปรับประสิทธิภาพระดับต้น

แอ็พพลิเคชันสำหรับการจัดการประสิทธิภาพ

รายงานจากประสบการณ์: Performance Adjuster (QPFRADJ)

System i Navigator:

ในการเข้าใช้งานพารามิเตอร์การปรับโดยใช้ System i Navigator ให้ปฏิบัติตามคำแนะนำต่อไปนี้:

1. จาก System i Navigator ให้ขยาย **My Connections** → **connection** → **Work Management** → **Memory Pools** → **Active Pools or Shared Pools**
2. คลิกขวาพูลซึ่งคุณต้องการปรับและคลิก **Properties**.
3. คลิกแท็บ **Tuning** .

จากหน้าต่าง Shared Properties – Tuning คุณสามารถปรับเปลี่ยนค่าเฉพาะ เช่น เปอร์เซนต์การจัดสรรพูล, page faults ต่อวินาที และระดับความสำคัญด้วยตนเองได้

อินเตอร์เฟซแบบอักษร:

คำสั่ง: Work with Shared Pool (WRKSHRPOOL)

เลือก อีพ ชั้น 11 - แสดงข้อมูลการปรับ.

การจัดการ configuration ของพูล

ถ้าต้องการเปลี่ยนขนาดของพูล, activity level หรืออีพชั้นการเพจ ให้ใช้ System i Navigator หรือคำสั่งอินเตอร์เฟซแบบอักษร

หลักการที่เกี่ยวข้อง

“แบบแผนการกำหนดหมายเลขพูล” ในหน้า 27

พูลมีแบบแผนการกำหนดหมายเลขอยู่สองชุด: ชุดหนึ่งใช้ภายใน ระบบย่อยและอีกชุดหนึ่งใช้ในระบบทั้งหมด ระบบย่อยจะใช้ชุดของหมายเลขที่ อ้างอิงถึงพูลที่ระบบย่อยใช้ ดังนั้น เมื่อคุณสร้างหรือเปลี่ยน subsystem description คุณสามารถกำหนดหนึ่งหรือหลายพูลและกำหนดหมายเลขพูลเป็น 1, 2, 3 และต่อไป นี่คือการกำหนดพูลระบบย่อย และไม่สอดคล้องกับหมายเลขพูลที่แสดงบนจอแสดงผล Work with System Status (WRKSYSSTS)

ข้อมูลที่เกี่ยวข้อง

การจัดการประสิทธิภาพระบบ

การปรับประสิทธิภาพระดับต้น

แอ็พพลิเคชันสำหรับการจัดการประสิทธิภาพ

รายงานจากประสบการณ์: Performance Adjuster (QPFRADJ)

System i Navigator:

ถ้าต้องการเข้าใช้งานค่า configuration ของแชร์พูลโดยใช้ System i Navigator, use the following instructions:

1. จาก System i Navigator ให้ขยาย My Connections → connection → Work Management → Memory Pools → Active Pools or Shared Pools.
2. คลิกขวาพูลซึ่งคุณต้องการปรับและคลิก Properties.
3. คลิกแท็บ Configuration .

จากหน้าต่าง Shared Properties – Configuration คุณสามารถปรับเปลี่ยน ค่าเฉพาะ เช่น ขนาดของพูล, activity level หรืออีพชั้นการเพจด้วยตนเองได้

อินเตอร์เฟซแบบอักษร:

คำสั่ง: Work with Shared Pool (WRKSHRPOOL)

การเปลี่ยนขนาดของพูลหน่วยความจำ

ขนาดของพูลหน่วยความจำมีผลโดยตรงกับปริมาณของงานที่ระบบย่อยสามารถประมวลผลได้ ระบบย่อยยังมีหน่วยความจำมาก, ยิ่งสามารถทำงานให้เสร็จ สมบูรณ์ได้มากขึ้น นับเป็นเรื่องสำคัญที่คุณควรมอนิเตอร์ระบบของคุณอย่างรอบคอบ ก่อนจะเริ่มการเปลี่ยนพารามิเตอร์ของพูลหน่วยความจำของคุณ คุณยังต้องตรวจสอบชำระระดับเหล่านี้เป็นระยะ เนื่องจากอาจต้องมีการปรับบางอย่างใหม่

ตรวจสอบให้แน่ใจว่าคุณได้ปิดตัวปรับระบบ ก่อนที่คุณจะเริ่มเปลี่ยนขนาดของ พูลหน่วยความจำด้วยตนเอง ตัวปรับระบบจะปรับขนาดของพูลหน่วยความจำที่แบ่ง ใช้โดยอัตโนมัติให้เหมาะสมกับปริมาณงานที่ระบบกำลังทำงานอยู่ ถ้าไม่ได้ ปิดตัวปรับระบบ การเปลี่ยนแปลงที่คุณทำด้วยตนเองอาจถูกเปลี่ยน โดยอัตโนมัติโดยตัวปรับ

ปิดตัวปรับระบบโดยการเปลี่ยนค่ากำหนดของระบบ Automatically adjust memory pools และ activity levels (QPFRADJ) เป็น 0 (0 = ไม่มีการปรับเปลี่ยน)

หลักการที่เกี่ยวข้อง

“แบบแผนการกำหนดหมายเลขพูล” ในหน้า 27

พูลมีแบบแผนการกำหนดหมายเลขอยู่สองชุด: ชุดหนึ่งใช้ภายใน ระบบย่อยและอีกชุดหนึ่งใช้ในระบบทั้งหมด ระบบย่อยจะใช้ชุดของหมายเลขที่อ้างอิงถึงพูลที่ระบบย่อยใช้ ดังนั้น เมื่อคุณสร้างหรือเปลี่ยน subsystem description คุณสามารถกำหนดหนึ่งหรือหลายพูลและกำหนดหมายเลขพูลเป็น 1, 2, 3 และต่อไป นี้คือ การกำหนดพูลระบบย่อย และไม่สอดคล้องกับหมายเลขพูลที่แสดงบนจอแสดงผล Work with System Status (WRKSYSSTS)

ข้อมูลที่เกี่ยวข้อง

การจัดการประสิทธิภาพระบบ

การปรับประสิทธิภาพระดับต้น

แอ็พพลิเคชันสำหรับการจัดการประสิทธิภาพ

รายงานจากประสบการณ์: Performance Adjuster (QPFRADJ)

System i Navigator:

1. จาก System i Navigator, ให้ขยาย My Connections → Connection → 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 คุณ สามารถเปลี่ยนจำนวนของหน่วยความจำที่กำหนดได้ หน่วยความจำที่กำหนดคือจำนวนสูงสุดของหน่วยความจำที่พูลสามารถใช้ได้ ตัวเลขที่คุณใส่ในที่นี้ควรจะเป็นขนาดหน่วยความจำที่คุณคิดว่าพูลจะต้องการใช้เพื่อสนับสนุนระบบย่อยที่พูลให้บริการอยู่

หมายเหตุ: Base pool คือพูลหน่วยความจำที่ไม่มีจำนวนของหน่วยความจำที่กำหนดไว้ แต่มีขนาดหน่วยความจำต่ำสุดที่จำเป็นในการใช้รัน Base pool ประกอบด้วยทุกสิ่งที่ไม่ได้ถูกจัดสรรไว้ในที่อื่น ตัวอย่างเช่น คุณอาจจะมีหน่วยความจำอยู่ 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 ซึ่งอาจทำให้ประสิทธิภาพโดยรวมของระบบลดลงด้วย

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

นี่สอดคล้องกับพูล 2 บนจอแสดงผล WRKSYSSTS

หมายเหตุ: ค่ากำหนดของระบบ QBASPOOL ควบคุมขนาดต่ำสุดของ base pool เท่านั้น Base pool มีหน่วยความจำทั้งหมดที่ไม่ได้ถูกจัดสรรให้กับพูลอื่น

การเปลี่ยนขนาดของ shared pool:

Command: Change Shared Storage Pool (CHGSHRPOOL)

การเปลี่ยนขนาดของ shared pools มีผลบังคับใช้ในทันที ถ้า shared pool แอ็คทีฟ และมีหน่วยความจำที่พร้อมใช้งานมากพอ

คำสั่ง: Work with Shared Storage Pools (WRKSHRPOOL)

คำสั่งนี้ช่วยให้คุณเข้าถึง ข้อมูลชื่อและสถานะของ shared pools โดยการใช้เมนูอ็อปชัน คุณสามารถเปลี่ยนค่าของขนาดพูลและ activity level สูงสุดได้

การสร้างพูลหน่วยความจำส่วนตัว

พูลหน่วยความจำส่วนตัว (เรียกอีกอย่างว่าพูลหน่วยความจำที่ผู้ใช้กำหนด) สามารถใช้โดยระบบย่อยที่ได้มาจาก IBM หรือโดยระบบย่อยที่ผู้ใช้กำหนด คุณสามารถกำหนด definition พูลหน่วยความจำได้สูงสุดถึง 10 definition สำหรับระบบย่อยหนึ่ง คุณสามารถสร้างพูลหน่วยความจำส่วนตัวใน subsystem description

ถ้าต้องการสร้างพูลหน่วยความจำส่วนตัว ให้ใช้อินเตอร์เฟซแบบอักขระ

คำสั่ง: Create Subsystem Description (CRTSBSD) POOLS parameter.

Command: Change Subsystem Description (CHGSBSD) POOLS parameter

หมายเหตุ: แม้ว่าแต่ละ subsystem description สามารถมีพูลหน่วยความจำที่ผู้ใช้กำหนดได้มากถึง 10 พูล แต่มีข้อจำกัดในการดำเนินการที่ว่า พูลหน่วยความจำที่สามารถแอ็คทีฟในเวลาใดๆ ไม่สามารถมีมากกว่า 64 พูล (ตัวเลขนี้รวมถึง Base memory pool และ Machine memory pool) ถ้าใช้งานครบตามขีดจำกัดการจัดสรรสูงสุด ก่อนจะมีการจัดสรรพูลหน่วยความจำทั้งหมดสำหรับระบบย่อย Base pool จะถูกใช้สำหรับขั้นตอนการเรอิตใดๆ ที่ยังคงต้องการพูลหน่วยความจำอยู่

หลักการที่เกี่ยวข้อง

“ชนิดของพูลหน่วยความจำ” ในหน้า 25

บนระบบของคุณ หน่วยเก็บข้อมูลหลักทั้งหมดสามารถจัดสรรแบบโลจิคัลที่เรียกว่า พูลหน่วยความจำ. หน่วยความจำทั้งหมดในระบบสามารถแบ่งเป็นแบบส่วนตัวหรือแบบแบ่งใช้ อย่างไรก็ตาม มีพูลหน่วยความจำส่วนตัว, พูลหน่วยความจำแบบแบ่งใช้ และพูลหน่วยความจำแบบแบ่งใช้พิเศษ พูลหน่วย ความจำสูงสุดทั้งหมด 64 พูล สามารถแอ็คทีฟได้ในเวลาเดียวกัน ไม่ว่าจะแบบ ส่วนตัวหรือแบบแบ่งใช้ก็ตาม

ข้อมูลที่เกี่ยวข้อง

การจัดการประสิทธิภาพระบบ

การปรับประสิทธิภาพระดับต้น

แอ็พพลิเคชันสำหรับการจัดการประสิทธิภาพ

รายงานจากประสิทธิภาพ: Performance Adjuster (QPFRADJ)

ค่ากำหนดของระบบประสิทธิภาพ: ขนาดพูลหน่วยความจำเครื่อง

ค่ากำหนดของระบบประสิทธิภาพ: ขนาดต่ำสุดของพูลหน่วยความจำหลัก

ค่ากำหนดของระบบประสิทธิภาพ: thread ที่มีลิสต์สูงสุดของพูลหน่วยความจำหลัก

การจัดการคิวงาน

เมื่อคุณจัดการงานบนระบบ คุณอาจพบว่ามีความจำเป็น ต้องดำเนินการงานที่กำลังรออยู่ในคิวงาน บางทีอาจมีบางคน ต้องการ รันงานในทันที แต่งานรออยู่ในคิวโดยมีระดับความสำคัญต่ำ หรือบางทีคุณต้องการปฏิบัติการบำรุงรักษาบางอย่าง บนระบบย่อย และต้องการย้ายงานทั้งหมดไปที่คิวซึ่งไม่ได้เชื่อมโยงกับระบบย่อยนั้น

ข้อมูลต่อไปนี้อธิบายวิธีการดำเนินการงานการจัดการเหล่านี้

หลักการที่เกี่ยวข้อง

“คิวงาน” ในหน้า 68

คิวงานประกอบด้วยรายการลำดับของงาน ที่รอการประมวลผลโดยระบบย่อย คิวงานจะเป็นที่แรก ที่งานแบบแบดซ์ที่ส่ง ไปพักรอ ก่อนที่จะไปทำงาน อยู่ในระบบย่อย งานจะถูกพักอยู่ที่นั่นจนกว่าจะมีสภาพตรงตามเงื่อนไขการทำงานต่างๆ

การกำหนดคิวงานที่ระบบย่อย

ถ้าต้องการกำหนด job queue entry ที่ subsystem description ให้ใช้ อินเทอร์เน็ตแบบอักษร

คำสั่ง: Add Job Queue Entry (ADDJOBQE)

พารามิเตอร์ บนคำสั่งนี้ระบุ:

- จำนวนของงานที่สามารถแอดคิพีในเวลาเดียวกันบนคิวงานนี้ (MAXACT)
- การจัดลำดับที่ระบบย่อยจะจัดการกับงานจากคิวงานนี้ (SEQNBR)
- จำนวนงานที่สามารถแอดคิพีได้พร้อมกันในแต่ละเก้าระดับความสำคัญเป็นจำนวนเท่าใด (MAXPTYn) (n=1 through 9)

ตัวอย่าง: ตัวอย่างต่อไปนี้จะใส่เพิ่ม job queue entry สำหรับ คิวงาน JOBQA ใน TEST subsystem description ไม่มีจำนวนสูงสุดของงานที่สามารถ แอดคิพีบนคิวงานนี้ และงานจะถูกประมวลผลโดยมี หมายเลขลำดับเป็นห้า

```
ADDJOBQE SBS(D(TE)ST) JOBQ(LIBA/JOBQA) MAXACT(*NOMAX) SEQNBR(5)
```

หลักการที่เกี่ยวข้อง

“วิธีการทำงานของคิวงาน” ในหน้า 69

คิวงานถูกจัดสรรโดยระบบย่อยผ่านทาง job queue entry. สามารถวางงานบนคิวงานได้ แม้ว่ายังไม่ได้เริ่มระบบย่อย. เมื่อระบบย่อยเริ่มขึ้น, ระบบย่อยจะประมวลผลงานบนคิว.

งานที่เกี่ยวข้อง

“การลบ job queue entry” ในหน้า 191

คุณสามารถลบ job queue entries ออกจาก subsystem description โดยใช้อินเทอร์เน็ตแบบอักษร งานบนคิวงานยังคงอยู่ บนคิว เมื่อ job queue entry ถูกลบออกจาก subsystem description จะไม่สามารถลบ job queue entry ถ้างานแอดคิพีในปัจจุบันใดๆ ถูกเริ่มจาก คิวงาน

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

งาน 9

แต่ละ job queue entry ในสถานการณ์จำลองนี้ถูกระบุเป็น MAXACT(*NOMAX) ในอันดับแรก ระบบย่อยจะเลือกงานจากคิวงาน A เนื่องจาก job queue entry มีหมายเลขลำดับต่ำสุด ถ้าจำนวนสูงสุดของงานในระบบย่อย คือ 3 (พารามิเตอร์ MAXJOBS(3) บนคำสั่ง Create Subsystem Description (CRTSBSD)) ระบบย่อยสามารถเลือกงานทั้งหมดจากคิวงาน A ให้แอดคิพีได้พร้อมกัน

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

หลักการที่เกี่ยวข้อง

“วิธีการทำงานของคิวงาน” ในหน้า 69

คิวงานถูกจัดสรรโดยระบบย่อยผ่านทาง job queue entry. สามารถวางงานบนคิวงานได้ แม้ว่าจะยังไม่ได้เริ่มระบบย่อย. เมื่อระบบย่อยเริ่มขึ้น, ระบบย่อยจะประมวลผลงานบนคิว.

การเปลี่ยนจำนวนของงานที่รันได้พร้อมกันในคิวงาน

ระบบย่อย QBASE ถูกจัดส่งมาพร้อมกับ job queue entry สำหรับคิวงาน QBATCH entry นี้อนุญาตให้มีงานแบ็คซ์ที่จะรันได้เพียงหนึ่งงานเท่านั้นในแต่ละครั้ง ถ้าคุณต้องการให้สามารถรันงานแบ็คซ์ได้พร้อมกันหลายงานจากคิวงานนั้น คุณต้องเปลี่ยน job queue entry

ถ้าต้องการเปลี่ยนจำนวนของงานที่รันได้พร้อมกันจากคิวงาน ให้ใช้อินเตอร์เฟสแบบอักษร

คำสั่ง: Change Job Queue Entry (CHGJOBQE)

ตัวอย่าง: คำสั่งต่อไปนี้อุญาตให้รันงานแบ็คซ์ได้พร้อมกันสองงานจาก คิวงาน QBATCH ในระบบย่อย QBASE (สามารถใช้คำสั่งนี้ได้ตลอดเวลาและคำสั่งนี้มีผลบังคับใช้ทันที)

```
CHGJOBQE SBSDB(QBASE) JOBQ(QBATCH) MAXACT(2)
```


หลักการที่เกี่ยวข้อง

“วิธีการนำงานมาจากหลายคิวงาน” ในหน้า 73

ระบบย่อยประมวลผลงานจากคิวงานตามหมายเลขลำดับ เนื่องจากระบบย่อยสามารถมีมากกว่าหนึ่ง job queue entry ดังนั้นจึงสามารถจัดสรร มากกว่าหนึ่งคิวงานได้

“วิธีการนำงานมาจากคิวงาน” ในหน้า 70

มีปัจจัยหลายอย่างที่กำหนดวิธีการเลือกงานจากคิวงาน และเริ่มต้น.

“Job queue entry” ในหน้า 71

job queue entry ระบุคิวงานซึ่งงานจะถูกเลือกสำหรับการรันในระบบย่อย พารามิเตอร์ใน job queue entry ที่ควบคุมวิธีการจัดการคิวงานมีอยู่ห้าพารามิเตอร์

การลบคิวงาน

เมื่อคุณล้างข้อมูลคิวงาน งานทุกงานบนคิวจะถูกลบออก รวมถึงงานใดๆ ที่อยู่ในสถานะถูกพักด้วย คุณสามารถใช้ System i Navigator หรืออินเทอร์เน็ตเฟสแบบอักษรเพื่อลบคิวงาน งานที่กำลังรันอยู่จะไม่ได้รับผลกระทบ เนื่องจากถือว่างานเหล่านั้นเป็นงานแอคทีฟและไม่ได้อยู่บนคิว อีกต่อไป

หลักการที่เกี่ยวข้อง

“คิวงาน” ในหน้า 68

คิวงานประกอบด้วยรายการลำดับของงาน ที่รอการประมวลผลโดยระบบย่อย คิวงานจะเป็นที่แรก ที่งานแบบแบตช์ที่ส่งไปพักรอ ก่อนที่จะไปทำงาน อยู่ในระบบย่อย งานจะถูกพักอยู่ที่นี้จนกว่าจะมีสภาพตรงตามเงื่อนไขการทำงานต่างๆ

System i Navigator:

ในการใช้ System i Navigator เพื่อลบคิวงาน ให้ปฏิบัติตามขั้นตอนเหล่านี้:

1. ขยาย My Connections → connection → Job Queues → Active Job Queues หรือ All Job Queues.
2. คลิกขวาที่คิวงานและคลิก Clear. หน้าต่าง Confirm Clear จะปรากฏขึ้น ในหน้าต่างนี้คุณสามารถระบุได้ว่า ต้องการให้มีการจัดทำบันทึกการใช้งานเมื่อล้างคิวหรือไม่

อินเทอร์เน็ตเฟสแบบอักษร:

คำสั่ง: Clear Job Queue (CLRJOBQ)

ตัวอย่าง: คำสั่งนี้จะลบงานทั้งหมดที่มีอยู่ในปัจจุบันในคิวงานที่ได้มาจาก IBM QBATCH งานใดๆ ที่ถูกอ่านอยู่ในปัจจุบันจะไม่ได้รับผลกระทบ

```
CLRJOBQ JOBQ(QGPL/QBATCH)
```

การสร้างคิวงาน

ถ้าต้องการสร้างคิวงาน ให้ใช้อินเทอร์เน็ตเฟสแบบอักษร

คำสั่ง: Create Job Queue (CRTJOBQ)

ตัวอย่าง: ตัวอย่างต่อไปนี้จะสร้างคิวงานที่มีชื่อว่า JOBQA ในไลบรารีLIBA:

```
CRTJOBQ JOBQ(LIBA/JOBQA) TEXT('test job queue')
```

หลังจากที่คุณ สร้างคิวงานแล้ว, ต้องกำหนดคิวงานที่ระบบย่อยก่อน จึงจะสามารถรันงานใดๆ ได้ เพื่อกำหนดคิวงานที่ระบบย่อย ให้ใส่เพิ่ม job queue entry ที่ subsystem description

หลักการที่เกี่ยวข้อง

“คิวงาน” ในหน้า 68

คิวงานประกอบด้วยรายการลำดับของงาน ที่รอการประมวลผลโดยระบบย่อย คิวงานจะเป็นที่แรก ที่งานแบบแบตช์ที่ส่งไปพักรอ ก่อนที่จะไปทำงาน อยู่ในระบบย่อยงานจะถูกพักอยู่ที่นี้จนกว่าจะมีสภาพตรงตามเงื่อนไขการทำงานต่างๆ

“วิธีการทำงานของคิวงาน” ในหน้า 69

คิวงานถูกจัดสรรโดยระบบย่อยผ่านทาง job queue entry. สามารถวางงานบนคิวงานได้ แม้ว่ายังไม่ได้เริ่มระบบย่อย. เมื่อระบบย่อยเริ่มขึ้น, ระบบย่อยจะประมวลผลงานบนคิว.

การลบคิวงาน

ถ้าต้องการลบคิวงาน ให้ใช้อินเตอร์เฟซแบบอักษร

ข้อจำกัด:

- คิวงานที่กำลังจะลบไม่สามารถมี entries ใดๆ งานทั้งหมดบนคิว ต้องเสร็จสมบูรณ์, ลบออก หรือย้ายไปที่คิวงานอื่น
- ระบบย่อยไม่สามารถแอนด์ที่ที่คิวงาน

วิธีการลบคิวงานมีมากกว่าหนึ่งวิธี แม้ว่าในที่นี้จะแสดงวิธีการลบสองวิธี แต่แนะนำให้ใช้วิธีการใช้คำสั่ง WRKJOBQ เนื่องจากวิธีนี้จะแสดงจำนวน และสถานะของงาน

คำสั่ง: Work with Job Queue (WRKJOBQ)

ถ้าจำนวนของงานเป็น 0 คุณสามารถใช้อ็อปชัน 4=ลบออก เพื่อลบคิวงานออกจากไลบรารีได้

ใช้ DLTJOBQ พร้อมกับ automated scripts และสภาวะแวดล้อมการล้างข้อมูล ระวังเมื่อใช้วิธีการนี้เนื่องจากการดำเนินการตีฟอลต์ของคำสั่งนี้คือ การค้นหารายชื่อไลบรารี และลบคิวงานแรกตรงที่ตรงกับชื่อที่ระบุ ถ้าคุณมีคิวงานสองคิวที่มีชื่อเหมือนกันในไลบรารีต่างกัน คุณอาจลบคิวงานที่ผิดออก คุณสามารถแก้ไขการดำเนินการนี้โดยการ ระบุไลบรารีเฉพาะ

คำสั่ง: Delete Job Queue (DLTJOBQ)

ตัวอย่าง: คำสั่งนี้จะลบคิวงาน SPECIALJQ ในไลบรารี SPECIALLIB

```
DLTJOBQ JOBQ(SPECIALLIB/SPECIALJQ)
```

หลักการที่เกี่ยวข้อง

“คิวงาน” ในหน้า 68

คิวงานประกอบด้วยรายการลำดับของงาน ที่รอการประมวลผลโดยระบบย่อย คิวงานจะเป็นที่แรก ที่งานแบบแบตช์ที่ส่งไปพักรอ ก่อนที่จะไปทำงาน อยู่ในระบบย่อยงานจะถูกพักอยู่ที่นี้จนกว่าจะมีสภาพตรงตามเงื่อนไขการทำงานต่างๆ

การกำหนดว่าระบบย่อยใดมีคิวงานที่ถูกจัดสรร

คุณสามารถกำหนดว่าระบบย่อยใดมีคิวงานที่ถูกจัดสรรได้โดยใช้ System i Navigator อินเทอร์เน็ต หรืออินเทอร์เน็ตแบบอักษร สิ่งนี้มีประโยชน์เมื่อคุณพบว่าจำเป็นต้อง ลบคิวงานออก เนื่องจากคุณไม่สามารถลบคิวงานซึ่งมีระบบย่อยแอนด์ที่ฟ

หลักการที่เกี่ยวข้อง

“วิธีการทำงานของคิวงาน” ในหน้า 69

คิวงานถูกจัดสรรโดยระบบย่อยผ่านทาง job queue entry. สามารถวางงานบนคิวงานได้ แม้ว่ายังไม่ได้เริ่มระบบย่อย. เมื่อระบบย่อยเริ่มขึ้น, ระบบย่อยจะประมวลผลงานบนคิว.

System i Navigator:

ในการดูรายละเอียดมีคิวงานที่ถูกจัดสรรให้ปฏิบัติตามข้อแนะนำต่อไปนี้:

1. จาก System i Navigator ให้ขยาย **Work Management** → **Job Queues** → **All Job Queues**.
2. ระบุตำแหน่งคิวงานในระนาบด้านขวาของ System i Navigator อินเทอร์เน็ต ระบบย่อยที่มีคิวงานที่ถูกจัดสรรจะแสดงในคอลัมน์ Subsystem

หากคุณไม่เห็นคอลัมน์ Subsystem ให้ใส่เพิ่มคอลัมน์ที่จอแสดงผล คลิกขวาที่ **All Job Queues** → **Customize this view** → **Columns**)

3. หรือคุณสามารถคลิกขวาที่คิวงาน และคลิก **Properties**. ระบบย่อยจะถูกแสดงรายการบนหน้า General ของหน้าต่าง Job Queue Properties

อินเทอร์เน็ตเฟสแบบอักษร:

คำสั่ง: WRKJOBQ JOBQ(LIBA/JOBQA) โดยที่ JOBQA เป็นชื่อคิวงาน

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 ไม่มีผลต่องานที่กำลังรันอยู่สามารถวางแผนเพิ่มเติม บนคิวงานในขณะที่คิวงานถูกพักได้ แต่งานจะไม่ถูกประมวลผล

เพื่อพักคิวงาน คุณสามารถใช้ System i Navigator หรือใช้อินเทอร์เน็ตแบบอักษร

หลักการที่เกี่ยวข้อง

“คิวงาน” ในหน้า 68

คิวงานประกอบด้วยรายการลำดับของงาน ที่รอการประมวลผลโดยระบบย่อย คิวงานจะเป็นที่แรก ที่งานแบบแบตช์ที่ส่งไปพักรอ ก่อนที่จะไปทำงาน อยู่ในระบบย่อย งานจะถูกพักอยู่ที่นั่นจนกว่าจะมีสภาพตรงตามเงื่อนไขการทำงานต่างๆ

System i Navigator:

จาก System i Navigator ให้ขยาย **My Connections** → **connection** → **Work Management** → **Job Queues** → **Active Job Queues** → **คลิกขวาที่คิว** → **Hold**.

อินเทอร์เน็ตเฟสแบบอักษร:

คำสั่ง: Hold Job Queue (HLDJOBQ)

ในตัวอย่างนี้ คิวงาน QBATCH จะถูกพัก งานทั้งหมดที่ไม่ได้รันอยู่ในขณะที่ออกคำสั่งนี้จะถูกพักไว้จนกว่า คิวจะถูกรีลีสหรือลบ

HLDJOBQ JOBQ(QBATCH)

การรีลีสคิวงาน

เมื่อคุณรีลีสคิวงาน งานทั้งหมดที่ถูกพักไว้อันเป็นผลมาจาก การพักคิวงานจะถูกรีลีสด้วย ถ้างานแต่ละงานถูกพักก่อนที่คิวงานจะถูกพัก งานจะไม่ถูก รีลีส

ในการรีลีสคิวงาน ให้ใช้ System i Navigator หรืออินเทอร์เฟซแบบอักษร

หลักการที่เกี่ยวข้อง

“คิวงาน” ในหน้า 68

คิวงานประกอบด้วยรายการลำดับของงาน ที่รอการประมวลผลโดยระบบย่อย คิวงานจะเป็นที่แรก ที่งานแบบแบตช์ที่ส่งไปพักรอ ก่อนที่จะไปทำงาน อยู่ในระบบย่อยงานจะถูกพักอยู่ที่นี้จนกว่าจะมีสภาพตรงตามเงื่อนไขการทำงานต่างๆ

System i Navigator:

จาก System i Navigator ให้ขยาย My Connections → Connection → Work Management → Job Queues → All Job Queues → Right-click the queue → Release.

อินเทอร์เฟซแบบอักษร:

คำสั่ง: Release Job Queue (RLSJOBQ)

ตัวอย่างนี้จะรีลีสคิวงาน QBATCH

RLSJOBQ JOBQ(QBATCH)

การย้ายงานไปที่คิวงานอื่น

มีหลายเหตุผลที่คุณอาจต้องการย้ายงานไปที่คิวอื่น ตัวอย่างเช่น ในบางครั้งงานอาจถูก backlogged ในคิวเนื่องจากเป็นงานที่ใช้เวลานาน บางครั้งรันใหม่ที่จัดตารางเวลาของงานขัดแย้งกับ งานใหม่ที่มีระดับความสำคัญสูงกว่าวิธีหนึ่งในการจัดการสถานการณ์นี้คือ การย้ายงานที่รอไปยังคิวอื่นที่ไม่ยุ่ง

คุณสามารถใช้อินเทอร์เฟซ System i Navigator หรืออินเทอร์เฟซแบบอักษรเพื่อย้ายงานจากคิวงานหนึ่งไปยังคิวอื่น

หลักการที่เกี่ยวข้อง

“วิธีการนำงานมาจากหลายคิวงาน” ในหน้า 73

ระบบย่อยประมวลผลงานจากคิวงานตามหมายเลขลำดับ เนื่องจากระบบย่อยสามารถมีมากกว่าหนึ่ง job queue entry ดังนั้นจึงสามารถจัดสรร มากกว่าหนึ่งคิวงานได้

“วิธีการนำงานมาจากคิวงาน” ในหน้า 70

มีปัจจัยหลายอย่างที่กำหนดวิธีการเลือกงานจากคิวงาน และเริ่มต้น.

System i Navigator:

ในการใช้ System i Navigator ให้ปฏิบัติตามข้อแนะนำต่อไปนี้:

1. จาก System i Navigator ให้ขยาย Work Management → Job Queues → All Job Queues.
2. ระบุตำแหน่งและเปิดคิวที่มีงานอยู่ในปัจจุบัน
3. คลิกขวาที่คุณต้องการย้าย 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)

การวางแผนบนคิวงาน

งานถูกวางแผนบนคิวงานโดยการย้ายงานที่มีอยู่จากคิวงานหนึ่งไปยังคิวงานอื่น หรือโดยการส่งงานใหม่ อย่างไม่อย่างหนึ่ง ใช้ System i Navigator เพื่อย้ายงานระหว่างคิวต่างๆ ใช้อินเตอร์เฟซแบบอักษรเพื่อส่งงานใหม่

หลักการที่เกี่ยวข้อง

“วิธีการนำงานมาจากหลายคิวงาน” ในหน้า 73

ระบบย่อยประมวลผลงานจากคิวงานตามหมายเลขลำดับ เนื่องจากระบบย่อยสามารถมีมากกว่าหนึ่ง job queue entry ดังนั้นจึงสามารถจัดสรร มากกว่าหนึ่งคิวงานได้

“วิธีการนำงานมาจากคิวงาน” ในหน้า 70

มีปัจจัยหลายอย่างที่กำหนดวิธีการเลือกงานจากคิวงาน และเริ่มต้น.

System i Navigator:

ในการใช้ System i Navigator อินเตอร์เฟซงานต้องมีอยู่แล้วในคิวงานอื่น จากนั้นคุณสามารถย้ายงานจากคิวหนึ่งไปยังคิวงานอื่นได้ (ถ้าต้องการวางแผนใหม่บนคิวงาน ให้ใช้อินเตอร์เฟซบรรทัดรับคำสั่ง)

1. จาก System i Navigator ให้ขยาย Work Management → Job Queues → All Job Queues.
2. คลิกขวางานที่คุณต้องการย้าย หน้าต่าง Move ที่คุณสามารถระบุคิวปลายทางจะเปิดขึ้น

อินเตอร์เฟซแบบอักษร:

ข้อมูลต่อไปนี้เป็นรายการของวิธีการใช้อินเตอร์เฟซแบบอักษร สำหรับการวางแผนใหม่บนคิวงานใหม่

- Submit Job (SBMJOB): อนุญาตให้งานที่กำลังรันอยู่ส่งงานอื่นไปที่คิวงานเพื่อรันในภายหลังเป็นงานแบ็คซ์ สามารถวางแผนได้เพียงหนึ่งองค์ประกอบของข้อมูลคำขอเท่านั้นบน message queue ของงานใหม่ ข้อมูลคำขอสามารถเป็นคำสั่ง CL ถ้า routing entry ที่ใช้สำหรับงานระบุ โปรแกรมการประมวลผลคำสั่ง CL (เช่น โปรแกรม QCMD ที่ได้มาจาก IBM)
- Add Job Schedule Entry (ADDJOBSCDE): ระบบส่งงานไปที่คิวงานโดยอัตโนมัติในเวลา และวันที่ที่ระบุไว้ใน job schedule entry
- Submit Database Jobs (SBMDBJOB): ส่งงานไปที่คิวงานเพื่อให้งานสามารถรันเป็นงานแบ็คซ์ได้ input stream ถูกอ่านจาก ไฟล์ฐานข้อมูลฟิสิกัลหรือจากไฟล์ฐานข้อมูลโลจิคัล อย่างไม่อย่างหนึ่ง ที่มีรูปแบบเร็คคอร์ดเดียว คำสั่งนี้อนุญาตให้คุณระบุชื่อไฟล์ของฐานข้อมูล และเมมเบอร์ของไฟล์, ชื่อของคิวงานที่จะใช้ และเลือกตัดสินใจว่างานที่จะส่งสามารถถูกแสดงโดยคำสั่ง Work with Submitted Jobs (WRKSBMJOB) หรือไม่
- Start Database Reader (STRDBRDR): อ่าน batch input stream จากฐานข้อมูล และวางแผนหนึ่ง หรือหลายงานบนคิวงาน
- Transfer Job (TFRJOB): ย้ายงานปัจจุบันไปยังคิวงานอื่นในระบบย่อยแอ็คทีฟ
- Transfer Batch Job (TFRBCHJOB): ย้ายงานปัจจุบันไปยังคิวงานอื่น

การค้นหาคิวงานทั้งหมดสำหรับงานเฉพาะ

คุณสามารถใช้ System i Navigator หรืออินเตอร์เฟซแบบอักษรอย่างใดอย่างหนึ่ง เพื่อค้นหาคิวงานสำหรับงานเฉพาะ

System i Navigator:

1. จาก System i Navigator ให้ขยาย My Connections → Connection → Basic Operations → Right-click Jobs → Customize this View → Include.
2. ใช้หน้าต่าง Jobs-Include เพื่อลดจำนวนของงานที่จะแสดง ตรวจสอบให้แน่ใจว่าฟิลด์ Job queue ถูกเซต เป็น All
3. เมื่อคุณคลิก OK งานทั้งหมดที่ตรงกับเกณฑ์ของคุณจะแสดงขึ้น

อินเตอร์เฟซแบบอักษร:

คำสั่ง: Work with Job Queues (WRKJOBQ)

ตัวอย่าง: ตัวอย่างต่อไปนี้จะสร้างรายการของงานทั้งหมดบนคิวงาน JOBQA.

WRKJOBQ JOBQ(LIBA/JOBQA)

การค้นหงานถ้าคุณไม่ทราบชื่อของคิวงาน:

ถ้าคุณไม่ทราบชื่อของคิวงาน ให้ปฏิบัติตามข้อแนะนำดังต่อไปนี้:

1. ป้อนคำสั่งโดยไม่ป้อนพารามิเตอร์ JOBQ หน้าต่าง Work with All Job Queues จะปรากฏขึ้นพร้อมกับรายการของคิวงานทั้งหมด ซึ่งคุณได้รับสิทธิให้ใช้งานได้
2. สแกนรายการนี้จนกว่าคุณจะเห็นชื่อของคิวงานที่อาจมีงาน ซึ่งคุณกำลังพยายามหาอยู่

หลังจากที่คุณพบงานในคิวงานแล้ว คุณสามารถดูที่งานนั้นได้โดยการป้อนงานพร้อมกับอ็อปชันสำหรับงานที่คุณต้องการดู จอแสดงผล Work with Job จะปรากฏขึ้น จอแสดงผลนี้แสดงอ็อปชันต่างๆ สำหรับการดู ข้อมูลทั้งหมดที่พร้อมใช้งานสำหรับงานที่คุณได้เลือก

ถ้าคุณทราบงานที่คุณจะค้นหา คำสั่งต่อไปนี้จะนำคุณไปยังจอแสดงผลงานโดยตรง

WRKJOB JOB(หมายเลข/ผู้ใช้/ชื่อ) OPTION(*DFNA)

ถ้าคุณไม่แน่ใจเกี่ยวกับงานที่จะค้นหา Work with Submitted Jobs (WRKSBMJOB) หรือ Work with User Jobs (WRKUSRJOB) สามารถช่วยได้

การระบุระดับความสำคัญสำหรับคิวงาน

ถ้าต้องการระบุลำดับการประมวลผลคิวงานโดยระบบย่อย ให้ใช้อินเตอร์เฟซแบบอักษร

คำสั่ง: Add Job Queue Entry (ADDJOBQE)

พารามิเตอร์ บนคำสั่งนี้ระบุ:

- จำนวนของงานที่สามารถแอดคิพีในเวลาเดียวกันบนคิวงานนี้ (MAXACT)
- การจัดลำดับที่ระบบย่อยจะจัดการกับงานจากคิวงานนี้ (SEQNBR)
- จำนวนงานที่สามารถแอดคิพีได้พร้อมกันในแต่ละเกียรดับความสำคัญเป็นจำนวนเท่าใด (MAXPTYn) (n=1 through 9)

การจัดการเอาต์พุตคิว

เอาต์พุตคิวจะช่วยคุณจัดการพรินเตอร์เอาต์พุตที่สร้างขึ้นเมื่อสิ้นสุดการทำงาน เป็นสิ่งสำคัญที่ต้องทำความเข้าใจวิธีการรักษาเอาต์พุตคิวของคุณอย่างมีประสิทธิภาพ เพื่อให้เอาต์พุตที่พิมพ์ของคุณดำเนินไปอย่างรวดเร็ว

พรินเตอร์เอาต์พุตอยู่บนเอาต์พุตคิว เอาต์พุตคิวกำหนดลำดับที่พรินเตอร์เอาต์พุตจะถูกประมวลผลโดยอุปกรณ์การพิมพ์ โดยการจัดการเอาต์พุตคิวของคุณ คุณสามารถมั่นใจถึงการประมวลผลที่ราบรื่นของ พรินเตอร์เอาต์พุตของคุณ

หลักการที่เกี่ยวข้อง

“เอาต์พุตคิว” ในหน้า 74

เอาต์พุตคิวคือพื้นที่ที่ printer output files (หรือเรียกว่าสพูลไฟล์) รอการถูกประมวลผลและถูกส่งไปยังพรินเตอร์ พรินเตอร์เอาต์พุตจะถูกสร้างขึ้นทั้งโดยระบบหรือโดยผู้ใช้ที่ใช้ไฟล์พิมพ์ อย่างใดอย่างหนึ่ง

การสร้างเอาต์พุตคิว

คำสั่ง Create Output Queue (CRTOUTQ) จะสร้างเอาต์พุตคิวใหม่สำหรับสพูลไฟล์ entry จะถูกวางบนเอาต์พุตคิว สำหรับแต่ละสพูลไฟล์ ลำดับในการบันทึกไฟล์ไปที่อุปกรณ์เอาต์พุตถูกกำหนดโดยระดับความสำคัญเอาต์พุตของสพูลไฟล์ และค่าที่ระบุบน Order of files on queue prompt (พารามิเตอร์ SEQ) ใช้อินเตอร์เฟซแบบอักษรในการสร้างเอาต์พุตคิว

คำสั่ง: CRTOUTQ (Create Output Queue)

ตัวอย่าง: คำสั่งนี้จะสร้างเอาต์พุตคิวที่มีชื่อว่า DEPTAPRT และวางเอาต์พุตคิวในไลบรารีปัจจุบัน เนื่องจากมีการระบุ AUT (*EXCLUDE) และสมมุติค่า OPRCTL(*YES) เอาต์พุตคิวจึงสามารถถูกใช้งานและควบคุมโดยผู้ใช้ที่เป็นคนสร้างคิวและผู้ใช้ที่มีสิทธิควบคุมงาน หรือสิทธิควบคุมสพูลเท่านั้น เนื่องจากมีการระบุ SEQ(*FIFO) สพูลไฟล์ จึงถูกวางในลำดับ first-in first-out (เข้าก่อนออกก่อน) บนคิว ถ้าผู้ใช้ใน Department A ได้รับอนุญาตให้ใช้เอาต์พุตคิวนี้ ต้องใช้คำสั่ง Grant Object Authority (GRTOBJAUT) เพื่อมอบสิทธิในการใช้งานที่จำเป็นให้แก่ผู้ใช้ ข้อมูลที่มีอยู่ในไฟล์ต่างๆบนคิวนี้สามารถถูกแสดงโดยผู้ใช้ที่เป็นเจ้าของไฟล์, โดยเจ้าของคิว, โดยผู้ใช้ที่มีสิทธิควบคุมงาน หรือโดยผู้ใช้ที่มีสิทธิควบคุมสพูลเท่านั้น 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')
```

การกำหนดเอาต์พุตคิวที่งาน หรือรายละเอียดของงาน

ก่อนที่คุณจะสามารถใช้เอาต์พุตคิวที่สร้างขึ้นใหม่ได้ คุณต้องกำหนด เอาต์พุตคิวที่งานหรือที่รายละเอียดของงาน คุณสามารถกำหนดเอาต์พุตคิวโดยการใช้ System i Navigator หรืออินเตอร์เฟซแบบอักษร

System i Navigator:

ในการใช้ System i Navigator เพื่อกำหนดเอาต์พุตคิวที่งาน ให้ปฏิบัติตามขั้นตอนดังต่อไปนี้:

1. จาก System i Navigator, ให้ขยาย **Work Management** → **Active Jobs**.
2. คลิกขวาที่งาน และคลิก **Properties** → **Printer Output**.

อินเตอร์เฟซแบบอักษร:

คุณยังสามารถเปลี่ยนรายละเอียดของงานเพื่อใช้เอาต์พุตคิวใหม่ได้ด้วย ดังนั้น งานทั้งหมดที่ใช้รายละเอียดของงานจะใช้เอาต์พุตคิวใหม่ ใช้อินเตอร์เฟซแบบอักษรเพื่อกำหนดเอาต์พุตคิวที่ รายละเอียดของงาน

คำสั่ง: Change Job Description (CHGJOB)

ตัวอย่างต่อไปนี้จะเปลี่ยนรายละเอียดของงาน AMJOBS เพื่อใช้เอาต์พุตคิว QPRINT

```
CHGJOB JOB(AMJOBS/AMJOBS) OUTQ(*LIBL/QPRINT)
```

การเข้าใช้งานพรินเตอร์เอาต์พุต

เนื่องจากคุณมีตัวเลือกในการดึงพรินเตอร์เอาต์พุตออกมาจากงานเมื่องานรันเสร็จสิ้นแล้ว (การแยกพรินเตอร์เอาต์พุตออกจากงานอย่างสมบูรณ์) คุณสามารถเข้าใช้งานพรินเตอร์เอาต์พุตของคุณได้ใน System i Navigator ผ่านทาง Basic Operations หรือทางการจัดการระบบงาน

System i Navigator:

ถ้าต้องการเข้าใช้งานพรินเตอร์เอาต์พุตของงานผ่าน Basic Operations, ให้ทำดังต่อไปนี้:

1. จาก System i Navigator, ให้ขยาย **My Connections** → **connection** → **Basic Operations** → **Job**.
2. คลิกขวางานที่คุณต้องการแสดงพรินเตอร์เอาต์พุต และคลิก **Printer Output**. หน้าต่าง Printer Output จะ ปรากฏขึ้น

ถ้าต้องการเข้าใช้งานพรินเตอร์เอาต์พุตผ่านทางโพลเดอร์ Output Queues ให้ทำ ดังต่อไปนี้:

1. จาก System i Navigator, ให้ขยาย **My Connections** → **connection** → **Work Management** → **Output Queues**.
2. เลือกเอาต์พุตคิวที่คุณต้องการแสดงผลพรินเตอร์เอาต์พุตด้วย (เช่น Qprint2) พรินเตอร์เอาต์พุตภายในเอาต์พุตคิวจะ ปรากฏขึ้น

อินเตอร์เฟซแบบอักษร:

คำสั่ง: Work with Output Queue (WRKOUTQ <output queue name>)

คำสั่ง: Work with Spooled Files (WRKSPLF JOB(ชื่องานที่ถูกต้อง)

การลบเอาต์พุตคิว

เมื่องานสร้างพรินเตอร์เอาต์พุต เอาต์พุตจะถูกส่งไปยังเอาต์พุตคิวเพื่อพิมพ์ ส่วนใหญ่แล้ว คุณจะพิมพ์พรินเตอร์เอาต์พุตทั้งหมดที่สร้างขึ้น System i Navigator ให้คุณสามารถล้างเอาต์พุตของคุณ โดยใช้ Clear อีอ็อปชัน การล้างเอาต์พุตคิวสามารถลบเอาต์พุตทั้งหมดออกจากคิวได้

System i Navigator:

ถ้าต้องการล้างเอาต์พุตคิว ให้ทำตามขั้นตอนเหล่านี้:

1. จาก System i Navigator ให้ขยาย **My Connections** → **Connection** → **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

คุณสามารถใช้อินเตอร์เฟซแบบอักษรเพื่อลบเอาต์พุตคิวได้

ก่อนที่จะสามารถลบเอาต์พุตคิวได้ เอาต์พุตคิวต้องเป็นไปตามข้อกำหนด ดังต่อไปนี้

เอาต์พุตคิวที่กำลังจะลบไม่สามารถมี entries ใดๆ เอาต์พุตสำหรับแต่ละไฟล์ต้องถูก พิมพ์, ลบออก, หรือย้ายไปที่เอาต์พุตคิวอื่น ระบบย่อยไม่สามารถแก้ไขคิวคิวที่ไม่สามารถถูกใช้โดย spooling writer จะไม่สามารถลบคิวออก ถ้าคิวถูกสร้างโดยระบบสำหรับพริเตอร์เฉพาะ

คำสั่ง: Delete Output Queue (DLTOUTQ)

คำสั่งนี้จะลบเอาต์พุตคิว PUNCH2 ออกจากระบบ

```
DLTOUTQ  OUTQ(PUNCH2)
```

การดูเอาต์พุตคิวในระบบ

เอาต์พุตคิวกำหนดลำดับที่พริเตอร์เอาต์พุตจะถูกส่งไปยังอุปกรณ์การพิมพ์ คุณสามารถดูเอาต์พุตคิวโดยใช้ System i Navigator.

ในการดูเอาต์พุตคิวในระบบ ให้ปฏิบัติตามข้อแนะนำต่อไปนี้:

1. จาก System i Navigator ให้ขยาย My Connections → connection → Work Management.
2. คลิก Output Queues.

จาก System i Navigator คุณสามารถปรับรายการของเอาต์พุตคิวที่คุณกำลังดู โดยใช้หน้าต่าง Include หน้าต่าง Include อนุญาตให้คุณใส่ข้อจำกัดของสิ่งที่แสดงใน System i Navigator. ตัวอย่างเช่น คุณสามารถรัน Include เพื่อแสดงเพียงบางเอาต์พุตคิวได้

ถ้าต้องการใช้ ฟังก์ชัน include ให้คลิกเมนู View แล้ว Customize this View.

การจัดการบันทึกการใช้งาน

งานส่วนใหญ่บนระบบของคุณมีบันทึกการใช้งานที่เชื่อมโยง บันทึกการใช้งานบอกให้คุณ ทราบถึงสิ่งต่างๆ มากมาย เช่น งาน จะเริ่มเมื่อไร, งานจะจบเมื่อไร, คำสั่งอะไรกำลังรันอยู่, เหตุผลของความล้มเหลวและข้อความแสดงความผิดพลาด ข้อมูลนี้ช่วยให้คุณเห็นภาพว่าวัฏจักรการรันงานเป็นอย่างไร

ข้อมูลต่อไปนี้อธิบายภารกิจหลายอย่างที่คุณสามารถปฏิบัติ เมื่อทำงานกับบันทึกการใช้งาน

หลักการที่เกี่ยวข้อง

“บันทึกการใช้งาน” ในหน้า 84

บันทึกการใช้งานมีข้อมูลที่เกี่ยวข้องกับคำขอที่ป้อนสำหรับ งาน บันทึกการใช้งานมีสองรูปแบบ รูปแบบการคงค้างและรูปแบบสพูล

การจัดการเซิร์ฟเวอร์บันทึกการใช้งาน

ระบบย่อย QSYSWRK ควบคุมเซิร์ฟเวอร์บันทึกการใช้งาน อย่างไรก็ตาม มีบางภารกิจ ที่คุณสามารถปฏิบัติเพื่อปรับหรือจัดการเซิร์ฟเวอร์บันทึกการใช้งาน

หลักการที่เกี่ยวข้อง

“บันทึกการใช้งาน” ในหน้า 84

บันทึกการใช้งานมีข้อมูลที่เกี่ยวข้องกับคำขอที่ป้อนสำหรับ งาน บันทึกการใช้งานมีสองรูปแบบ รูปแบบการคงค้างและรูปแบบสพูล

Reconfiguring เซิร์ฟเวอร์บันทึกการใช้งาน:

เมื่อจัดส่งมา เซิร์ฟเวอร์บันทึกการใช้งานจะรันใน QSYSWRK QSYSWRK จะแก้ไขอย่างต่อเนือง เพื่อเพิ่มประสิทธิภาพ คุณอาจต้อง reconfigure เซิร์ฟเวอร์บันทึกการใช้งานของคุณเพื่อรันในระบบย่อยอื่น

ถ้าต้องการ reconfigure เซิร์ฟเวอร์บันทึกการใช้งานเพื่อรันในระบบย่อยอื่น ให้ใช้อินเตอร์เฟซแบบอักขระและทำตามขั้นตอนเหล่านี้

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
2. เปลี่ยนคิวงานที่ระบุในรายละเอียดของงาน QJOBLOGSVR เป็นคิวงานที่มีอยู่ในระบบย่อยของคุณ
3. ใส่เพิ่ม QJOBLOGAJ autostart job entry (พร้อมกับ routing entry, ถ้าจำเป็น) ลงในระบบย่อยของคุณ การเพิ่มนี้จะทำให้เซิร์ฟเวอร์บันทึกการใช้งานเริ่มขึ้นโดยอัตโนมัติเมื่อระบบย่อยเริ่มขึ้น
 - 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)
```

หลักการที่เกี่ยวข้อง

“เซิร์ฟเวอร์บันทึกการใช้งาน” ในหน้า 87

โดยปกติ เซิร์ฟเวอร์บันทึกการใช้งานจะพิมพ์บันทึกการใช้งานของงานไปที่ สพูลไฟล์ คุณสามารถเรดต์บันทึกการใช้งานไปที่ปริ้นเตอร์หรือไปที่ outfile (ถ้ามี การระบุให้ทำโดยใช้คำสั่ง QMHCTLJL, Control job log API) อย่างไรก็ตามแนะนำให้ใช้วิธีการนี้สำหรับการจัดทำบันทึกการใช้งาน

การจบเซิร์ฟเวอร์บันทึกการใช้งาน:

คำสั่ง End Job Log Server (ENDLOGSVR) ใช้ในการจบเซิร์ฟเวอร์บันทึกการใช้งานตัวเดียว หรือหลายตัว เซิร์ฟเวอร์บันทึกการใช้งานจะบันทึกบันทึกการใช้งาน สำหรับงานที่อยู่ในสถานะคงค้างบันทึกการใช้งาน ถ้ามีงานเซิร์ฟเวอร์บันทึกการใช้งานที่แอ็คทีฟอยู่มากกว่าหนึ่งงานในขณะนี้ออกคำสั่งนี้ งานเซิร์ฟเวอร์บันทึกการใช้งานทั้งหมดจะถูกจบลง

คุณต้องมีสิทธิพิเศษ job control (*JOBCTL) ในการใช้คำสั่งนี้

สำคัญ: ถ้าคุณต้องการเพียงแต่หยุดการจัดทำบันทึกการใช้งาน เนื่องจาก เช่น บันทึกการใช้งานยาวมากหรือใช้รีซอร์สมากเกินไป ให้ดูหัวข้อที่เกี่ยวข้อง *การหยุดการจัดทำบันทึกการใช้งาน*.

เมื่อใช้คำสั่ง ENDLOGSVR คุณสามารถระบุได้ว่าคุณต้องการให้เซิร์ฟเวอร์จบแบบทันที (ไม่แนะนำ) หรือจบในลักษณะที่ควบคุม

หลักการที่เกี่ยวข้อง

“เซิร์ฟเวอร์บันทึกการใช้งาน” ในหน้า 87

โดยปกติ เซิร์ฟเวอร์บันทึกการใช้งานจะพิมพ์บันทึกการใช้งานของงานไปที่ สพูลไฟล์ คุณสามารถเรดต์บันทึกการใช้งานไปที่ปริ้นเตอร์หรือไปที่ outfile (ถ้ามี การระบุให้ทำโดยใช้คำสั่ง QMHCTLJL, Control job log API) อย่างไรก็ตามแนะนำให้ใช้วิธีการนี้สำหรับการจัดทำบันทึกการใช้งาน

งานที่เกี่ยวข้อง

“การหยุดการจัดทำบันทึกการใช้งานเฉพาะ” ในหน้า 223

ถ้าคุณต้องการหยุดการจัดทำบันทึกการใช้งานเฉพาะชั้นหนึ่ง ห้ามใช้คำสั่ง End Job Log Server (ENDLOGSVR) คำสั่ง ENDLOGSVR จะจบเซิร์ฟเวอร์บันทึกการใช้งานทั้งหมด ซึ่งส่งผลให้หยุดการจัดทำบันทึกการใช้งานทั้งหมด

“การลบไฟล์เอาต์พุตบันทึกการใช้งาน” ในหน้า 226

บันทึกการใช้งานจะถูกลบออกจากระบบ เมื่องานเสร็จสมบูรณ์ตามปกติตามปกติ หรือเมื่อใช้คำสั่ง Remove Pending Job Log (QWTRMVJL) API หรือ End Job (ENDJOB) อย่างใดอย่างหนึ่ง นอกจากนั้น ถ้ามีการระบุ “clear incomplete job logs” บน IPL งานทั้งหมดในบันทึกการใช้งานที่ค้างค้างจะถูกลบออกจากระบบในระหว่างการ IPL สามารถดูไฟล์เอาต์พุตบันทึกการใช้งานที่เหลือได้ข้างใต้ **Basic Operations** → **Printer Output**.

System i Navigator:

1. จาก System i 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 อย่างใดอย่างหนึ่ง

อินเตอร์เฟซแบบอักษร:

คำสั่ง: End Job Log Server (ENDLOGSVR)

การเริ่มเซิร์ฟเวอร์บันทึกการใช้งาน

โดยค่าดีฟอลต์, เซิร์ฟเวอร์ บันทึกการใช้งานจะเริ่มโดยอัตโนมัติเมื่อระบบย่อย QSYSWRK เริ่มต้น คุณสามารถเริ่มเซิร์ฟเวอร์บันทึกการใช้งาน โดยใช้คำสั่ง Start Job Log Server (STRLOGSVR)

เมื่อคุณใช้คำสั่ง STRLOGSVR คุณสามารถระบุจำนวนของเซิร์ฟเวอร์บันทึกการใช้งานเพิ่มเติมที่คุณต้องการเริ่ม หรือคุณสามารถปล่อยให้ระบบคำนวณจำนวนที่คุณต้องการได้ ถ้าจำนวนของเซิร์ฟเวอร์ที่ร้องขอ เกินกว่าจำนวนแอสต์ที่สูงสุดที่ใช้ได้, เฉพาะผลต่างระหว่างจำนวนสูงสุด และจำนวนปัจจุบันของเซิร์ฟเวอร์แอสต์ที่เฟ่านั้นจะถูกเริ่มต้น จำนวนสูงสุดของ เซิร์ฟเวอร์บันทึกการใช้งานที่สามารถแอสต์ที่หรืออยู่บนคิวงานในเวลาเดียวกันคือ 30

หลักการที่เกี่ยวข้อง

“เซิร์ฟเวอร์บันทึกการใช้งาน” ในหน้า 87

โดยปกติ เซิร์ฟเวอร์บันทึกการใช้งานจะพิมพ์บันทึกการใช้งานของงานไปที่ สพูลไฟล์ คุณสามารถเรอดับบันทึกการใช้งานไปที่ปริ้นเตอร์หรือไปที่ outfile (ถ้ามี การระบุให้ทำโดยใช้คำสั่ง QMHCTLJL, Control job log API) อย่างไรก็ตาม ไม่แนะนำให้ใช้วิธีการนี้สำหรับการจัดทำบันทึกการใช้งาน

System i Navigator:

ในการใช้ System i Navigator ให้ใช้ข้อแนะนำต่อไปนี้:

1. จาก System i Navigator คลิกขวาที่ระบบ end point ที่เซิร์ฟเวอร์บันทึกการใช้งานตั้งอยู่ และเลือก **Run Command**.
2. ในฟิลด์ **Command to run:** ให้พิมพ์ STRLOGSVR
3. คลิก **Prompt**.

4. หน้าต่าง Start Job Log Server จะแสดงขึ้นเพื่อช่วยคุณระบุ พารามิเตอร์ของคำสั่งนี้ กรอกข้อมูลในหน้าต่างให้สมบูรณ์ และคลิก **OK**. หน้าต่างปิดและคุณจะถูกส่งกลับไปหน้าต่าง Run Command
5. ขณะนี้คุณสามารถคลิก **OK** เพื่อรันคำสั่งในทันที หรือคุณสามารถคลิก **Schedule** เพื่อจัดตารางเวลา ที่คุณต้องการให้คำสั่งรัน อย่างเป็นอัตโนมัติอย่างหนึ่ง

อินเตอร์เฟซแบบอักษร:

คำสั่ง: Start Log Server (STRLOGSVR)

วิธีการแสดงบันทึกการใช้งาน

คุณสามารถดูบันทึกการใช้งานจากที่ใดก็ได้ภายในการจัดการระบบงานที่คุณเข้าไปใช้งาน เช่น ผ่านทางพื้นที่ระบบย่อย หรือพื้นที่พูลหน่วยความจำ คุณสามารถใช้ System i Navigator หรืออินเตอร์เฟซแบบอักษรเพื่อแสดงบันทึกการใช้งานได้

งานที่เกี่ยวข้อง

“การจัดทำพริ้นเตอร์เอาต์พุตจากบันทึกการใช้งานที่ค้าง” ในหน้า 227

งานที่ไม่มีการตั้งค่า System i Navigator Job Properties - Job Log นั้น ไฟล์ Produce a job log ที่เลือกจะไม่จัดทำบันทึกการใช้งาน แต่บันทึกการใช้งานจะอยู่ในบันทึกการใช้งานที่ค้างค้าง ถ้าต้องการจัดทำพริ้นเตอร์เอาต์พุตจากบันทึกการใช้งานที่อยู่ในบันทึกการใช้งานที่ค้าง ค้าง ให้ใช้อินเตอร์เฟซแบบอักษร

“การลบไฟล์เอาต์พุตบันทึกการใช้งาน” ในหน้า 226

บันทึกการใช้งานจะถูกลบออกจากระบบ เมื่องานเสร็จสมบูรณ์ตามปกติตามปกติ หรือเมื่อใช้คำสั่ง Remove Pending Job Log (QWTRMVJL) API หรือ End Job (ENDJOB) อย่างใดอย่างหนึ่ง นอกจากนั้น ถ้ามีการระบุ "clear incomplete job logs" บน IPL งานทั้งหมดในบันทึกการใช้งานที่ค้างค้างจะถูกลบออกจากระบบในระหว่างการ IPL สามารถดูไฟล์เอาต์พุตบันทึกการใช้งานที่เหลือได้ข้างใต้ **Basic Operations > Printer Output**.

“จะทำอะไรเมื่อบันทึกการใช้งานไม่แสดงขึ้น” ในหน้า 221

ใน System i Navigator ถ้าค้นหา และแสดงบันทึกการใช้งาน ไม่ว่าจะเป็งานแบ็ตช์หรืองานแบบโต้ตอบ ให้คลิกขวาที่งาน แล้วจึงคลิก Job log จากเมนู อย่างไรก็ตาม ขึ้นอยู่กับสถานะของงานของคุณหรือวิธีการตั้งค่าบันทึกการใช้งานในรายละเอียดของงาน, บันทึกการใช้งานของคุณอาจอยู่ในเอาต์พุตคิว หรืออาจอยู่ในสถานะการค้างค้างบันทึกการใช้งาน หรืออาจไม่พร้อมใช้งาน

“การหยุดระบบย่อย” ในหน้า 178

คุณสามารถใช้ System i Navigator หรืออินเตอร์เฟซแบบอักษรเพื่อหยุดหนึ่ง หรือหลายระบบย่อย และระบุสิ่งที่จะเกิดขึ้นกับงานแอ็คทีฟที่กำลังจะประมวลผล ไม่มีงานใหม่ หรือขั้นตอนการเราต์ใหม่ถูกเริ่มในระบบย่อยหลังจากที่ระบบย่อยถูกหยุด

ข้อมูลที่เกี่ยวข้อง

การจัดการการพิมพ์

ค่ากำหนดของระบบงาน: เวลาสูงสุดสำหรับการจบในทันที

System i Navigator:

ถ้าต้องการเข้าใช้บันทึกการใช้งานสำหรับงานแอ็คทีฟหรืองานเซิร์ฟเวอร์ให้ทำต่อไปนี้:

1. จาก System i Navigator, ให้ขยาย My Connections > connection > 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**

อินเตอร์เฟซแบบอักษร:

วิธีแสดงบันทึกการใช้งานขึ้นอยู่กับสถานะของงาน

- การใช้คำสั่ง **Work with Job Logs (WRKJOBLOG)** สามารถใช้เพื่อแสดงบันทึกการใช้งานที่ค้างไว้สำหรับงานเสร็จสิ้นแล้ว, สพูลไฟล์ของบันทึกการใช้งานทั้งหมด หรือทั้งสองอย่าง ตัวอย่างเช่น ในการแสดงรายการบันทึกการใช้งานที่ค้างไว้สำหรับงานทั้งหมดที่จบการใช้คำสั่งต่อไปนี้:

```
WRKJOBLOG JOBLOGSTT(*PENDING)
```

- หากงานยังคงแอกทีฟอยู่ (แบ็ตช์ หรืองานโต้ตอบ) หรือบนคิวงาน และยังไม่เริ่มต้น ให้ใช้ คำสั่ง **Display Job Log (DSPJOBLOG)** ตัวอย่างเช่น ในการแสดงบันทึกการใช้งานของงานโต้ตอบสำหรับผู้ใช้ JSMITH ที่จอภาพ WS1 ให้ใช้คำสั่งต่อไปนี้:

```
DSPJOBLOG JOB(nnnnnn/JSMITH/WS1)
```

โดย nnnnnn เป็นหมายเลขงาน

- หากงานจบลง และยังไม่ได้พิมพ์บันทึกการใช้งาน ให้ใช้ คำสั่ง **Display Spooled File (DSPSPLF)** ตัวอย่างเช่น ในการแสดงบันทึกการใช้งานสำหรับหมายเลขงาน 001293 สำหรับผู้ใช้ FRED ที่จอภาพ WS3 ให้ใช้คำสั่งต่อไปนี้:

```
DSPSPLF FILE(QPJOBLOG) JOB(001293/FRED/WS3)
```

ถ้าคุณมีข้อมูลไม่เพียงพอที่จะใช้คำสั่งข้างต้น คำสั่ง **Work with User Jobs (WRKUSRJOB)** หรือคำสั่ง **Work with Submitted Jobs (WRKSBMJOB)** อาจมีประโยชน์

จะทำอะไรเมื่อบันทึกการใช้งานไม่แสดงขึ้น

ใน System i Navigator ถ้าค้นหา และแสดงบันทึกการใช้งาน ไม่ว่าจะป็นงานแบ็ตช์ หรืองานแบบโต้ตอบ ให้คลิกขวาที่งาน แล้วจึงคลิก **Job log** จากเมนู อย่างไรก็ตาม ขึ้นอยู่กับสถานะของงานของคุณหรือวิธีการตั้งค่าบันทึกการใช้งานในรายละเอียดของงาน, บันทึกการใช้งานของคุณอาจอยู่ในเอดต์พุตคิว หรืออาจอยู่ในสถานะการคงค้างบันทึกการใช้งาน หรืออาจไม่พร้อมใช้งาน

ข้างล่างนี้คือขั้นตอนปฏิบัติถ้าเมนูอ็อปชัน Job log ไม่พร้อมใช้งาน สำหรับงานของคุณ

คำแนะนำ: เซ็ตการแสดงผลคอลัมน์สำหรับ Active jobs (หรือ Server jobs) ให้รวมคอลัมน์ Status สิ่งนี้จะช่วยให้ทราบตำแหน่งที่จะค้นหาบันทึกการใช้งานของคุณได้รวดเร็ว และง่ายขึ้น

ในการเข้าสู่บันทึกการใช้งาน: My Connections → *your system* → Work Management → Active Jobs or Server Jobs → Right-click the job and select Job log.

ถ้าเมนูข้อพจน Job log ไม่พร้อมใช้งาน หรือถ้าคุณได้รับข้อความแสดงความ ผิดพลาดที่ระบุว่า โปรแกรมไม่สามารถดึงข้อมูล บันทึกการใช้งาน ให้พิจารณาทำดัง ต่อไปนี้:

1. ตรวจสอบสถานะของงาน

ทางเลือก	คำอธิบาย
กำลังรันอยู่	ตรวจสอบหน้าต่าง Job Properties – Job Log และตรวจให้แน่ใจว่ามี การเลือกกล่อง Produce a job log ถ้าไม่ได้เลือกกล่องดังกล่าว จะไม่ มีการจัดทำบันทึกการใช้งาน
จบแล้ว	งานนี้ไม่ได้จบในลักษณะ ปกติ อาจเนื่องจากข้อผิดพลาดหรือการยุติ โดยผู้ใช้ คลิกขวาที่งาน แล้วจึงคลิกPrinter Output. ถ้าคุณไม่เห็น บันทึกการใช้งาน ของคุณที่นั่น ให้ตรวจสอบหน้าต่าง Job Properties – Job Log และตรวจให้แน่ใจ ว่ามีการเลือกเช็คบ็อกซ์ Produce printer output for job log
สมบูรณ์ – พรินเตอร์เอาต์พุตพร้อมใช้งาน	งานนี้ จบลงตามปกติ คลิกขวาที่งาน แล้วจึงคลิก Printer Output. ถ้าคุณไม่เห็นบันทึกการใช้งานของคุณที่นั่น ให้ตรวจสอบหน้าต่าง Job Properties – Job Log และตรวจให้แน่ใจว่ามีการเลือกฟิลด์ Create printer output for job log if job ends normally .
สมบูรณ์ – คงค้างบันทึกการใช้งาน	ไม่มีการจัดทำบันทึกการใช้งาน บันทึกการใช้งานยังคงค้างอยู่จนกว่าจะถูกลบออก คุณต้องใช้คำสั่ง Display Job Log (DSPJOBLOG) เพื่อดูบันทึกการใช้งานที่ค้างไว้

2. บันทึกการใช้งานอาจถูกสฟูลไปที่เอาต์พุตคิวและพิมพ์แล้ว ซึ่งทำให้บันทึกถูกลบออกจากระบบ

3. อีกสาเหตุหนึ่งที่เป็นไปได้คือบันทึกการใช้งานได้ถูกลบออกโดย ผู้ใช้อื่น

งานที่เกี่ยวข้อง

“วิธีการแสดงบันทึกการใช้งาน” ในหน้า 220

คุณสามารถดูบันทึกการใช้งานจากที่ใดก็ได้ภายในการจัดการระบบงานที่คุณเข้าไปใช้งาน เช่น ผ่านทางพื้นที่ระบบย่อย หรือพื้นที่พูลหน่วยความจำ คุณสามารถใช้ System i Navigator หรืออินเทอร์เน็ตเฟสแบบอักษระเพื่อแสดงบันทึกการใช้งานได้

การระบุเอาต์พุตคิวสำหรับบันทึกการใช้งาน

โดยค่าดีฟอลต์ ไฟล์พรินเตอร์ที่จะใช้ในการสฟูลบันทึกการใช้งานคือ QPJOBLOG คุณสามารถมีไฟล์พรินเตอร์ QPJOBLOG ได้หลายไฟล์บนระบบของคุณ ใน QSYS เอาต์พุตคิวที่แอดีทริบิวต์ OUTQ จะใช้คือ QEZJOBLOG ในไลบรารี QUSRSYS เมื่อระบบสร้างบันทึกการใช้งาน ระบบจะค้นหาไฟล์พรินเตอร์ QPJOBLOG ในรายชื่อไลบรารีของงาน ไฟล์แรกที่พบคือไฟล์ที่ ระบบจะใช้ คุณใช้อินเทอร์เน็ตเฟสแบบอักษระในการปรับเปลี่ยนการตั้งค่าเหล่านี้

1. การเปลี่ยนแอดีทริบิวต์ไฟล์พรินเตอร์ QPJOBLOG OUTQ เป็น *JOB

a. คำสั่ง: Change Printer File CHGPRTF FILE(QPJOBLOG) OUTQ(*JOB)

2. การเปลี่ยนแอดีทริบิวต์ OUTQ ของงานเป็นเอาต์พุตคิวที่คุณต้องการ คุณสามารถทำสิ่งนี้ได้โดยการใช้อินเทอร์เน็ตเฟสแบบ อักษระ หรือ System i Navigator.

a. คำสั่ง: Change Job CHGJOB OUTQ(MYLIB/MYOUTQ)

b. System i Navigator: **Work Management** → **Active Jobs** → **คลิกขวาที่งาน และเลือก Properties** → **Printer Tab**
ข้อมูลที่เกี่ยวข้อง

การควบคุมการพิมพ์ไปที่เอาต์พุตคิวหรือพรินเตอร์

การหยุดการจัดทำบันทึกการใช้งานเฉพาะ

ถ้าคุณต้องการหยุดการจัดทำบันทึกการใช้งานเฉพาะชิ้นหนึ่ง ห้ามใช้คำสั่ง End Job Log Server (ENDLOGSVR) คำสั่ง ENDLOGSVR จะจบเซิร์ฟเวอร์บันทึกการใช้งานทั้งหมด ซึ่งส่งผลให้หยุดการจัดทำบันทึกการใช้งานทั้งหมด

แต่ให้ใช้ขั้นตอนต่อไปนี้ในการหยุดการจัดทำบันทึกการใช้งาน เฉพาะแทน

1. จาก System i Navigator ให้คลิกขวาที่งานที่คุณต้องการหยุดการจัดทำบันทึกการใช้งาน และคลิก **Properties**. (**My Connections** → **connection** → **Work Management** → **Active Jobs or Server Jobs**)
2. คลิกแท็บ **Job Log**.
3. ยกเลิกการเลือกกล่อง **Produce a job log** และคลิก **OK**.

การจัดทำบันทึกการใช้งานจะยุติลงและบันทึกการใช้งานจะอยู่ในสถานะ บันทึกการใช้งานคงค้าง

หลักการที่เกี่ยวข้อง

“วิธีการสร้างบันทึกการใช้งาน” ในหน้า 85

บันทึกการใช้งานจะ พร้อมให้ใช้งานได้เมื่อต้องการ และไม่มีการสร้างบันทึกการใช้งาน ที่ไม่ต้องการงานที่เกี่ยวข้อง

“การจบเซิร์ฟเวอร์บันทึกการใช้งาน” ในหน้า 218

คำสั่ง End Job Log Server (ENDLOGSVR) ใช้ในการจบเซิร์ฟเวอร์บันทึกการใช้งานตัวเดียว หรือหลายตัว เซิร์ฟเวอร์บันทึกการใช้งานจะบันทึกบันทึกการใช้งาน สำหรับงานที่อยู่ในสถานะคงค้างบันทึกการใช้งาน ถ้ามีงานเซิร์ฟเวอร์บันทึกการใช้งานที่แอคทีฟอยู่มากกว่าหนึ่งงานในขณะนี้ออกคำสั่งนี้ งานเซิร์ฟเวอร์บันทึกการใช้งานทั้งหมดจะถูกจบลง

“การลบไฟล์เอาต์พุตบันทึกการใช้งาน” ในหน้า 226

บันทึกการใช้งานจะถูกลบออกจากระบบ เมื่องานเสร็จสมบูรณ์ตามปกติตามปกติ หรือเมื่อใช้คำสั่ง Remove Pending Job Log (QWTRMVJL) API หรือ End Job (ENDJOB) อย่างใดอย่างหนึ่ง นอกจากนั้น ถ้ามีการระบุ "clear incomplete job logs" บน IPL งานทั้งหมดในบันทึกการใช้งานที่คงค้างจะถูกลบออกจากระบบในระหว่างการ IPL สามารถดูไฟล์เอาต์พุตบันทึกการใช้งานที่เหลือได้ข้างใต้ **Basic Operations** → **Printer Output**.

“การควบคุมข้อมูลบันทึกการใช้งานแบ็คซ์” ในหน้า 225

สำหรับแบ็คซ์แอฟพลิเคชัน คุณอาจต้องการเปลี่ยนจำนวนของข้อมูลที่บันทึก ระดับไฟล์บันทึก (LOG(40 *NOLIST)) ที่ระบุในรายละเอียดของงาน สำหรับระบบย่อยที่ได้มาจาก IBM QBATCH นับเป็นไฟล์บันทึกที่สมบูรณ์ ถ้างานจบลงอย่างผิดปกติ ถ้างานจบลงอย่างปกติ จะไม่มีการจัดทำบันทึกการใช้งาน

การป้องกันการจัดทำบันทึกการใช้งาน

การป้องกันการจัดทำบันทึกการใช้งานเป็นสิ่งที่มีความจำเป็น ถ้าคุณทราบอยู่แล้วว่า คุณจะไม่ต้องการบันทึกการใช้งานและคุณต้องการสงวนรีซอร์สของระบบ เมื่อคุณระบุว่าคุณไม่ต้องการให้จัดทำบันทึกการใช้งาน บันทึกการใช้งานจะไม่ถูกจัดทำ และยังคงค้างอยู่จนกว่าจะถูกลบโดยคำสั่ง Remove Pending Job Log (QWTRMVJL) หรือคำสั่ง End Job (ENDJOB)

ในการป้องกันการจัดทำบันทึกการใช้งาน ให้ใช้ข้อแนะนำต่อไปนี้:

1. ใน System i Navigator ให้เปิดหน้าต่าง **Job Properties – Job Log** (**My Connections** → **connection** → **Work Management** → **Active Jobs** (หรือ **System Jobs**) → **คลิกขวางาน** → **Properties** → **แท็บ Job Log**)

2. ยกเลิกการเลือกกล่อง Produce a job log และคลิก OK.

หลักการที่เกี่ยวข้อง

“วิธีการสร้างบันทึกการใช้งาน” ในหน้า 85

บันทึกการใช้งานจะ พร้อมให้ใช้งานได้เมื่อต้องการ และไม่มีการสร้างบันทึกการใช้งาน ที่ไม่ต้องการงานที่เกี่ยวข้อง

“การลบไฟล์เอาต์พุตบันทึกการใช้งาน” ในหน้า 226

บันทึกการใช้งานจะถูกลบออกจากระบบ เมื่องานเสร็จสมบูรณ์ตามปกติตามปกติ หรือเมื่อใช้คำสั่ง Remove Pending Job Log (QWTRMVJL) API หรือ End Job (ENDJOB) อย่างใดอย่างหนึ่ง นอกจากนั้น ถ้ามีการระบุ "clear incomplete job logs" บน IPL งานทั้งหมดในบันทึกการใช้งานที่ค้างจะถูกลบออกจากระบบในระหว่างการ IPL สามารถดูไฟล์เอาต์พุตบันทึกการใช้งานที่เหลือได้ข้างใต้ **Basic Operations** → **Printer Output**.

“การควบคุมข้อมูลบันทึกการใช้งานแบ็ตซ์” ในหน้า 225

สำหรับแบ็ตซ์แอฟพลิเคชัน คุณอาจต้องการเปลี่ยนจำนวนของข้อมูลที่บันทึก ระดับไฟล์บันทึก (LOG(40 *NOLIST)) ที่ระบุในรายละเอียดของงาน สำหรับระบบย่อยที่ได้มาจาก IBM QBATCH นับเป็นไฟล์บันทึกที่สมบูรณ์ ถ้างานจบลงอย่างผิดปกติ ถ้างานจบลงอย่างปกติ จะไม่มีการจัดทำบันทึกการใช้งาน

การควบคุมข้อมูลในบันทึกการใช้งาน

เมื่อจัดการกับปัญหา คุณอาจต้องการบันทึกจำนวนสูงสุดของ ข้อมูลสำหรับงานที่เกิดปัญหาขึ้นบ่อย อีกทางหนึ่งคือ คุณอาจจะ ไม่ต้องการสร้างบันทึกการใช้งานสำหรับงานที่จบลงตามปกติ หรือ คุณอาจไม่ต้องการรวมข้อมูล

คุณสามารถควบคุมข้อมูลที่จะใส่เพิ่มลงในบันทึกการใช้งานได้ โดยการตั้งค่า ระดับข้อความ, ความรุนแรงของข้อความ หรือ คาร์ระดับเนื้อความใน รายละเอียดของงาน อย่างไรก็ตาม ถ้าคุณต้องการควบคุมข้อมูลที่จะบันทึกลงในบันทึกการใช้งานของงานเฉพาะ ให้ใช้หน้าต่าง **Job Properties – Job Log** ใน System i Navigator.

หน้าต่างนี้อ่อนุญาตให้คุณ ควบคุมสิ่งต่างๆ ดังนี้:

- จะจัดทำบันทึกการใช้งานหรือไม่ และจัดทำบันทึกการใช้งานด้วยวิธีการใด
- จะทำอะไรเมื่อใช้งานถึงขนาดสูงสุด
- จะลืกรคำสั่งจากโปรแกรม CL หรือไม่
- จะเก็บรักษาข้อความในบันทึกการใช้งานหรือไม่ และข้อความอะไรที่ควร เก็บรักษาไว้ (ระดับการลืกรและความรุนแรงของข้อความ)
- จะสร้างพรินเตอร์เอาต์พุตสำหรับบันทึกการใช้งานหรือไม่ถ้างานจบลงอย่างปกติ และจะพิมพ์อะไร

ถ้าต้องการใช้หน้าต่าง Job Properties – Job Log ให้ปฏิบัติตามขั้นตอนต่อไปนี้:

1. จาก System i Navigator, ให้เปิดหน้าต่าง Job Properties ของงาน และคลิกแท็บ **Job Log My Connections** → **Connection** → **Work Management** → **Active Job** → **คลิกขวาที่** → **Properties**.
2. สำหรับรายละเอียดเพิ่มเติมเกี่ยวกับอ็อปชันต่างๆ ที่พร้อมใช้บนหน้าต่างนี้ ให้อ้างอิงคำอธิบายทางออนไลน์

หลักการที่เกี่ยวข้อง

“วิธีการสร้างบันทึกการใช้งาน” ในหน้า 85

บันทึกการใช้งานจะ พร้อมให้ใช้งานได้เมื่อต้องการ และไม่มีการสร้างบันทึกการใช้งาน ที่ไม่ต้องการงานที่เกี่ยวข้อง

“การล้างบันทึกใช้งานที่ค้างค้าง” ในหน้า 228

การล้างข้อมูล หรือการลบงานออกจากบันทึกการใช้งานที่ค้างค้างสามารถทำได้หลายวิธี คุณสามารถจบงานด้วยค่า 0 สำหรับพารามิเตอร์ Maximum log entries (LOGLMT) ถ้างานจบลงแล้ว คุณสามารถรัน Remove Pending Job Log (QWTRMVJL) API คุณยังสามารถใช้คำสั่ง Work with Job Logs (WRKJOBLOG)

“การลบไฟล์เอาต์พุตบันทึกการใช้งาน” ในหน้า 226

บันทึกการใช้งานจะถูกลบออกจากระบบ เมื่องานเสร็จสมบูรณ์ตามปกติตามปกติ หรือเมื่อใช้คำสั่ง Remove Pending Job Log (QWTRMVJL) API หรือ End Job (ENDJOB) อย่างใดอย่างหนึ่ง นอกจากนั้น ถ้ามีการระบุ “clear incomplete job logs” บน IPL งานทั้งหมดในบันทึกการใช้งานที่ค้างค้างจะถูกลบออกจากระบบในระหว่างการ IPL สามารถดูไฟล์เอาต์พุตบันทึกการใช้งานที่เหลือได้ข้างใต้ **Basic Operations → Printer Output.**

การเปลี่ยนระดับไฟล์บันทึกของงาน:

ระดับไฟล์บันทึกของงานคือระดับตัวเลขที่กำหนดให้กับ ชนิดข้อความที่จะบันทึก คุณสามารถเปลี่ยนระดับไฟล์บันทึก ในรายละเอียดของงานได้โดยการใช้อินเตอร์เฟซแบบอักษร อย่างไรก็ตาม ถ้าคุณต้องการเปลี่ยนระดับไฟล์บันทึกของงานเฉพาะ ให้ใช้หน้าต่าง **Job Properties – Job Log** ใน System i Navigator.

ถ้าต้องการเข้าใช้งานหน้าต่าง **Job Properties – Job Log** ให้ปฏิบัติตามขั้นตอนดังต่อไปนี้:

1. จาก System i Navigator, ให้ขยาย **My Connections → Work Management → Active Jobs.**
2. เลือกงานและคลิกขวาที่ **Properties**
3. จากหน้าต่าง properties ของงานเฉพาะ ให้เลือกแท็บ **Job Log** และเปลี่ยนระดับการบันทึก

หลักการที่เกี่ยวข้อง

“ข้อความ” ในหน้า 89

ข้อความประกอบด้วย ชื่องาน, ชนิดข้อความ, วันที่และเวลาที่ส่ง, การดำเนินการที่เกิดขึ้น และการดำเนินการที่จำเป็นเพื่อแก้ไขปัญหาลักษณะนี้มีประโยชน์เมื่อคุณพยายามแก้ไขปัญหาใดๆ ที่อาจเกิดขึ้นบนเซิร์ฟเวอร์ของคุณ คุณสามารถเข้าใช้งานบันทึกการใช้งานสำหรับงานเซิร์ฟเวอร์ผ่านทาง System i Navigator. ข้อความแบ่งเป็นสองประเภท ข้อความที่แจ้งเตือนได้ และข้อความที่บันทึกใน บันทึกการใช้งาน

“บันทึกการใช้งาน” ในหน้า 84

บันทึกการใช้งานมีข้อมูลที่เกี่ยวข้องกับคำขอที่ป้อนสำหรับงาน บันทึกการใช้งานมีสองรูปแบบ รูปแบบการคงค้างและรูปแบบสูญ

งานที่เกี่ยวข้อง

“การล้างบันทึกใช้งานที่ค้างค้าง” ในหน้า 228

การล้างข้อมูล หรือการลบงานออกจากบันทึกการใช้งานที่ค้างค้างสามารถทำได้หลายวิธี คุณสามารถจบงานด้วยค่า 0 สำหรับพารามิเตอร์ Maximum log entries (LOGLMT) ถ้างานจบลงแล้ว คุณสามารถรัน Remove Pending Job Log (QWTRMVJL) API คุณยังสามารถใช้คำสั่ง Work with Job Logs (WRKJOBLOG)

การควบคุมข้อมูลบันทึกการใช้งานแบ็ตซ์:

สำหรับแบ็ตซ์เอ็นพีเคชัน คุณอาจต้องการเปลี่ยนจำนวนของข้อมูลที่บันทึก ระดับไฟล์บันทึก (LOG(40 *NOLIST)) ที่ระบุในรายละเอียดของงาน สำหรับระบบย่อยที่ได้มาจาก IBM QBATCH นับเป็นไฟล์บันทึกที่สมบูรณ์ ถ้างานจบลงอย่างผิดปกติ ถ้างานจบลงอย่างปกติ จะไม่มีการจัดทำบันทึกการใช้งาน

สามารถควบคุมบันทึกการใช้งานที่ระดับคิวงาน (QBATCH) โดยการปรับ การตั้งค่าบันทึกการใช้งานสำหรับงานระบบย่อย QBATCH อีพชั่นในการควบคุมวิธีการจัดทำ บันทึกการใช้งานที่ระดับงานระบบย่อยจะเหมือนกับอีพชั่นในการควบคุม บันทึกการใช้งานที่ระดับงานปกติ

ถ้าต้องการปรับเปลี่ยนการตั้งค่าบันทึกการใช้งานสำหรับระบบย่อยคิวงาน ให้ทำดังต่อไปนี้:

จาก System i Navigator, เปิด **Subsystem Properties – Job Log** หน้าต่างสำหรับระบบย่อยคิวงาน (**Work Management** → **Subsystems** → **Active Subsystems** → **QBATCH** → **คลิกขวาที่งาน QBATCH** → **Properties** → **แท็บ Job Log**)

หมายเหตุ: หากคุณยกเลิก **Produce a job log field** ฟิลด์ (*PND) สำหรับระบบย่อย บันทึกการใช้งานที่ระบุที่ระบบย่อยจะไม่ ถูกแสดงรายการพร้อมกับพริ้นเตอร์เอาต์พุตอื่น จากนั้น คุณต้องใช้คำสั่ง **Display Job Log (DSPJOBLOG)** เพื่อดู บันทึกการใช้งานที่ค้างไว้

ถ้างานแบ็ตช์กำลังรันบนโปรแกรม CL คำสั่งโปรแกรม CL จะถูกบันทึกเฉพาะเมื่อ **LOGCLPGM(*YES)** ถูกระบุบนคำสั่ง **Create Control Language Program (CRTCLPGM)** หรือคำสั่ง **Change Program (CHGPGM)**

หลักการที่เกี่ยวข้อง

“บันทึกการใช้งาน” ในหน้า 84

บันทึกการใช้งานมีข้อมูลที่เกี่ยวข้องกับคำขอที่ป้อนสำหรับ งาน บันทึกการใช้งานมีสองรูปแบบ รูปแบบการคงค้างและรูปแบบสพูล

งานที่เกี่ยวข้อง

“การลบไฟล์เอาต์พุตบันทึกการใช้งาน”

บันทึกการใช้งานจะถูกลบออกจากระบบ เมื่องานเสร็จสมบูรณ์ตามปกติตามปกติ หรือเมื่อใช้คำสั่ง **Remove Pending Job Log (QWTRMVJL)** API หรือ **End Job (ENDJOB)** อย่างใดอย่างหนึ่ง นอกจากนั้น ถ้ามีการระบุ “clear incomplete job logs” บน IPL งานทั้งหมดในบันทึกการใช้งานที่คงค้างจะถูกลบออกจากระบบในระหว่างการ IPL สามารถดูไฟล์เอาต์พุต บันทึกการใช้งานที่เหลือได้ข้างใต้ **Basic Operations** → **Printer Output**.

“การหยุดการจัดทำบันทึกการใช้งานเฉพาะ” ในหน้า 223

ถ้าคุณต้องการหยุดการจัดทำบันทึกการใช้งานเฉพาะชิ้นหนึ่ง ห้ามใช้คำสั่ง **End Job Log Server (ENDLOGSVR)** คำสั่ง **ENDLOGSVR** จะจบเซิร์ฟเวอร์บันทึกการใช้งานทั้งหมด ซึ่งส่งผลให้หยุดการจัดทำบันทึกการใช้งานทั้งหมด

“การป้องกันการจัดทำบันทึกการใช้งาน” ในหน้า 223

การป้องกันการจัดทำบันทึกการใช้งานเป็นสิ่งที่มีความปลอดภัย ถ้าคุณทราบอยู่แล้วว่า คุณจะไม่ต้องการบันทึกการใช้งานและ คุณต้องการสแกนรีเซ็ตของระบบ เมื่อคุณระบุว่าคุณไม่ต้องการให้จัดทำบันทึกการใช้งาน บันทึกการใช้งานจะไม่ถูกจัดทำ และยังคงค้างอยู่บนกว่าจะถูกลบโดยคำสั่ง **Remove Pending Job Log (QWTRMVJL)** หรือคำสั่ง **End Job (ENDJOB)**

การลบไฟล์เอาต์พุตบันทึกการใช้งาน

บันทึกการใช้งานจะถูกลบออกจากระบบ เมื่องานเสร็จสมบูรณ์ตามปกติตามปกติ หรือเมื่อใช้คำสั่ง **Remove Pending Job Log (QWTRMVJL)** API หรือ **End Job (ENDJOB)** อย่างใดอย่างหนึ่ง นอกจากนั้น ถ้ามีการระบุ “clear incomplete job logs” บน IPL งานทั้งหมดในบันทึกการใช้งานที่คงค้างจะถูกลบออกจากระบบในระหว่างการ IPL สามารถดูไฟล์เอาต์พุตบันทึกการใช้งานที่เหลือได้ข้างใต้ **Basic Operations** → **Printer Output**.

เพื่อลบบันทึกการใช้งานที่พบใน **Printer Output** ให้คลิกขวาที่ชื่อไฟล์ของบันทึกการใช้งานที่คุณต้องการลบออก และคลิก **Delete**.

วิธีการพิจารณาว่าการลบบันทึกการใช้งานมีความปลอดภัยหรือไม่

การตัดสินใจว่าจะเก็บรักษาบันทึกการใช้งานหรือจะลบออกถือเป็นสิ่งที่ท้าทาย บันทึกการใช้งาน คือสิ่งที่คุณจำเป็นต้องเก็บไว้ เพื่อให้คุณสามารถแก้ไขปัญหาได้ บันทึกการใช้งาน คือสิ่งที่คุณไม่ต้องการเก็บเนื่องจากทำให้ระบบของคุณยุ่งเหยิง เมื่อตัดสินใจว่าจะลบบันทึกการใช้งานใด หรือจะป้องกันการลบบันทึกการใช้งานใด ให้พิจารณาแนวทางดังต่อไปนี้:

- นี่เป็นงานที่คุณสามารถแก้ไขได้ง่ายโดยไม่ต้องดูที่บันทึกการใช้งานหรือไม่?
- นี่เป็นงานที่คล้ายกับงานอื่นในระบบหรือไม่? ถ้างานล้มเหลวงานที่คล้ายกันอื่นจะล้มเหลวด้วยหรือไม่? ถ้าใช่ คุณอาจต้องเก็บงานที่สร้างบันทึกการใช้งาน เพียงงานเดียวเท่านั้น

หลักการที่เกี่ยวข้อง

“บันทึกการใช้งาน” ในหน้า 84

บันทึกการใช้งานมีข้อมูลที่เกี่ยวข้องกับคำขอที่ป้อนสำหรับงาน บันทึกการใช้งานมีสองรูปแบบ รูปแบบการคงค้างและรูปแบบสพูล

งานที่เกี่ยวข้อง

“วิธีการแสดงบันทึกการใช้งาน” ในหน้า 220

คุณสามารถดูบันทึกการใช้งานจากที่ใดก็ได้ภายในการจัดการระบบงานที่คุณเข้าไปใช้งาน เช่น ผ่านทางพื้นที่ระบบย่อย หรือพื้นที่พูลหน่วยความจำ คุณสามารถใช้ System i Navigator หรืออินเทอร์เฟซแบบอักษรเพื่อแสดงบันทึกการใช้งานได้

“การจบเซิร์ฟเวอร์บันทึกการใช้งาน” ในหน้า 218

คำสั่ง End Job Log Server (ENDLOGSVR) ใช้ในการจบเซิร์ฟเวอร์บันทึกการใช้งานตัวเดียว หรือหลายตัว เซิร์ฟเวอร์บันทึกการใช้งานจะบันทึกบันทึกการใช้งานสำหรับงานที่อยู่ในสถานะคงค้างบันทึกการใช้งาน ถ้ามีงานเซิร์ฟเวอร์บันทึกการใช้งานที่แอคทีฟอยู่มากกว่าหนึ่งงานในขณะนี้ออกคำสั่งนี้ งานเซิร์ฟเวอร์บันทึกการใช้งานทั้งหมดจะถูกจบลง

“การหยุดการจัดทำบันทึกการใช้งานเฉพาะ” ในหน้า 223

ถ้าคุณต้องการหยุดการจัดทำบันทึกการใช้งานเฉพาะขึ้นหนึ่ง ห้ามใช้คำสั่ง End Job Log Server (ENDLOGSVR) คำสั่ง ENDLOGSVR จะจบเซิร์ฟเวอร์บันทึกการใช้งานทั้งหมด ซึ่งส่งผลให้หยุดการจัดทำบันทึกการใช้งานทั้งหมด

“การป้องกันการจัดทำบันทึกการใช้งาน” ในหน้า 223

การป้องกันการจัดทำบันทึกการใช้งานเป็นสิ่งที่มีความจำเป็น ถ้าคุณทราบอยู่แล้วว่า คุณจะไม่ต้องการบันทึกการใช้งานและคุณต้องการสงวนรีซอร์สของระบบ เมื่อคุณระบุว่าคุณไม่ต้องการให้จัดทำบันทึกการใช้งาน บันทึกการใช้งานจะไม่ถูกจัดทำ และยังคงค้างอยู่จนกว่าจะถูกลบโดยคำสั่ง Remove Pending Job Log (QWTRMVJL) หรือคำสั่ง End Job (ENDJOB)

“การควบคุมข้อมูลในบันทึกการใช้งาน” ในหน้า 224

เมื่อจัดการกับปัญหา คุณอาจต้องการบันทึกจำนวนสูงสุดของ ข้อมูลสำหรับงานที่เกิดปัญหาขึ้นบ่อย อีกทางหนึ่งคือ คุณอาจไม่ต้องการสร้างบันทึกการใช้งานสำหรับงานที่จบลงตามปกติ หรือ คุณอาจไม่ต้องการรวมข้อมูล

“การควบคุมข้อมูลบันทึกการใช้งานแบ็ตซ์” ในหน้า 225

สำหรับแบ็ตซ์แอ็พพลิเคชัน คุณอาจต้องการเปลี่ยนจำนวนของข้อมูลที่บันทึก ระดับไฟล์บันทึก (LOG(40 *NOLIST)) ที่ระบุในรายละเอียดของงาน สำหรับระบบย่อยที่ได้มาจาก IBM QBATCH นับเป็นไฟล์บันทึกที่สมบูรณ์ ถ้างานจบลงอย่างผิดปกติ ถ้างานจบลงอย่างปกติ จะไม่มีการจัดทำบันทึกการใช้งาน

การจัดทำพรินเตอร์เอาต์พุตจากบันทึกการใช้งานที่คงค้าง

งานที่ไม่มีการตั้งค่า System i Navigator Job Properties – Job Log นั้น ไฟล์ Produce a job log ที่เลือกจะไม่จัดทำบันทึกการใช้งาน แต่บันทึกการใช้งานจะอยู่ในบันทึกการใช้งานที่คงค้าง ถ้าต้องการจัดทำพรินเตอร์เอาต์พุตจากบันทึกการใช้งานที่อยู่ในบันทึกการใช้งานที่คง ค้าง ให้ใช้อินเทอร์เฟซแบบอักษร

คำสั่ง: Display Job Log (DSPJOBLOG)

หลักการที่เกี่ยวข้อง

“การคงค้างบันทึกการใช้งาน” ในหน้า 86

สภาพการคงค้างบันทึกการใช้งานมีใช้กันเป็นเวลาหลายปีแล้ว เมื่อแอ็ดทริบิวต์บันทึกการใช้งานของงานเป็น *PND จะไม่มีการสร้างบันทึกการใช้งาน คุณสามารถควบคุมวิธีการ และสถานการณ์ที่มีการสร้างบันทึกการใช้งานสำหรับงานเฉพาะได้งานที่เกี่ยวข้อง

“วิธีการแสดงบันทึกการใช้งาน” ในหน้า 220

คุณสามารถดูบันทึกการใช้งานจากที่ใดก็ได้ภายในการจัดการระบบงานที่คุณเข้าไปใช้งาน เช่น ผ่านทางพื้นที่ระบบย่อย หรือพื้นที่พูลหน่วยความจำ คุณสามารถใช้ System i Navigator หรืออินเทอร์เน็ตเฟสแบบอักษรเพื่อแสดงบันทึกการใช้งานได้

การล้างบันทึกใช้งานที่คงค้าง

การล้างข้อมูล หรือการลบงานออกจากบันทึกการใช้งานที่คงค้างสามารถทำได้หลายวิธี คุณสามารถจบงานด้วยค่า 0 สำหรับพารามิเตอร์ Maximum log entries (LOGMLT) ถ้างานจบลงแล้ว คุณสามารถรัน Remove Pending Job Log (QWTRMVJL) API คุณยังสามารถใช้คำสั่ง Work with Job Logs (WRKJOBLOG)

ในการจบงานด้วย LOGMLT เช็ตเป็น 0 ให้ใช้ System i Navigator หรืออินเทอร์เน็ตเฟสแบบอักษร

หลักการที่เกี่ยวข้อง

“การคงค้างบันทึกการใช้งาน” ในหน้า 86

สภาพการคงค้างบันทึกการใช้งานมีใช้กันเป็นเวลาหลายปีแล้ว เมื่อแอ็ดทริบิวต์บันทึกการใช้งานของงานเป็น *PND จะไม่มีการสร้างบันทึกการใช้งาน คุณสามารถควบคุมวิธีการ และสถานการณ์ที่มีการสร้างบันทึกการใช้งานสำหรับงานเฉพาะได้งานที่เกี่ยวข้อง

“การควบคุมข้อมูลในบันทึกการใช้งาน” ในหน้า 224

เมื่อจัดการกับปัญหา คุณอาจต้องการบันทึกจำนวนสูงสุดของ ข้อมูลสำหรับงานที่เกิดปัญหาขึ้นบ่อย อีกทางหนึ่งคือ คุณอาจไม่ต้องการสร้างบันทึกการใช้งานสำหรับงานที่จบลงตามปกติ หรือ คุณอาจไม่ต้องการรวมข้อมูล

“การเปลี่ยนระดับไฟล์บันทึกของงาน” ในหน้า 225

ระดับไฟล์บันทึกของงานคือระดับตัวเลขที่กำหนดให้กับ ชนิดข้อความที่จะบันทึก คุณสามารถเปลี่ยนระดับไฟล์บันทึกในรายละเอียดของงานได้โดยการใช้อินเทอร์เน็ตเฟสแบบอักษร อย่างไรก็ตาม ถ้าคุณต้องการเปลี่ยนระดับไฟล์บันทึกของงานเฉพาะ ให้ใช้หน้าต่าง Job Properties - Job Log ใน System i Navigator.

ข้อมูลที่เกี่ยวข้อง

คำสั่ง Change Cleanup (CHGCLNUP)

โปรแกรมออกสำหรับปรับการล้างอัตโนมัติ

System i Navigator:

1. จาก System i Navigator ให้ขยาย Work Management → Active Jobs.
2. ระบุตำแหน่งงานที่คุณต้องการจบ
3. คลิกขวาที่งานและคลิก Delete/End.
4. บนหน้าต่าง Confirm Delete/End ให้เช็ต Delete printer output เป็น No.
5. จัดการหน้าต่าง Confirm Delete/End ให้สมบูรณ์ และคลิก Delete.

อินเทอร์เน็ตเฟสแบบอักษร:

คำสั่ง: End Job (ENDJOB LOGMLT(0))

การจัดการ job accounting

ฟังก์ชัน job accounting ไม่แฉีกที่พีโดยค่าดีฟอลต์ ต้องใช้ขั้นตอนเริ่มต้น บางอย่างเพื่อตั้งค่าฟังก์ชันนี้ ข้อมูลต่อไปนี้อธิบายวิธีการตั้งค่า job accounting และปฏิบัติการกิจทั่วไปส่วนใหญ่ที่เกี่ยวข้องกับ job accounting

หลักการที่เกี่ยวข้อง

“Job accounting” ในหน้า 94

ฟังก์ชัน job accounting รวบรวมข้อมูลเพื่อให้คุณสามารถกำหนดว่าใครกำลังใช้ระบบของคุณและเขากำลังใช้รีซอร์สระบบอะไรอยู่ และยังช่วยคุณในการประเมินการใช้โดยรวมของระบบของคุณ Job accounting เป็นสิ่งที่เลือกได้ คุณต้องทำขั้นตอนพิเศษเพื่อตั้งค่า job accounting คุณสามารถร้องขอให้ระบบรวบรวมข้อมูลการจัดทำบัญชีรีซอร์สของงาน, ข้อมูลการจัดทำบัญชี พรินเตอร์ไฟล์ หรือทั้งสองอย่าง คุณยังสามารถกำหนด accounting codes ที่โปรไฟล์ผู้ใช้ หรืองานเฉพาะได้

ข้อมูลที่เกี่ยวข้อง

การจัดการเจอร์นัล

การตั้งค่าการทำเจอร์นัล

การตั้งค่า job accounting

ถ้าต้องการตั้งค่า job accounting ให้ใช้อินเตอร์เฟซแบบอักษร

1. การสร้าง journal receiver สามารถสร้าง journal receiver ด้วยชื่อ และไลบรารีใดๆ ที่คุณเลือก ขณะแนะนำให้ใช้ชื่อตามหลักการตั้งชื่อ เช่น ACGJRN1 เพื่อให้สามารถสร้าง receiver 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 และคุณต้องมีสิทธิในการใส่เพิ่มอ็อบเจกต์ที่ไลบรารี QSYS

a. คำสั่ง: Create Journal (CRTJRN)

CRTJRN JRN(QSYS/QACGJRN) JRNRCV(USERLIB/ACGJRN1) AUT(*EXCLUDE)

journal receiver ควรเหมือนกับ receiver ที่สร้างในขั้นตอน 1 สามารถเซตสิทธิเป็นสิ่งที่ใดๆ ที่คุณเลือก แต่แนะนำให้ใช้ *EXCLUDE เนื่องจากสามารถใช้ข้อมูลที่รวบรวมเพื่อคิดค่าธรรมเนียมสำหรับการใช้รีซอร์สจากผู้ใช้ได้

3. การเปลี่ยนค่ากำหนดของระบบ journal accounting information (QACGLVL) สามารถเซตค่ากำหนดของระบบเป็นข้อมูล journal job accounting หรือข้อมูล พรินเตอร์ หรือทั้งสองอย่าง *JOB จะจัดทำ job (JB) journal entry ในขณะที่ *PRINT จะจัดทำ direct print (DP) หรือ spooled print (SP) journal entry ค่า *NONE หมายความว่า ไม่มีการทำเจอร์นัลสำหรับเจอร์นัล QACGJRN ข้อมูล Job accounting จะถูกเจอร์นัล เฉพาะสำหรับงานที่เริ่มหลังจากที่ได้เซตค่ากำหนดของระบบเป็นค่าอื่นที่ไม่ใช่ *NONE แล้วเท่านั้น

a. คำสั่ง: Work with System Values (WRKSYSVAL) or Change System Value (CHGSYSVAL)

CHGSYSVAL SYSVAL(QACGLVL) VALUE(' *JOB *PRINT')

4. เซตพารามิเตอร์ accounting code ACGCDE สำหรับแต่ละโปรไฟล์ผู้ใช้ สามารถเซต accounting code เป็นสตริงตัวอักษรผสมตัวเลข ที่ยาวได้ถึง 15 อักขระ หากการกำหนดผู้ใช้ปัจจุบันเป็นสิ่งสำคัญสำหรับการวิเคราะห์ job accounting journal entry ของคุณ ขอแนะนำให้คุณเซตพารามิเตอร์ 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 (CHGJOB) หรือ Create Job Description (CRTJOB)

accounting code ดีฟอลต์สำหรับ รายละเอียดของงานคือ *USRPRF ซึ่งหมายความว่าระบบจะใช้ accounting code จากโปรไฟล์ผู้ใช้ของงาน ถ้ามีการระบุค่าอื่นที่ไม่ใช่ *USRPRF ในรายละเอียดของงาน ค่านี้จะนำหน้า accounting code ที่ระบุ ในโปรไฟล์ผู้ใช้

หลักการที่เกี่ยวข้อง

“เกี่ยวกับ accounting code” ในหน้า 99

Initial accounting code (ยาวได้ถึง 15 อักขระ) ของงานที่กำหนดโดยค่าพารามิเตอร์ ACGCDE (accounting code) ในรายละเอียดของงาน และโปรไฟล์ผู้ใช้สำหรับงาน

การควบคุมการกำหนดโค้ดบัญชี

ส่วนที่สำคัญของแอปพลิเคชันการประมวลผลข้อมูลใดๆ คือ การตรวจสอบให้มั่นใจว่ามีการระบุฟิลด์การควบคุมที่ถูกต้องสำหรับ job accounting codes อาจต้องใช้ ฟังก์ชันการตรวจสอบความถูกต้องที่ซับซ้อน ซึ่งไม่เพียงแต่ตรวจสอบเฉพาะ การมีอยู่จริงของโค้ดการพิสูจน์ตัวตนเท่านั้น แต่ยังตรวจสอบผู้ใช้ที่ได้รับอนุญาต ให้ใช้โค้ดที่ระบุด้วย

สามารถกำหนดโค้ดบัญชีผู้ใช้ได้ในพื้นที่ดังต่อไปนี้:

- โปรไฟล์ผู้ใช้
- รายละเอียดของงาน
- In a job (Change Accounting Code (CHGACGCDE) command)

ถ้าจำเป็นต้องควบคุมการกำหนดโค้ดบัญชีผู้ใช้, ให้พิจารณาข้อมูลดังต่อไปนี้:

1. ก่อนการป้อนโค้ดบัญชีผู้ใช้ในโปรไฟล์ผู้ใช้ให้ตรวจให้แน่ใจว่า โค้ดถูกต้องสำหรับผู้ใช้
2. ควบคุมการเปลี่ยนโค้ดบัญชีกับคำสั่ง Change Job Description (CHGJOB) โดยป้อนเฉพาะสิทธิเจ้าหน้าที่ดูแลความปลอดภัยที่คำสั่ง CHGACGCDE
 - หรือใช้คำสั่ง CHGACGCDE เพื่ออนุญาตให้ผู้ใช้เปลี่ยนโค้ดบัญชีงานของตนเอง หรืองานอื่น ในการเปลี่ยนงานอื่น ผู้ใช้ต้องมีสิทธิพิเศษ *JOBCTL ด้วย
3. ใช้โปรแกรม CL และคำสั่งเพื่อป้องกันการเปลี่ยนโค้ดบัญชีผู้ใช้สำหรับงานบนคิวงาน หรือป้องกันไม่ให้งานหนึ่งเปลี่ยนโค้ดบัญชีผู้ใช้ของงานอื่น ตัวอย่างเช่น ควรให้สิทธิในคำสั่ง CHGACGCDE เป็นส่วนตัว และรวมอยู่ในโปรแกรม CL ที่มีการเปลี่ยนแปลงงานปัจจุบันเท่านั้น (เช่น เมื่อมีการระบุ JOB(*)) คำสั่งจะถูกอนุญาตให้ใช้ที่เหมาะสม

หลักการที่เกี่ยวข้อง

“การรักษาความปลอดภัยและ job accounting” ในหน้า 98

เฉพาะเจ้าหน้าที่ดูแลความปลอดภัย (หรือโปรแกรมที่ได้รับสิทธินี้) หรือผู้ใช้ที่มีสิทธิ *ALLOBJ และ *SECADM เท่านั้นที่สามารถเปลี่ยนค่ากำหนดของระบบ Journal accounting information (QACGLVL) ได้

“เกี่ยวกับ accounting code” ในหน้า 99

Initial accounting code (ยาวได้ถึง 15 อักขระ) ของงานที่กำหนดโดยค่าพารามิเตอร์ ACGCDE (accounting code) ในรายละเอียดของงาน และโปรไฟล์ผู้ใช้สำหรับงาน

การแสดงผลข้อมูลที่รวบรวม

หลังจากรวบรวมข้อมูลในเจอร์นัล job accounting journal คุณต้องบันทึก journal entry ไปที่แสดงไฟล์ และแสดงผล

ถ้าต้องการทำเช่นนี้, ให้ปฏิบัติตามขั้นตอนเหล่านี้:

หมายเหตุ: ในตัวอย่างต่อไปนี้ ชื่อเจอร์นัล job accounting คือ QACGJRN

1. สร้างก๊อปปี้ของ 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) ENTTYP(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 หรือแสดงไปที่ไฟล์โดยการสร้าง และรันเคียวรีโดยใช้คำสั่ง Work with Queries (WRKQRY)

การแปลง job accounting journal entry

คุณสามารถใช้พารามิเตอร์ OUTFILE บนคำสั่ง Display Journal (DSPJRN) เพื่อบันทึก job accounting journal entry ที่ไฟล์ฐานข้อมูลที่คุณสามารถประมวลผลได้

พารามิเตอร์ OUTFILE ช่วยให้คุณสามารถตั้งชื่อไฟล์ หรือเมมเบอร์ได้ ถ้ามีเมมเบอร์อยู่ เมมเบอร์จะถูกลบออกก่อนที่จะบันทึก เร็กคอร์ด ถ้าไม่มีเมมเบอร์อยู่ เมมเบอร์จะถูกใส่เพิ่ม ถ้าไฟล์ไม่มีอยู่ ไฟล์จะถูกสร้างโดยใช้รูปแบบเร็กคอร์ด QJORDJE รูปแบบนี้กำหนดฟิลด์หัวข้อ มาตรฐานสำหรับแต่ละ journal entry แต่ข้อมูล job accounting จะถูกกำหนดเป็นฟิลด์ขนาดใหญ่หนึ่งฟิลด์

เพื่อหลีกเลี่ยงการที่ต้องประมวลผลข้อมูล accounting เป็นฟิลด์ขนาดใหญ่หนึ่งฟิลด์ จึงได้มีการจัดไฟล์อ้างอิงฟิลด์สองไฟล์ เพื่อช่วยคุณในการประมวลผล job accounting journal entry ไฟล์ QSYS/QAJBACG4 มีรูปแบบเร็กคอร์ด QAWTJAJ4 และใช้สำหรับ JB entry ไฟล์ QSYS/QAPTACG5 มีรูปแบบเร็กคอร์ด QSPJAPT5 และใช้สำหรับ DP หรือ SP entry ใช้รูปแบบเดียวกันสำหรับ printer file entry ทั้งหมดโดยไม่คำนึงว่าเอาต์พุตจะเป็น SP (spooled) หรือ DP (nonspooled) DP entry สำหรับไฟล์ที่พิมพ์โดยตรงมีฟิลด์บางฟิลด์ที่ไม่ได้ใช้; ฟิลด์เหล่านี้ มีพื้นที่เปล่า

แนวทางบางอย่างที่คุณอาจจะใช้มีดังนี้:

- JB entry และ DP หรือ SP entry พื้นฐานสามารถประมวลผลได้โดยการสร้างแอดฟุตไฟล์สองไฟล์โดยใช้รูปแบบไฟล์อ้างอิงของฟิลด์ที่นำมา และการรันคำสั่ง DSPJRN หนึ่งครั้งสำหรับ JB และหนึ่งครั้งสำหรับ DP หรือ SP การทำเช่นนี้ช่วยให้คุณสามารถกำหนดโลจิคัลไฟล์ บนฟิลส์คัลไฟล์สองไฟล์ และใช้โปรแกรมภาษาชั้นสูงในการประมวลผล described file ภายนอก
- คุณสามารถประมวลผลเฉพาะ JB entries โดยการสร้างไฟล์โดยใช้ไฟล์อ้างอิงของ ฟิลด์ที่นำมา (QSYS/QAJBACG4) ฟิลส์คัลไฟล์หนึ่งเพื่อสร้าง described file ภายนอกได้ จากนั้นฟิลส์นี้สามารถถูกประมวลผลโดยเคียวรียูทิลิตี้หรือ โปรแกรมภาษาชั้นสูงได้
- คุณสามารถแปลง journal entry ทั้งสองชนิดโดยใช้รูปแบบดีฟอลต์ DSPJRN ของ QJORDJE จากนั้นคุณสามารถใช้ program-described file เพื่อประมวลผล journal entries ในโปรแกรมภาษาชั้นสูง

DDS ต่อไปนี้จะกำหนดฟิลส์คัลไฟล์สำหรับ JB journal entries โดยใช้ไฟล์อ้างอิงของฟิลด์ QAJBACG4 ใน QSYS คุณสามารถสร้างไฟล์ (โดยใช้คำสั่ง Create Physical File (CRTPF)) ที่มีชื่อ (QAJBACG4) เหมือนกันกับชื่อไฟล์ต้นแบบได้

```
R QAWTJAJ4 FORMAT(QSYS/QAJBACG4)
```

DDS ต่อไปนี้จะกำหนดฟิลส์คัลไฟล์สำหรับ DP หรือ SP journal entry โดยใช้ไฟล์อ้างอิงของฟิลด์ QAPTACG5 ใน QSYS คุณสามารถสร้างไฟล์ (โดยใช้คำสั่ง CRTPF) ที่มีชื่อ (QAPTACG5) เหมือนกันกับชื่อไฟล์ต้นแบบได้

```
R QSPJAPT5 FORMAT(QSYS/QAPTACG5)
```

คุณสามารถระบุคีย์ฟิลด์ในฟิลส์คัลไฟล์ ฟิลส์คัลไฟล์หนึ่ง; อย่างไรก็ตามในตัวอย่างนี้ จะใช้โลจิคัลไฟล์สำหรับการจัดลำดับ ถ้าคุณสร้างฟิลส์คัลไฟล์สองไฟล์ (หนึ่งไฟล์สำหรับ JB และหนึ่งไฟล์สำหรับ DP หรือ SP) ด้วยเมมเบอร์ที่มีชื่อเหมือนกัน คุณสามารถใช้คำสั่ง DSPJRN ต่อไปนี้เพื่อแปลง entry ได้ สมมุติว่าคุณได้สร้างฟิลส์คัลไฟล์ 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)
```

คุณสามารถควบคุมการใช้และเกณฑ์การเลือกของคำสั่ง DSPJRN เพื่อให้คุณจะได้ไม่ต้องแปลง entry เดียวกันหลายครั้ง ตัวอย่างเช่น คุณสามารถเลือก entries ทั้งหมดในช่วงของวันที่ระบุ คุณสามารถแปลง entry ทั้งหมดที่ cutoff point สำหรับการวิเคราะห์ job accounting ของคุณ เช่น รายเดือน อาจมีการใช้หนึ่งหรือหลาย journal receiver ในระหว่างเดือน ให้สังเกตว่าการใช้คำสั่ง DSPJRN ที่เมมเบอร์เดียวกันในแต่ละครั้ง ทำให้เมมเบอร์ถูกลบออกก่อนที่จะใส่เพิ่ม entry ใหม่ได้ ห้ามใช้พารามิเตอร์ JOB ของคำสั่ง DSPJRN เนื่องจากบาง entry จะถูกสร้างขึ้นสำหรับงานโดยงานระบบ ดังนั้น จึงไม่ปรากฏตามที่คาดไว้

การอนุญาตให้ประมวลผลฟิลส์คัลไฟล์ทั้งสองแบบ:

ป้อน DDS ดังต่อไปนี้เพื่อสร้างโลจิคัลไฟล์เพื่อให้สามารถ ประมวลผลทั้งสองฟิลส์คัลไฟล์ได้ การทำเช่นนี้ทำให้คุณสามารถอ่านไฟล์เดี่ยวในลำดับ accounting code และพิมพ์รายงานโดยใช้โปรแกรมภาษาชั้นสูง:

```
R QAWTJAJ4 PFILE(YYYY/QAJBACG4)
K JACDE
R QSPJAPT5 PFILE(YYYY/QAPTACG5)
K JACDE
```

การประมวลผลเรีกคอร์ด Job Accounting พื้นฐาน:

ถ้าคุณ ต้องการใช้โลจิคัลไฟล์เพื่อประมวลผลเฉพาะเรีกคอร์ด job accounting พื้นฐาน ในลำดับ accounting code ตามชื่อผู้ใช้ คุณสามารถป้อน DDS ต่อไปนี้ สำหรับโลจิคัลไฟล์:

```
R QAWTJAJ4 PFILE(YYYY/QAJBACG4)
K JACDE
K JAUSER
```

โลจิคัลไฟล์นี้สามารถถูกประมวลผลโดยเคียวรียูทิลิตี้ หรือโดยโปรแกรมภาษาชั้นสูง ถ้าเกิดการจบระบบอย่างผิดปกติ ชื่องาน ที่ถูกต้องใน 30 ไบต์แรกของฟิลด์ JARES ใน journal entry จะอธิบายถึงงานระบบที่บันทึก entry ที่ IPL ถัดไป ไม่ใช่งานที่ใช้รีซอร์ส ด้วยเหตุผลนี้ การวิเคราะห์ที่ใดๆ ที่ทำบน JB entries จึงควรใช้ฟิลด์ JAJOB, JAUSER และ JANBR

การกู้คืน และ job accounting

ถ้างานจบอย่างผิดปกติ accounting entry สุดท้ายจะถูกบันทึก และ accounting entries ที่บันทึกก่อนหน้านี้ทั้งหมดจะปรากฏขึ้น ในเจอร์นัล ถ้าเกิดการจบ ระบบอย่างผิดปกติ accounting data ต่อไปนี้จะหายไป ที่ขั้นตอนการเรตต์ครั้งสุดท้ายหรือ end-of-accounting segment หลังสุด โดยขึ้นอยู่กับว่าสิ่งใดเกิดขึ้นหลังสุด

- ข้อมูลจำนวนของบรรทัดและหน้าที่พิมพ์
- จำนวนของไฟล์ที่สร้าง
- การดำเนินการ put, get และอัปเดตฐานข้อมูล
- การดำเนินการอ่านและบันทึกการสื่อสาร
- การดำเนินการ Auxiliary I/O
- เวลา transaction
- จำนวนของฟิลด์ transaction
- เวลาแอนด์ที่ฟ
- เวลาที่หยุดชั่วคราว

หลังจาการจบระบบอย่างผิดปกติ เวลาในการทำงานให้เสร็จสมบูรณ์ในเจอร์นัลจะไม่เหมือนกับเวลาในข้อความ CPF1164 ข้อความจะใช้เวลาที่ใกล้เคียง กับเวลาของการจบระบบมากที่สุด แต่ job accounting journal entries จะถูกส่งไปที่เจอร์นัลในระหว่าง IPL และเวลาในการทำงานให้สมบูรณ์คือ เวลาในระบบปัจจุบัน ซึ่งช้ากว่าเวลาที่เกิดการจบระบบ อย่างผิดปกติ

ถ้าระบบจบอย่างผิดปกติ journal entries บางอย่างอาจสูญหายได้ entry เหล่านี้ คือ entry ที่ถูกบันทึกไปที่เจอร์นัล แต่ไม่ได้ถูก force ที่ดีสก์ (สิ่งนี้เท่ากับ FORCE(*NO) บนคำสั่ง Send Journal Entry (SNDJRNE)) รวมถึงสิ่งดังต่อไปนี้:

- JB entry ที่เกิดขึ้นโดยคำสั่ง Change Accounting Code (CHGACGCDE)
- DP และ SP entry

เมื่อใดก็ตามที่งานเสร็จสมบูรณ์ accounting code entry หลังสุดจะถูก force ที่ดีสก์ (เหมือนระบุ FORCE(*YES) บนคำสั่ง SNDJRNE) เมื่อใดก็ตามที่ accounting entry ถูก force ที่ดีสก์ entry ก่อนหน้านี้ทั้งหมดใน เจอร์นัล โดยไม่คำนึงถึงงานที่ทำให้เกิด จะถูก force ที่ดีสก์

ข้อยกเว้น

ถ้ามีการระบุเฉพาะ *PRINT accounting บนระบบ จะไม่มีการทำ FORCE(*YES) journal entry การจบงานใดๆ ดังนั้น ถ้า accounting entry ที่สำคัญยังถูกบันทึกโดยคำสั่ง CHGACGCDE คุณต้องตรวจสอบให้แน่ใจว่า entry จะไม่สูญหายไปในการนี้

ของการจบบระบบอย่างผิดปกติ คุณสามารถใช้คำสั่ง SNDJRNE และระบุบูธอพชั่น FORCE(*YES) ถ้าไฟล์จะถูกเจอร์นัลที่ accounting journal ด้วย การเปลี่ยนแปลงฐานข้อมูลใดๆ จะถูก forced ที่เจอร์นัลเสมอ และสิ่งนี้ทำให้ accounting entry ก่อนหน้านี้นี้ทั้งหมดถูก force ด้วย

ถ้าเกิดการจบบระบบอย่างผิดปกติหรือคุณ เปลี่ยน accounting code ของงานที่ไม่ใช่ของคุณเอง ชื่องานที่ถูกต้องใน 30 ไบต์แรก ของฟิลด์ JARES ใน journal entry อธิบายถึง งานระบบที่บันทึก JB entry ที่ IPL ถัดไป ไม่ใช่งานที่ใช้รีซอร์ส ควรใช้ฟิลด์ JAJOB, JAUSER, และฟิลด์ JANBR สำหรับการวิเคราะห์

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 (DSPJRN) เพื่อบันทึก accounting journal entry ที่ไฟล์ฐานข้อมูลที่คุณสามารถประมวลผลได้

คุณยังสามารถใช้คำสั่ง Receive Journal Entry (RCVJRNE) บนเจอร์นัล QACGJRN เพื่อรับ entry ถูกบันทึกไปที่เจอร์นัล QACGJRN ถ้า job accounting journal หรือ journal receivers เสียหาย ระบบจะดำเนินการต่อไปและบันทึก accounting data ในบันทึกประวัติ (ที่ผ่านมา) เพื่อกู้คืนจากความเสียหายของเจอร์นัล หรือ 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 through JABYTE for SP and DP entries

สำหรับโปรแกรมตัวอย่างให้อ้างอิงส่วนในหนังสือคู่มือ CL Programming ที่อธิบายการประมวลผล ไฟล์ QHST สำหรับข้อความความสมบูรณ์ของงาน

ข้อความ CPF1164 ประกอบไปด้วย เร็กคอร์ดสามเร็กคอร์ดเสมอ และข้อความ CPF1303 ประกอบด้วยเร็กคอร์ดสี่ เร็กคอร์ดเสมอ ข้อมูลที่มีอยู่ในฟิลด์ journal prefix มาตรฐานไม่รวม อยู่ในข้อความนี้ ข้อมูลทั้งหมดที่ต้องการคือข้อมูลเกี่ยวกับการจบงาน, วันที่ และเวลา สามารถดูข้อมูลนี้ได้ ใน เร็กคอร์ด 1 ของข้อความ CPF1303

การอ้างอิง

คุณอาจจำเป็นต้องอ้างอิงหัวข้อที่มีประโยชน์เหล่านี้ ในขณะที่ใช้การจัดการระบบงาน

(IBM i5/OS Information Center, Version 6 Release 1 (V6R1) → Systems management → Work management → Reference)

Server job table

คุณสามารถใช้ตารางเซิร์ฟเวอร์นี้เป็นส่วนอ้างอิงในการค้นหาวิธีแก้ปัญหาเซิร์ฟเวอร์, รายละเอียดของงาน และระบบย่อยกับรายการอื่น

ตัวค้นหาค่ากำหนดของระบบ

ใช้ตัวค้นหาค่ากำหนดของระบบในการค้นหาข้อมูลเกี่ยวกับค่ากำหนดของระบบ คุณสามารถค้นหาประเภทของค่ากำหนดของระบบ เมื่อค่าปรากฏใน System i Navigator หรือค้นหาชื่อค่ากำหนดของระบบที่คุณใช้อินเตอร์เฟซแบบอักษร

Work Management APIs

Work management API ดำเนินการฟังก์ชันที่ใช้ในแอปพลิเคชันต่างๆ หน้า Work Management API จะแสดงรายการ API ที่ดึงข้อมูล และจัดการงาน, พูลหน่วยความจำระบบย่อย, คิวงานระบบย่อย, พื้นที่ข้อมูล, เน็ตเวิร์กแอ็ดทริบิวต์, สถานะของระบบ, ค่ากำหนดของระบบ และ flight recorder และรวมถึงรายการของโปรแกรมทางออกการจัดการระบบงานด้วย

ตัวค้นหา IPL SRC

ใช้ตัวค้นหา IPL system reference code (SRC) ในการค้นหาข้อมูลเกี่ยวกับข้อความ SRC ที่จะแสดงบนระบบของคุณ เมื่อคุณทำ IPL SRC บังชี้สถานะ ของ IPL และมีประโยชน์ในการวิเคราะห์ปัญหา คุณสามารถค้นหา SRC โดยใช้ชื่อ หรือแสดงรายการของ SRC ที่ใช้ทั่วไปได้

Group job

ข้อมูลต่อไปนี้เกี่ยวกับงานกลุ่มถูกรวมไว้เป็นเอกสารอ้างอิงสำหรับการดูแลรักษาสถานะแวดล้อมแบบเก่า ในสถานะแวดล้อมยุคคอมพิวเตอร์ในปัจจุบัน เวอร์กสแตชันแบบ single ที่มี session แยกสำหรับฟังก์ชันต่างกันเป็นเรื่องปกติ

งานกลุ่มคล้ายคลึงกับงานโต้ตอบแบบรองที่ร้องขอด้วยการกดปุ่ม System Request อย่างไรก็ตาม สามารถเริ่มต้นงานกลุ่มได้มากถึง 16 กลุ่ม ในแต่ละการ sign on ที่เวิร์กสแตชัน (รวม 32 กลุ่มเมื่อมีงานโต้ตอบแบบรอง) และโปรแกรมแอปพลิเคชันสามารถจัดการอินเทอร์รับต์ได้ง่ายยิ่งขึ้น

ประโยชน์ของงานกลุ่ม

ต่อไปนี้เป็นรายการประโยชน์ของงานกลุ่มบางส่วน

- ผู้ใช้เวิร์กสแตชันสามารถกดปุ่ม Attention เพื่ออินเทอร์รัปต์งานในงานกลุ่มแบบโต้ตอบหนึ่ง, เปลี่ยนเป็นงานกลุ่มแบบโต้ตอบอื่นๆได้ และกลับไปยังงานกลุ่มเดิมได้อย่างรวดเร็ว คีย์ Attention สามารถใช้งานได้ด้วยคำสั่ง Set Attention Program (SETATNPGM) และสามารถใช้อย่างเป็นอิสระในงานกลุ่มต่างๆ
- การใช้งานกลุ่มที่มี display station pass-through เป็นวิธีที่สะดวก รวดเร็วในการเปลี่ยนระหว่างงานแบบโต้ตอบอันหลากหลายบนระบบต่างๆมากมายในเน็ตเวิร์กหนึ่ง

แนวคิดเกี่ยวกับงานกลุ่ม

- งานกลุ่มใช้กับงานแบบโต้ตอบเท่านั้น
- สามารถมีงานกลุ่มได้ถึง 16 กลุ่มในหนึ่งกลุ่ม (มีเพิ่ม 16 กลุ่มเพื่อพร้อมใช้งานได้ ถ้าผู้ใช้ถ่ายโอนไปยังงานโต้ตอบแบบรอง)
- งานกลุ่มมีลักษณะเฉพาะสำหรับผู้ใช้แต่ละคน (งานกลุ่มไม่ได้ถูกแชร์โดยผู้ใช้หลายคน)
- งานกลุ่มเพียงหนึ่งกลุ่มจะแอ็คทีฟในแต่ละครั้ง (งานกลุ่มอื่นจะถูกพักไว้)
- แต่ละงานกลุ่มเป็นอิสระ และมีบันทึกการใช้งาน, ไฟล์ที่สพูลไว้, QTEMP โลบรารี ฯลฯ
- งานกลุ่มหนึ่งถูกเรียกโดยคำสั่ง Transfer to Group Job (TFRGRPJOB) คำสั่งนี้โดยปกติรันจากโปรแกรมเมนู user-written ที่ถูกเรียกใช้งานด้วยการกดคีย์ Attention (คำสั่ง SETATNPGM ต้องถูกรันมาก่อน)
- พื้นที่ข้อมูลกลุ่ม 512 ไบต์สามารถใช้เพื่อส่งผ่านข้อมูลระหว่างกลุ่มหนึ่งไปยังอีกกลุ่มหนึ่ง พื้นที่ข้อมูลกลุ่มนี้ถูกสร้างขึ้นโดยนัยด้วยคำสั่ง Change Group Attributes (CHGGRPA) หนังสือคู่มือการทำโปรแกรม CL มีข้อมูลพื้นที่ข้อมูลกลุ่มต่างๆ

การเปลี่ยนไป และเปลี่ยนจากงานกลุ่ม

ในการเปลี่ยนงานที่ไม่เป็นกลุ่มเป็นงานกลุ่ม และในการเปลี่ยนงานกลุ่มกลับไปเป็นงานที่ไม่เป็นกลุ่ม (หากเป็นเพียงงานเดียวในกลุ่ม) ให้ใช้คำสั่ง Change Group Attributes (CHGGRPA)

การสร้างงานกลุ่มใหม่

ในการสร้างงานกลุ่มใหม่ให้ใช้คำสั่ง Transfer Group Job (TFRGRPJOB)

หมายเหตุ: หลังจากการใช้คำสั่ง TFRGRPJOB แต่ละครั้ง ต้องใช้คำสั่ง SETATNPGM เพื่อเซตเปิดคีย์ Attention หากจำเป็น

การถ่ายโอนจากงานกลุ่มหนึ่งไปยังอีกกลุ่ม

ในการถ่ายโอนจากงานกลุ่มหนึ่งไปยังงานอีกกลุ่มหนึ่งในกลุ่มเดียวกัน ให้ใช้คำสั่ง Transfer Group Job (TFRGRPJOB)

หมายเหตุ:

1. หลังจากการใช้คำสั่ง TFRGRPJOB แต่ละครั้ง ต้องใช้คำสั่ง SETATNPGM เพื่อเซตเปิดคีย์ Attention หากจำเป็น
2. ถ้าคุณอยู่ในการดำเนินการอัปเดต ให้ใช้คำสั่ง Check Record Lock (CHKRCDLCK) เพื่อตรวจสอบว่างานมีการล็อกเรกคอร์ดใดก่อนถ่ายโอนไปยังงานกลุ่มอีกกลุ่มหรือไม่

การควบคุมการถ่ายโอนจากงานกลุ่มหนึ่งไปยังอีกกลุ่ม

คุณสามารถถ่ายโอนการควบคุมจากงานกลุ่มหนึ่งไปยังอีกกลุ่มได้ ถ้าคุณมีโปรแกรม Attention-key handling เมื่อกดคีย์ Attention โปรแกรม Attention-key-handling สามารถเสนอเมนูหนึ่ง (ที่ผู้ใช้สามารถเลือกกลุ่มงานได้) หรือถ่ายโอนผู้ใช้ไปยัง

อีกงานกลุ่มหนึ่งได้อย่างใดอย่างหนึ่งได้ การสนับสนุน Attention-key-handling ทำให้การถ่ายโอนการควบคุมจากงานกลุ่มหนึ่งไปยังอีกกลุ่มง่าย และรวดเร็ว โดยไม่ต้องจบงานหนึ่งเพื่อยังงานอีกงานหนึ่ง

การถ่ายโอนงานกลุ่มหนึ่งโดยไม่ต้องดูเมนู

คุณสามารถใช้คีย์ Attention เพื่อถ่ายโอนโดยตรงไปยังงานอีกงานโดยไม่ต้องดูเมนูได้ ตัวอย่างเช่น โปรแกรม Attention-key-handling สำหรับงานกลุ่ม A สามารถถ่ายโอนไปยังงานกลุ่ม B โปรแกรม Attention-key-handling สำหรับงานกลุ่ม B สามารถถ่ายโอนกลับไปยังงานกลุ่ม A การทำเช่นนั้นอนุญาตให้ใช้การเคาะคีย์หนึ่งครั้งเพื่อเปลี่ยนฟังก์ชันไปมาได้

การจบงานกลุ่ม

- ในการจบกลุ่มงานในกลุ่ม ให้ใช้คำสั่ง End Group Job (ENDGRPJOB)
- ในการจบงานกลุ่มทั้งหมดในกลุ่ม ให้ใช้คำสั่ง SIGNOFF

หมายเหตุ: คำสั่ง ENDJOB สนับสนุนพารามิเตอร์ ADLINTJOBS หาก *GRPJOB ถูกระบุไว้และงานถูกระบุไว้บนพารามิเตอร์ JOB เป็นงานกลุ่ม งานทั้งหมดที่เกี่ยวข้องกับกลุ่มจะจบลง

นอกจากนั้น คำสั่ง End Group Job (ENDGRPJOB) ไม่สนับสนุนสัญญาณ SIGTERM อย่างไรก็ตาม คำสั่ง End Job (ENDJOB) ไม่สนับสนุนสัญญาณ SIGTERM

การตรวจสอบให้แน่ใจว่างานกลุ่มปกติจบ

ในบางสภาวะแวดล้อม อาจจำเป็นต้องให้ผู้ใช้ชั้นปลายจบงานกลุ่มเฉพาะอย่างถูกต้อง แทนที่จะใช้คำสั่ง ENDGRPJOB ตัวอย่างเช่น สมมุติว่าผู้ใช้อาจมีงานกลุ่มที่มีการอัปเดตที่ซับซ้อน และคุณต้องการให้แน่ใจว่างานจบลงแล้วอย่างปกติ อีกตัวอย่างหนึ่งคือ ถ้าผู้ใช้อาจอยู่ระหว่างเซสชัน SEU และควรทำฟังก์ชันเสร็จสิ้นอย่างปกติ

เป็นไปได้ที่การบรรลุดังกล่าวนี้ด้วยการสนับสนุนจากระบบ ตัวอย่างเช่น คุณสามารถใช้ข้อแนะนำต่อไปนี้:

1. เช็ทวิตซ์ในพื้นที่ข้อมูลกลุ่มที่สามารถทดสอบได้โดยแต่ละงานกลุ่มเพื่อทำหน้าที่เหมือนสวิตซ์ปีดระบบ นั่นคือ เมื่อเช็ทเปิดการสลับ ฟังก์ชันกลุ่มงานควรจบลง
2. การเข้าใช้ชื่องานกลุ่มที่แอดทีฟโดยใช้คำสั่ง RTVGRPA และ GRPJOB return variable
3. เปรียบเทียบแต่ละชื่อที่เข้าใช้งาน (เริ่มต้นด้วยงานกลุ่มรอง) กับรายการชื่องานกลุ่มที่กำหนดไว้ก่อนแล้วที่ควรจบลงอย่างถูกต้อง
4. หากชื่องานกลุ่มไม่อยู่ในรายการ สามารถจบงานกลุ่มทันทีโดยคำสั่ง ENDGRPJOB
5. หากต้องจบงานอย่างถูกต้อง ให้ถ่ายโอนงานกลุ่มโดยใช้คำสั่ง TFRGRPJOB

โปรแกรม Attention-key-handling สำหรับทุกกลุ่มต้องไวต่อสวิตซ์ปีดระบบ และป้องกันการถ่ายโอนไปยังอีกงานกลุ่มหนึ่งถ้าเช็ทเปิดสวิตซ์ไว้

ถ้าคุณมีโปรแกรมการควบคุมสำหรับแต่ละงานกลุ่มที่ควบคุมสิ่งที่เกิดเมื่อผู้ใช้จบฟังก์ชันของงานกลุ่ม (ตัวอย่างเช่น โปรแกรมอัปเดต) ยังสามารถทดสอบสวิตซ์ปีดระบบและทำการย้อนกลับได้ ซึ่งถือเป็นการจบงานกลุ่ม และย้อนการควบคุมกลับไปยังงานกลุ่มที่แอดทีฟก่อน

โปรแกรม Attention-key-handling สามารถใช้คำสั่ง CHKRCDLCK เพื่อกำหนดว่าผู้ใช้เวิร์กสเตชันกดคีย์ Attention เมื่อแอ็พพลิเคชันมีเร็คคอร์ดที่ถูกล็อกไว้สำหรับอัปเดตหรือไม่ ในกรณีนี้ โปรแกรม attention อาจส่งข้อความแนะนำผู้ใช้ให้ดำเนินการให้เสร็จสิ้นก่อนการใช้คีย์ Attention

ทฤษฎีงานกลุ่ม

คำสั่ง CHGGRPA ระบุงานปัจจุบันเป็นงานกลุ่ม และให้ชื่องานกลุ่มเพื่อระบุงานในกลุ่มให้แตกต่าง (ที่จุดนี้ กลุ่มมีเพียงหนึ่งงานกลุ่ม) แต่ละงานกลุ่มมีลักษณะเฉพาะสำหรับแต่ละผู้ใช้ ผู้ใช้ต่างๆสองคนไม่ต้องแชร์งานกลุ่มเดียวกัน เมื่อระบุงานหนึ่งเป็นงานกลุ่ม งานนั้นจะสามารถเรียกงานกลุ่มใหม่ได้ นอกจากนี้ ยังมีข้อจำกัดเกี่ยวกับงานกลุ่ม (เช่น RRTJOB, TFRJOB อาจไม่ถูกใช้) เมื่อมีงานแอคทีฟเพียงหนึ่งงานในกลุ่ม งานนั้นอาจกลายเป็นงานที่ไม่เป็นงานกลุ่มได้

การอนุญาตให้งานกลุ่มสื่อสารได้

ในการอนุญาตให้งานกลุ่มสื่อสารระหว่างกัน พื้นที่ข้อมูล 512 ไบต์แบบพิเศษจะเรียกพื้นที่ข้อมูลกลุ่มจะถูกสร้างขึ้นโดยอัตโนมัติเมื่องานกลางเป็นงานกลุ่ม พื้นที่ข้อมูลกลุ่มสามารถเข้าถึงโดยงานในกลุ่ม ด้วยการตั้งค่าพิเศษ *GDA ในพารามิเตอร์ DTAARA ของคำสั่งพื้นที่ข้อมูล

การเรียกงานกลุ่ม

การใช้งานกลุ่มไม่จำเป็นต้องใช้วิธี Attention-key-menu ตามที่อธิบายไว้ในหัวข้อนี้ งานกลุ่มหนึ่งสามารถถูกเรียกจากโปรแกรมแอสัมบลีเคชันใด หรือโดยพารามิเตอร์ GRPJOB(*SELECT) บนคำสั่ง TFRGRPJOB

ฟังก์ชันงานกลุ่ม และคำขอระบบ

ฟังก์ชันงานกลุ่มคล้ายกับฟังก์ชัน System Request ในส่วนที่มีงานที่แอคทีฟงานเดียวเท่านั้นในแต่ละช่วงในขณะที่งานอื่นถูกพักไว้ งานกลุ่มแตกต่างจากคำขอระบบในทางต่อไปนี้ :

- การเริ่มงานกลุ่มไม่จำเป็นต้อง sign on มีการใช้โปรไฟล์ผู้ใช้ และสถานะแวดล้อมเดียวกัน
- งานกลุ่มมากถึง 16 กลุ่มสามารถมีอยู่ได้ในแต่ละครั้ง ผู้ใช้ต้องเลือกงานกลุ่มที่จะถ่ายโอนไป ในขณะที่การใช้คำขอระบบอนุญาตให้ผู้ใช้ถ่ายโอนระหว่างงานสองงานเท่านั้น โดยปกติแล้วในงานกลุ่ม เมนูหนึ่งที่เข้าถึงโดยการกดคีย์ Attention อนุญาตให้ผู้ใช้เลือกงานกลุ่มเพื่อถ่ายโอนได้ การใช้งานกลุ่มร่วมกับคำขอระบบสำหรับรวม 32 กลุ่มพร้อมใช้สำหรับผู้ใช้คนเดียวนั้นเป็นไปได้ อย่างไรก็ตาม งาน 32 กลุ่มอยู่ในกลุ่มแยกกันสองกลุ่ม โดยแต่ละกลุ่มมีพื้นที่ข้อมูลของตนเอง และแอคทีฟิตว์ด์กลุ่มอื่นๆ
- ฟังก์ชันคำขอระบบอนุญาตให้ผู้ใช้เวิร์กสเตชันพักงาน ในขณะที่คีย์บอร์ดถูกล็อก และฟังก์ชันแอสัมบลีเคชันอยู่ระหว่างดำเนินการ ซึ่งสิ่งนี้สามารถอินเทอร์รัปต์ลำดับแบบโลจิคัลของเหตุการณ์ได้ ตัวอย่างเช่น เร็กคอร์ดอาจถูกทิ้งให้ล็อกไว้ในทางตรงกันข้าม คีย์ Attention จะแอคทีฟเมื่อคีย์บอร์ดถูกล็อกสำหรับการอินพุตแล้วเท่านั้น และแอสัมบลีเคชันยังสามารถควบคุมเมื่อคีย์ Attention แอคทีฟ และป้องกันการใช้ในเวลาที่ไม่มีเหมาะสม ฟังก์ชันคำขอระบบพร้อมใช้งานเสมอถ้าผู้ใช้เวิร์กสเตชันมีสิทธิในการใช้ฟังก์ชันนี้

หมายเหตุ: โปรแกรมออกจาก Presystem Request Program ถูกเรียกเมื่อผู้ใช้กดคีย์คำขอระบบ ระบบปฏิบัติการเรียกโปรแกรมออก user-written ผ่านทางเครื่องมือลงทะเบียนเมื่อผู้ใช้กดคีย์คำขอระบบ พารามิเตอร์หนึ่งถูกใช้สำหรับการอินพุต และเอาต์พุต หลังจากโปรแกรมออกจากเครื่องมือลงทะเบียนถูกเรียกใช้ เมนูคำขอระบบจะถูกเรียกตามค่าที่ถูกกลับเข้าไปในแฟล็กแสดงผลเมนูคำขอระบบ สำหรับข้อมูลเพิ่มเติม ให้ดูที่การอ้างอิง API ระบบ

โปรแกรม Attention key handling

คุณสามารถระบุโปรแกรมเป็นโปรแกรม Attention-key-handling ที่ระดับการเรียกพิเศษ โปรแกรม Attention-key-handling รันในงานเดียวกัน และมีแอตทริบิวต์, ค่าแทนที่ และสิทธิในระดับกลุ่มงานเหมือนกันกับโปรแกรมที่ออกคำสั่ง SETATNPGM อย่างไรก็ตาม สิทธิที่ได้รับจากโปรแกรมไม่ได้เกิดจากโปรแกรมที่ถูกอินเทอร์รัปต์ คุณยังอาจจะระบุโปรแกรม Attention-key-handling ในโปรไฟล์ผู้ใช้ได้

การบ่งชี้โปรแกรมเป็น attention key handling

ในการบ่งชี้โปรแกรมเป็นโปรแกรม Attention-key-handling program ให้ใช้คำสั่ง Set Attention Program (SETATNPGM) ที่ระบุ SET(*ON) คำสั่งนี้จะแสดงโปรแกรมนี้ในระดับการเรียกในการรันคำสั่งของงาน ถ้ากดคีย์ Attention key งานที่รันอยู่จะถูกอินเทอร์รัปต์, จอแสดงผลจะได้รับการบันทึก และเรียกโปรแกรม Attention-key-handling program ขึ้นมา ไม่มีพารามิเตอร์ใดที่ถูกส่งผ่านไปยังโปรแกรม Attention-key-handling เมื่อถูกเรียกขึ้นมา

หมายเหตุ: โปรแกรมออกจาก Pre-attention Program จะถูกเรียกขึ้นมา เมื่อผู้ใช้กดปุ่ม System Attention ระบบปฏิบัติการจะเรียกโปรแกรมออกที่เขียนโดยผู้ใช้จากเครื่องมือ registration เมื่อผู้ใช้กดปุ่ม System Attention โดยไม่มีพารามิเตอร์อื่นพุด และเอาต์พุด หลังจากเรียกโปรแกรมออกจากเครื่องมือ registration ขึ้นมา โปรแกรม system attention program จะถูกเรียกด้วย

ผลกระทบของระดับการเรียกบนสถานะคีย์ attention

คำสั่ง SETATNPGM เน้นในการเรียก นั่นคือ คำสั่ง SETATNPGM ที่ออกในระดับการเรียกหนึ่ง ทำให้โปรแกรม Attention-key-handling ได้รับผลกระทบที่ระดับการเรียกปัจจุบัน ตลอดจนระดับการเรียกที่ต่ำกว่าด้วย จนกว่าคำสั่ง SETATNPGM อีกอันหนึ่งจะรันเพื่อเปลี่ยนโปรแกรม Attention-key-handling หรือสถานะคีย์ Attention เมื่อใดก็ตามที่โปรแกรมที่ออกคำสั่ง SETATNPGM ย้อนกลับ จอแสดงผลจะถูกกู้คืน และโปรแกรม Attention-key-handling และสถานะคีย์ Attention จะถูกรีเซ็ตไปยังจุดก่อนการเรียกปัจจุบัน หากคำสั่ง Transfer Control (TRFCTL) ถูกใช้แทนคำสั่ง RETURN สถานะจะไม่ถูกรีเซ็ตจนกว่าโปรแกรมจะถูกโอนให้ย้อนกลับ

ใช้คีย์ attention เมื่อใด

ใช้คีย์ Attention เพื่อเรียกโปรแกรม Attention-key-handling ในการใช้เวิร์สเตชันปกติ คีย์ Attention สามารถกดได้เมื่อปลดล็อกคีย์บอร์ดเท่านั้น นั่นคือ โปรแกรมพร้อมสำหรับอินพุตแล้ว สิ่งนี้เกิดขึ้นเมื่อใช้การทำงานแบบอ่าน หรือบันทึก-อ่าน หรือใช้คีย์เวิร์ด UNLOCK DDS ในการทำการบันทึก

การใช้คีย์ Attention แตกต่างจากการใช้คีย์ System Request ในส่วนที่โปรแกรมแอ็พพลิเคชันเป็นตัวควบคุม ในขณะที่ถูกอินเทอร์รัปต์ได้

ข้อยกเว้น

ข้อยกเว้นในส่วนนี้เกิดขึ้นกับโปรแกรมแอ็พพลิเคชันที่ทำงานแบบไม่ต้องรอบนไฟล์อุปกรณ์ต่างๆ การกดคีย์ Attention ทำให้โปรแกรมเหล่านี้ถูกอินเทอร์รัปต์ที่จุดใดก็ได้โดยโปรแกรม Attention-key-handling (แม้ว่าไฟสัญญาณยับยั้งการอินพุตอาจแสดงขึ้น คีย์บอร์ดจะถูกปลดล็อกกระหว่างการทำงานโดยไม่ต้องรอ) โปรแกรมแอ็พพลิเคชันปฏิบัติหน้าที่ที่ไวต่อการเปลี่ยนแปลง (โดยเฉพาะอย่างยิ่งระหว่างการดำเนินการแบบไม่ต้องรอ) จึงควรถูกปกป้องด้วยการรัน SETATNPGM PGM (*CURRENT) SET(*OFF) ก่อน และรัน SETATNPGM PGM(*CURRENT) SET(*ON) หลังโค้ดที่ไวต่อการเปลี่ยนแปลง

หมายเหตุ: โปรแกรมภาษาระดับสูงสามารถใช้คำสั่ง SETATNPGM โดยการเรียก QCMD EXC

ไม่ควรใช้คีย์ attention เมื่อใด

คีย์ Attention key ไม่สามารถใช้เพื่อเรียกโปรแกรม Attention-key-handling เมื่อมีเงื่อนไขต่อไปนี้:

- คีย์บอร์ดถูกล็อกไว้ (ให้สังเกตข้อยกเว้นที่อธิบายไว้ก่อนหน้านี้เกี่ยวกับการทำงานแบบไม่ต้องรอ)
- เมนู System Request หรือการทำงานของมันในส่วนใดถูกใช้อยู่
- จอภาพข้อความแสดงผลถูกแสดง
- ตัว i5/OS โหลดโปรแกรมกำลังเรียกโปรแกรม Attention-key-handling อยู่ ทำให้โปรแกรมแก็ทที่พอยู แต่หากโปรแกรมออก SETATNPGM อีกอัน ทำให้สามารถใช้งานคีย์ Attention ได้
- เซสชัน BASIC กำลังดำเนินการอยู่ หรือโปรแกรม BASIC ถูกเรียกขึ้นมา

คีย์ Attention และเซสชัน BASIC

ในเซสชัน BASIC คีย์ Attention ถูกจัดการโดย BASIC ตามความเหมาะสม ตัวอย่างเช่น หากโปรแกรม BASIC ถูกเรียกหลังจากคำสั่ง SETATNPGM ได้เปิดคีย์ Attention ขึ้น คีย์ Attention key จะถูกจัดการโดย BASIC หลังจากโปรแกรม BASIC สิ้นสุด โปรแกรม Attention-key-handling จะมีผลบังคับอีกครั้ง

เคล็ดลับการเข้าโค้ดโปรแกรม Attention key handling

ข้อควรระวังเป็นสิ่งจำเป็นในการกำหนดโปรแกรม Attention-key-handling เนื่องจากโปรแกรม Attention-key-handling ทำงานเดียวกันกับโปรแกรมที่ทำงานอยู่เมื่อกดคีย์ Attention ดังนั้น โปรแกรมที่ถูกอินเทอร์รัปต์ไม่ได้รับการปกป้องโดยการล็อกที่มีอยู่ หากโปรแกรมถูกอินเทอร์รัปต์มีการล็อกอ็อบเจกต์เฉพาะ, โปรแกรมคีย์ Attention key program เนื่องจากโปรแกรมทำงานเดียวกัน เป็นส่วนหนึ่งของงานที่มีล็อกเฉพาะ

แนวทางดังต่อไปนี้ถูกแนะนำสำหรับการกำหนดโปรแกรม Attention-key-handling :

- ให้ใช้ฟังก์ชันธรรมดา เช่น เมนู ที่ให้ผู้ใช้เวิร์กสแตชันสามารถโอนไปยังกลุ่มงานอีกกลุ่มหนึ่ง หรือไปยังงานโต้ตอบรอง
- หลีกเลี่ยงการอ้างถึงอ็อบเจกต์ หรือฟังก์ชันที่อาจใช้อยู่ เมื่อกดคีย์ Attention
- หลีกเลี่ยงการเรียกฟังก์ชันแบบ non-recursive เมื่อกดคีย์ Attention ฟังก์ชัน Non-recursive เป็นฟังก์ชันที่ไม่สามารถถูกอินเทอร์รัปต์ได้ แล้วจึงเรียกอีกครั้ง ฟังก์ชันมากมาย เช่น โปรแกรมภาษาระดับสูง และยูทิลิตี้ เช่น DFU เป็น non-recursive
- หลีกเลี่ยงการให้ออปชันที่ช่วยให้ผู้ใช้เวิร์กสแตชันสามารถแสดงจอภาพ command entry เป็นส่วนหนึ่งของงานปัจจุบัน สำหรับผู้ใช้ที่เป็นโปรแกรมเมอร์ การแสดงเมนูที่รวมอ็อปชันสำหรับการแสดง command entry มีความหมาย การแสดงผล command entry ควรระบุเป็นงานกลุ่มต่างหาก (ตัวอย่างเช่น โดยการระบุ INLGRPPGM(QCMD) บนคำสั่ง TFRGRPJOB) ซึ่งการทำเช่นนี้หลีกเลี่ยงการใช้อ็อบเจกต์ที่ใช้อยู่แล้ว
- โปรแกรม Attention-key-handling ไม่มีสิทธิ์ที่ใช้โดยโปรแกรมที่ดำเนินการอยู่ก่อนกดคีย์ Attention key
- โปรแกรม Attention-key-handling ไม่มีพื้นที่ข้อมูลของตนเอง (*LDA) เนื่องจากมีข้อมูลแบบโลคัลข้อมูลเดียวต่องาน และโปรแกรม Attention-key-handling รันในงานเดียวกันกับโปรแกรมที่ถูกอินเทอร์รัปต์ ทั้งสองโปรแกรมจึงแชร์พื้นที่ข้อมูลแบบโลคัลอันเดียวกัน
- โปรดทราบว่าการทำงานของอุปกรณ์ read-from-invited อาจ time-out ได้ระหว่างเวลาที่โปรแกรม Attention-key-handling รันอยู่ ดังนั้น ถ้า time-out ที่ใกล้เสร็จสิ้นในโปรแกรมยังอยู่ระหว่างกระบวนการ ในขณะที่โปรแกรม Attention-key-handling ยังรันอยู่ไม่ว่าจะทำอะไร อันเป็นผลจาก time-out จะเกิดขึ้นระหว่าง return ไปยังโปรแกรม ตัวอย่างเช่น ถ้าเกิดเงื่อนไขต่อไปนี้ โปรแกรมจะออกจากการ return จาก Attention key handler:

- ค่า WAITRCD ในไฟล์ตั้งไว้ที่ 60 วินาที
- โปรแกรมถูกตั้งให้ออก ถ้าไม่กดปุ่มภายในหนึ่งนาที
- โปรแกรมคีย์ Attention จะถูกเรียก และรันนานกว่านาทีนั้น

แต่ควรปฏิบัติตามข้อควรระวัง เนื่องจากจะมีการตรวจสอบข้อมูลพร้อมใช้งานก่อนการตรวจสอบว่า time-out เสร็จสมบูรณ์แล้ว หากกดปุ่มทันทีหลังจากออกจาก Attention key handler ข้อมูลอาจพร้อมใช้งานอยู่ ซึ่งทำให้อุปกรณ์ read-from-invited สมบูรณ์ และ time-out ไม่รับการตรวจสอบ ลักษณะดังกล่าวนี้ทำให้เกิดผลที่ไม่คาดคิดขึ้นได้

เคล็ดลับประสิทธิภาพงานกลุ่ม

หัวข้อนี้ให้เคล็ดลับบางประการในการรักษาประสิทธิภาพของระบบที่ดีที่สุดเมื่อใช้งานกลุ่ม

- ผลกระทบบนระบบสำหรับงานที่ค้างจำนวนมากโดยปกติมีเพียงเล็กน้อย หากข้อกำหนดเกี่ยวกับแหล่งเก็บข้อมูลหลักที่ใช้ไม่เป็นปัจจัยร่วม
- เมื่อคำสั่ง TFRGRPJOB รันอยู่ และงานใหม่ต้องเริ่มต้น โอเวอร์เฮดที่เกี่ยวข้องจะประมาณเดียวกันกับการ sign on เข้าระบบ เมื่อคำสั่งรันอยู่ และงานกลุ่มเริ่มต้นแล้ว โอเวอร์เฮดที่จำเป็นจะประมาณเดียวกับที่ใช้การโอนถ่ายไปยังอ็อปชันงานรองบนเมนู System Request เมื่องานรองแอนด์ที่พอย์อยู่แล้ว
- หากงานกลุ่มต้องรันด้วยความถี่ใดๆ ควรป้องกันงานกลุ่มไม่ให้จบลง นั่นคือ ห้ามจบโปรแกรม แต่ใช้คำสั่ง TFRGRPJOB เพื่อป้องกันงานเริ่มทุกครั้งที่ต้องใช้ฟังก์ชันงานกลุ่ม
- คำสั่ง SETATNPGM ทำให้จอแสดงผลปัจจุบันถูกบันทึก เมื่อคีย์ Attention และถูกกู้คืนเมื่อจบโปรแกรม Attention key-handling ซึ่งคล้ายคลึงกับการใช้เมนู System Request และมีผลกระทบอย่างชัดเจนบนเวิร์กสเตชันแบบรีโมต
- การควบคุมจำนวนงานที่แอนด์ที่พบนระบบ (พารามิเตอร์ MAXJOBS บนคำสั่ง CRTSBSD) ไม่ได้รับผลกระทบจากงานกลุ่มที่แอนด์ที่พในเวลาใด
- ค่าระบบทั้งหมดที่ควบคุมการสร้างโครงสร้างงาน (QACTJOB และ QADLACTJ และ QTOTJOB และ QADLTOTJ) ได้รับผลกระทบ ค่าเหล่านี้อาจจำเป็นต้องเพิ่มขึ้นเพื่ออนุญาตให้ใส่เพิ่มงานกลุ่มได้

การแก้ปัญหาสำหรับการจัดการระบบงาน

หัวข้อนี้ช่วยคุณแก้ปัญหาทั่วไปบางอย่างที่เกิดขึ้นในการจัดการระบบงาน

งานของฉันทหยุดทำงาน

ตารางเหล่านี้แสดงสาเหตุที่เป็นไปได้ที่ทำให้งานหยุดทำงาน

งานกำลังรอที่จะล็อกบนอ็อบเจ็กต์	
วิธีการวินิจฉัย:	ให้ดูสถานะของงานใน System i Navigator; ดู Determining the status of a job. งานที่กำลังรอการล็อกจะมีสถานะเป็น <i>กำลังรอการล็อก</i> .
วิธีแก้ไข:	ดูรายการของอ็อบเจ็กต์ที่ล็อกสำหรับงาน เพื่อกำหนดอ็อบเจ็กต์ที่งานกำลังรอที่จะล็อก แล้วใช้การดำเนินการแบบ Lock Holders กับอ็อบเจ็กต์เพื่อระบุว่างานใดที่กำลังล็อกอ็อบเจ็กต์นั้นอยู่ จากนั้นคุณต้องกำหนดว่าทำไมงานดังกล่าวจึงต้องถูกล็อกเอาไว้ และจะทำเช่นไรเพื่อปลดล็อกมันออกให้ได้

งานถูกพัก		
	วิธีการวินิจฉัย:	ให้ดูสถานะของงานใน System i Navigator; ดู Determining the status of a job.
	วิธีแก้ไข:	คลิกขวาที่งานและคลิก Release .

ต่อไปนี้เป็นสาเหตุที่พอจะเป็นไปได้ที่งานในคิวงานอาจหยุดการทำงาน :

คิวงานถูกพัก		
	วิธีการวินิจฉัย:	ดูสถานะของคิวงานใน System i Navigator;
	วิธีแก้ไข:	<ol style="list-style-type: none"> ย้ายงานไปไว้ในคิวงานที่ไม่ได้ถูกพักไว้ให้ดู การย้ายงานไปยังคิวงานอื่น. รีเซ็ตคิวงาน หากต้องการทำเช่นนั้น ให้คลิกขวาที่งานและคลิก Release.

คิวงานไม่ได้ถูกจัดสรรไว้ให้โดยระบบย่อยแอ็คทีฟ		
	วิธีการวินิจฉัย:	ดูสถานะของคิวงานใน System i Navigator.
	วิธีแก้ไข:	<ol style="list-style-type: none"> ย้ายงานไปยังคิวงานที่ได้รับการจัดสรรไว้โดยระบบย่อยที่แอ็คทีฟให้ดู การย้ายงานไปยังคิวงานอื่น . สตาร์ทระบบย่อยที่มี job queue entry สำหรับคิวงานนี้ให้ดูที่ การเริ่มระบบย่อย. ใส่ job queue entry ของคิวงานดังกล่าวลงไปในระบบย่อยแอ็คทีฟ โดยใช้คำสั่ง Add Job Queue Entry (ADDJOBQE).

ครบจำนวนสูงสุดของระบบย่อย		
	วิธีการวินิจฉัย:	ดูค่างานแอ็คทีฟสูงสุดสำหรับระบบย่อยใน System i Navigator. หากต้องการทำเช่นนั้น ให้คลิกขวาที่ระบบย่อยและคลิก Properties .
	วิธีแก้ไข:	<ol style="list-style-type: none"> ย้ายงานไปไว้ในคิวงานอื่น ให้ดู การย้ายงานไปยังคิวงานอื่น. เพิ่มค่าสูงสุด หากต้องการทำเช่นนั้น ให้ใช้คำสั่ง Change Subsystem Description (CHGSBSD)

ครบจำนวนสูงสุดของคิวงาน		
	วิธีการวินิจฉัย:	ดูค่างานแอ็คทีฟสูงสุดสำหรับคิวงานใน System i Navigator. หากต้องการทำเช่นนั้น ให้คลิกขวาที่คิวงานและคลิก Properties . แล้วจึงเลือกแท็บ Activity .
	วิธีแก้ไข:	<ol style="list-style-type: none"> ย้ายงานไปไว้ในคิวงานอื่น ให้ดู การย้ายงานไปยังคิวงานอื่น. เพิ่มค่าสูงสุด หากต้องการทำเช่นนั้น ให้ใช้คำสั่ง Change Job Queue Entry (CHGJOBQE)

ครบค่าสูงสุดสำหรับระดับความสำคัญ		
	วิธีการวินิจฉัย:	ระบุระดับความสำคัญของคิวงาน ของงานโดยการดูที่คุณสมบัติของงานนั้น จากนั้น ดูงานแอคทีฟสูงสุดด้วยค่าระดับความสำคัญงานสำหรับคิวงานใน System i Navigator. หากต้องการทำเช่นนั้น ให้คลิกขวาที่คิวงานและคลิก Properties . จากนั้นเลือกแท็บ Activity และคลิกปุ่ม Advanced .
	วิธีแก้ไข:	<ol style="list-style-type: none"> ย้ายงานไปไว้ในคิวงานอื่น ให้ดู การย้ายงานไปยังคิวงานอื่น . เปลี่ยนระดับความสำคัญของคิวงานของงาน ให้ดู การกำหนดระดับความสำคัญสำหรับคิวงาน . เพิ่มค่าสูงสุด หากต้องการทำเช่นนั้น ให้ใช้คำสั่ง Change Job Queue Entry (CHGJOBQE)

งานของฉันประสบปัญหาทำงานช้า

ต่อไปนี้เป็นสาเหตุที่เป็นไปได้ว่า เพราะเหตุใดงานจึงมีประสิทธิภาพไม่ดี

หน่วยความจำไม่เพียงพอ		
	วิธีการวินิจฉัย:	ให้ดูที่คุณสมบัติของงานเพื่อระบุว่างานกำลังรันอยู่ในพูลหน่วยความจำ อันใด จากนั้น ดูคุณสมบัติของพูลหน่วยความจำใน System i Navigator โดยดูที่ Checking memory pool usage . อัตราความผิดพลาดสูงในพูลบอกได้ว่ามีหน่วยความจำในพูลที่ไม่เพียงพอ หรือในพูลมีงานเป็นจำนวนมากเกินไปที่แย่งใช้หน่วยความจำ กันอยู่
	วิธีแก้ไข:	<ol style="list-style-type: none"> ให้เปิดตัวปรับแต่งระบบ ในกรณีที่คุณไม่ได้ใช้งานอยู่ให้ดู ค่ากำหนดของระบบประสิทธิภาพ: ปรับพูลหน่วยความจำ และ activity levels โดยอัตโนมัติ สำหรับข้อมูลเกี่ยวกับการปรับพูลหน่วยความจำ และ activity level โดยอัตโนมัติ ถ้าเป็นไปได้ ให้ปรับแต่งพูลด้วยตัวคุณเองโดยให้เพิ่มจำนวนหน่วย ความจำในพูล หรือลด activity level ของพูลหน่วยความจำลง คุณอาจต้องการตรวจสอบพูลเครื่องเพื่อตรวจสอบขนาดของหน่วยความจำที่ใช้งานว่าไม่มีผลต่องานทั้งหมดบนระบบ

Activity level ต่ำเกินไป		
	วิธีการวินิจฉัย:	ดูที่คุณสมบัติของงานเพื่อระบุ สถานะและดูว่างานกำลังรันอยู่ในพูลหน่วยความจำอันใด ถ้างานแสดงสถานะเป็น <i>Waiting for activity level</i> ให้ดูคุณสมบัติของพูลหน่วยความจำใน System i Navigator โดยดูที่ Checking memory pool usage . อัตราการเปลี่ยนไปสู่สถานะการไม่มีสิทธิในพูลที่มีสูงนั้นเป็นการชี้ให้เห็นว่าในพูลมีจำนวนงานมากเกินไปซึ่งแย่งกันใช้หน่วยความจำกันอยู่
	วิธีแก้ไข:	<ol style="list-style-type: none"> ให้เปิดตัวปรับแต่งระบบ ในกรณีที่คุณไม่ได้ใช้งานอยู่ให้ดู ค่ากำหนดของระบบประสิทธิภาพ: ปรับพูลหน่วยความจำ และ activity level โดยอัตโนมัติ สำหรับข้อมูลเกี่ยวกับการปรับพูลหน่วยความจำ และ activity level ให้ปรับพูลด้วยตนเองโดยการเพิ่ม activity level ของพูลหน่วยความจำ

รีซอร์ส CPU ไม่เพียงพอ	
วิธีการวินิจฉัย:	ดูคอลัมน์ CPU % สำหรับงาน และงานอื่นในรายการ Active Job ของ System i Navigator. หากระบบยุ่งมากงานของคุณอาจได้รับรีซอร์ส CPU ไม่เพียงพอที่จะทำงานให้เสร็จสิ้นได้
วิธีแก้ไข:	<ol style="list-style-type: none"> 1. ถ้าเป็นไปได้ ให้จบหรือพักงานที่ไม่จำเป็นบนระบบ 2. หากงานจำนวนหนึ่งคือการทำงานที่เน้นใช้ CPU มาก ให้เปลี่ยนระดับความสำคัญในการรันงานเหล่านี้ (ค่าระดับความสำคัญการรันที่สูงขึ้นเท่ากับระดับความสำคัญการรันที่ต่ำลงของงาน)

อ็อปชันการเพจพูลหน่วยความจำ	
วิธีการวินิจฉัย:	หากแอ็พพลิเคชันคือการทำงานที่เน้นใช้ดิสก์ หาก CPU ทำงานอย่างไม่เต็มประสิทธิภาพและมีหน่วยความจำไม่เพียงพอ การใช้ expert cache จะมีประโยชน์มาก
วิธีแก้ไข:	Expert cache สามารถเปิดใน System i Navigator โดยการเปลี่ยนอ็อปชัน Paging สำหรับ shared memory pool เป็น Calculated อ็อปชัน Paging อยู่ที่แท็บ Configuration ของหน้า Properties ของพูลหน่วยความจำและจะมีอยู่บนพูลแบบแบ่งใช้เท่านั้น (ไม่ใช่พูลส่วนตัว)

ระดับความสำคัญในการรันของงานต่ำ	
วิธีการวินิจฉัย:	ในการกำหนดระดับความสำคัญในการรันของงานที่สัมพันธ์กับงานอื่นๆในระบบ ให้ดูที่ Viewing job attributes.
วิธีแก้ไข:	ถ้าหากงานมีระดับความสำคัญในการรันที่ต่ำ (ค่าสูง) โดยเปรียบเทียบกับงานอื่นๆ และไม่ได้ใช้ CPU มากนัก เพราะงานที่ระดับความสำคัญที่สูง (ค่าต่ำ) จะใช้ CPU รีซอร์สส่วนใหญ่อยู่แล้ว คุณอาจจำเป็นต้องเพิ่มระดับความสำคัญของการรันให้มากขึ้น ให้ดูที่ Viewing job attributes. และ บนระบบที่มีการใช้งาน CPU สูง และงานมีระดับความสำคัญการรันต่ำ ให้ดูที่ ค่ากำหนดของระบบประสิทธิภาพ : ปรับระดับความสำคัญองงานภายในช่วงระดับความสำคัญ และ ค่ากำหนดของระบบประสิทธิภาพ : ปรับระดับความสำคัญองงานแบบโต้ตอบ. ค่ากำหนดของระบบอาจมีประโยชน์

สำหรับข้อมูลเพิ่มเติมเกี่ยวกับประสิทธิภาพ ให้ดูที่ Performance. หากคุณต้องการข้อมูลเพิ่มเติมเกี่ยวกับวิธีการปรับประสิทธิภาพบนระบบของคุณให้ดูที่ Tuning performance.

การสืบสวนงานแบบ prestart

หัวข้อนี้แนะนำเสนอขั้นตอนที่ช่วยให้คุณในการตอบคำถาม "ฉันจะ ค้นหาผู้ใช้จริงของงานแบบ prestart และกำหนดรีซอร์สที่ใช้โดยงานแบบ prestart นั้นได้อย่างไร?"

System i Navigator

คุณสามารถใช้ System i Navigator การดู work management และการมอนิเตอร์ Management Central เพื่อทำการวิเคราะห์ที่แบบเรียลไทม์ถึงสิ่งที่เกิดขึ้นบนระบบของคุณ

1. ใช้มุมมอง Server Jobs เพื่อดูงานเซิร์ฟเวอร์ที่แอ็คทีฟและผู้ใช้ ปัจจุบัน (My Connections → connection → Work Management → Server Jobs)
 - คลิกขวาที่ Server Jobs container แล้วเลือก **Customize this view** → **Columns** และตรวจให้แน่ใจว่า Current User, Total CPU Time และ Total CPU DB Time อยู่ใน รายการ **Columns to be displayed**

- ถ้ารายการงานเซิร์ฟเวอร์ที่แฉีกที่ฟของคุณมีขนาดใหญ่, คุณสามารถจำกัดสิ่งที่ จะแสดงโดยใช้ชื่องาน, หมายเลขงาน, ผู้ใช้ปัจจุบัน หรือสถานะ คลิกขวาที่ Server Jobs container และคลิก **Customize this view** → **Include** .
- คุณสามารถเรียงลำดับการแสดงผลของรายการงานเซิร์ฟเวอร์ที่แฉีกที่ฟได้โดยการ คลิกบนหัวข้อคอลัมน์
- คุณสามารถเซตจอสแสดงผลให้รีเฟรชโดยอัตโนมัติในช่วงเวลาที่สม่ำเสมอได้ (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 เพื่อตั้งค่าการมอนิเตอร์ระบบที่มอนิเตอร์ CPU Utilization โดยรวม (ขยาย **Management Central** → **Monitors** ให้คลิกขวา **System**, และเลือก **New Monitor**.)
 - ในขณะที่การมอนิเตอร์รันอยู่ คุณสามารถคลิกจุดใดจุดหนึ่งเพื่อดูรายละเอียด ระดับถัดไปได้ ตัวอย่างเช่น เมื่อมอนิเตอร์ CPU Utilization คุณสามารถแสดงรายการของงานที่มี CPU utilization สูงสุดได้ จากนั้นคุณสามารถ คลิกขวาที่งานที่กำลังใช้ CPU เป็นจำนวนมาก และคลิก **Properties** เพื่อแสดงคุณสมบัติของงาน (ให้ดูคำอธิบายทางออนไลน์ สำหรับรายละเอียดเพิ่มเติมเกี่ยวกับวิธีการใช้การมอนิเตอร์ระบบ)
3. ใช้ Management Central เพื่อตั้งค่าการมอนิเตอร์งานเพื่อมอนิเตอร์งาน เซิร์ฟเวอร์ที่เหมาะสม และแจ้งให้คุณทราบเมื่องานเหล่านั้นเริ่มใช้รีซอร์สมากเกินไป (ขยาย **Management Central** → **Monitors** ให้คลิกขวา **Job** และเลือก **New Monitor**.)

อินเทอร์เฟซแบบอักษร

คำสั่ง: Work with Active Job (WRKACTJOB)

คำสั่งนี้จะแสดงผู้ใช้ปัจจุบัน ของ initial thread (ซึ่งคืองานเมื่องานถูก single threaded) นี้เป็นข้อมูลเดียวกับที่แสดงอยู่ใน GUI หลักการที่เกี่ยวข้อง

“Prestart job entry” ในหน้า 54

คุณกำหนดงานแบบ prestart โดยใช้ prestart job entry prestart job entry ไม่มีผลต่อการจัดสรรอุปกรณ์หรือการกำหนดค่าของการเริ่มโปรแกรม

“งานแบบ prestart สำหรับเซิร์ฟเวอร์” ในหน้า 16

ในต้นแบบงานแบบ prestart จะมีงานการฟังหลักหนึ่งงาน โดยทั่วไป เรียกว่างาน daemon หรืองาน listener และหลายงานเซิร์ฟเวอร์ที่ประมวลผล คำขอไคลเอ็นต์งาน daemon จะรอฟังคำขอการเชื่อมต่ออยู่บนพอร์ต เมื่อได้รับการเชื่อมต่อใหม่ daemon จะทำงานทั่วไป แล้วให้ socket descriptor ที่งานเซิร์ฟเวอร์แบบ prestart ที่กำลังรออยู่

งานที่เกี่ยวข้อง

“การใส่เพิ่ม prestart job entries” ในหน้า 183

Prestart job entries ระบุงานแบบ prestart ที่สามารถเริ่มต้น เมื่อระบบย่อยเริ่มต้น หรือเมื่อป้อนคำสั่ง Start Prestart Jobs (STRPJ) คุณสามารถใส่เพิ่ม prestart job entries ที่ subsystem description โดยใช้อินเทอร์เฟซแบบอักษร

“การเปลี่ยน prestart entry” ในหน้า 188

คุณสามารถเปลี่ยน prestart job entry ใน subsystem description ที่ระบุได้ ระบบย่อยอาจแฉีกที่ฟอยู่เมื่อ prestart job entry ถูก เปลี่ยน การทำการเปลี่ยนแปลงที่ entry เมื่อระบบย่อยแฉีกที่ฟจะมีผลต่อการดำเนิน การในอนาคต งานแบบ prestart ใหม่ใดๆ ที่เริ่มขึ้นหลังจากใช้คำสั่ง จะใช้ ค่าที่เกี่ยวข้องกับงานใหม่ คำสั่งนี้ระบุงานแบบ prestart ที่เริ่มขึ้นเมื่อระบบย่อยเริ่มต้น หรือเมื่อมีการใช้คำสั่ง Start Prestart Jobs (STRPJ)

“การลบ prestart job entry” ในหน้า 192

คุณสามารถลบ prestart job entries ออกจาก subsystem description โดยใช้อินเตอร์เฟซแบบอักษร จะไม่สามารถลบ prestart job entry ถ้างานแอ็คทีฟในปัจจุบันใดๆ ถูกเริ่มโดยใช้ entry

ข้อมูลที่เกี่ยวข้องกับการจัดการระบบงาน

กลุ่มหัวข้อ information center อื่นๆ มีข้อมูลที่เกี่ยวข้องกับกลุ่มหัวข้อ Work Management

รายงานจากประสบการณ์

บันทึกจากประสบการณ์การจัดการระบบงานช่วยให้คุณเห็นถึงวิธีปฏิบัติในโลกที่แท้จริงในการใช้เครื่องมือการจัดการระบบงานในงานประจำวันของคุณ

Networking

ความเข้าใจถึงเทคโนโลยีการทำเน็ตเวิร์กของคุณนับเป็นส่วนที่สำคัญยิ่ง ของโซลูชันอ็อบเจกต์ทั้งหมดของบริษัทของคุณ เรียนรู้วิธีการเชื่อมต่อธุรกิจของคุณกับ อินเทอร์เน็ต, ตั้งค่าอีเมล และให้บริการอ็อบเจกต์มัลติมีเดียแก่ Web browser clients คุณสามารถรวมไฟล์และการให้บริการพิมพ์, การจัดการโปรไฟล์ผู้ใช้ และการดำเนินการเน็ตเวิร์ก ค้นหาข้อมูลเกี่ยวกับเซิร์ฟเวอร์ Windows ที่สามารถถูกรวมเข้าไปในเซิร์ฟเวอร์ และอ่านเกี่ยวกับการรักษาความปลอดภัยซึ่งสามารถช่วยปกป้องรีซอร์สของคุณได้

Retrieve Network Attributes (QWCRNETA) API

Retrieve Network Attributes (QWCRNETA) API ช่วยคุณดึงข้อมูล เน็ตเวิร์กแอ็ททริบิวต์

Retrieve IPL Attributes (QWCRIPLA) API

Retrieve IPL Attributes (QWCRIPLA) API ย้อนการตั้งค่าแอ็ททริบิวต์ที่ใช้ระหว่างIPL API นี้ช่วยสนับสนุนแบบ คล้ายคลึงกันนี้ให้คำสั่ง Display IPL Attributes (DSPIPLA)

ประสิทธิภาพ

การทำความเข้าใจกระบวนการต่างๆ ทั้งหมดที่มีผลต่อประสิทธิภาพของระบบ นับเป็นสิ่งที่ท้าทายสำหรับผู้ใช้ที่ไม่มีประสบการณ์ การแก้ไขปัญหาเกี่ยวกับประสิทธิภาพ ต้องใช้ชุดของเครื่องหมายจำนวนมากอย่างมีประสิทธิภาพ แต่ละเครื่องมือมีเซต ของข้อกำหนดและฟังก์ชันที่ได้รับการสนับสนุนไม่ซ้ำกัน แม้ว่า你能รวบรวมและวิเคราะห์ข้อมูล ประสิทธิภาพแล้ว แต่อาจไม่ทราบว่าจัดการกับข้อมูลเหล่านั้น อย่างไร หัวข้อนี้จะช่วยให้คุณทราบถึงภารกิจและ เครื่องมือที่เกี่ยวข้องกับ การจัดการประสิทธิภาพ

Performance explorer

Performance explorer รวบรวมข้อมูลรายละเอียดเกี่ยวกับแอ็พพลิเคชัน, โปรแกรม หรือรีซอร์สระบบเฉพาะ และให้ ข้อมูลเชิงลึกรายละเอียดเกี่ยวกับปัญหาประสิทธิภาพเฉพาะ สิ่งนี้รวมความสามารถทั้งเพื่อดำเนินประเภท และระดับ การติดตามหลายแบบ และเพื่อรันรายงานในรายละเอียด

การบริหารเวลา

ภายในส่วนประกอบของการบริหารเวลาของ System i Navigator คุณสามารถทำงานกับเขตเวลา และฟังก์ชันการ ปรับเปลี่ยนเวลาได้ ด้วยฟังก์ชันเหล่านี้ คุณสามารถเลือกเขตเวลาสำหรับระบบของคุณเพื่อใช้ และปรับเวลาของ ระบบได้

ค่ากำหนดของระบบ

ค่ากำหนดของระบบ คือ ชั้นของข้อมูลที่มีผลกระทบต่อสถานะแวดล้อมการปฏิบัติงานของระบบ ค่ากำหนดของระบบ ไม่ใช่อ็อบเจกต์ บนระบบ แต่ค่ากำหนดของระบบมีข้อมูลการควบคุมสำหรับการดำเนินการ ของส่วนบางส่วนของ ระบบ

ภาคผนวก. คำประกาศ

ข้อมูลนี้ถูกพัฒนาขึ้นสำหรับผลิตภัณฑ์และบริการที่เสนอขายในประเทศไทย

IBM อาจไม่สามารถจัดเตรียมผลิตภัณฑ์ บริการ หรือคุณลักษณะพิเศษที่กล่าวถึงในเอกสารนี้ในประเทศอื่นๆ ได้โปรดปรึกษาตัวแทนของ IBM สำหรับข้อมูลเกี่ยวกับผลิตภัณฑ์และบริการที่เสนอขายอยู่ในท้องที่ของคุณ การอ้างอิงเกี่ยวกับผลิตภัณฑ์ โปรแกรม หรือบริการของ IBM มิได้มีเจตนาบอกกล่าว หรือแสดงนัยยะว่าเฉพาะผลิตภัณฑ์ โปรแกรม หรือบริการของ IBM เท่านั้นที่สามารถใช้ได้ ผลิตภัณฑ์ โปรแกรม หรือบริการที่ทำงานได้เท่าเทียมกัน ซึ่งไม่ละเมิดทรัพย์สินทางปัญญาของ IBM อาจสามารถใช้แทนกันได้ อย่างไรก็ตาม เป็นความรับผิดชอบของผู้ใช้ที่จะประเมินผล และตรวจสอบการทำงานของผลิตภัณฑ์ โปรแกรม หรือบริการที่ไม่ใช่ของ IBM

IBM อาจมีสิทธิบัตรหรือเอกสารซึ่งอยู่ระหว่างการดำเนินการขอสิทธิบัตรที่ครอบคลุมถึงประเด็นที่อธิบายไว้ในเอกสารนี้ การตกแต่งเอกสารนี้ใหม่ไม่ได้ทำให้คุณได้รับการอนุญาตจากสิทธิบัตรเหล่านั้น คุณสามารถสอบถามเกี่ยวกับการอนุญาตใช้สิทธิได้โดยเขียนส่งไปที่ :

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

สำหรับการสอบถามการอนุญาตใช้สิทธิ์เกี่ยวกับข้อมูล double-byte (DBCS) ให้ติดต่อ IBM Intellectual Property Department ในประเทศของคุณ หรือส่งคำถามไปที่ :

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

ย่อหน้าต่อไปนี้ไม่มีผลบังคับใช้กับสหราชอาณาจักร หรือประเทศอื่นๆ ที่ซึ่งบริการและผลิตภัณฑ์ไม่สอดคล้องกับกฎหมายท้องถิ่น: INTERNATIONAL BUSINESS MACHINES CORPORATION จัดเตรียมข้อมูลนี้ “ตามสภาพที่เป็น” โดยไม่มีการรับประกันใดๆ ทั้งโดยชัดแจ้งหรือโดยนัย ตลอดจนไม่มีการรับประกันโดยนัยต่อความสามารถในการจัดจำหน่าย การไม่ละเมิด หรือความเหมาะสมสำหรับวัตถุประสงค์อย่างใดอย่างหนึ่ง ทั้งนี้ในบางรัฐไม่อนุญาตให้มีการจำกัดความรับผิดชอบในการรับประกันโดยชัดแจ้ง หรือโดยนัยในการทำธุรกรรมบางอย่าง ดังนั้น ข้อความข้างต้นอาจใช้ไม่ได้กับคุณ

ข้อมูลนี้อาจมีความไม่ถูกต้องทางเทคนิคหรือความผิดพลาดทางการพิมพ์ การเปลี่ยนแปลงข้อมูลในนี้จะมีเป็นระยะๆ ซึ่งจะสอดคล้องกับการตีพิมพ์ในครั้งใหม่ IBM อาจทำการปรับปรุง และ/หรือเปลี่ยนแปลงในผลิตภัณฑ์ และ/หรือโปรแกรมที่ได้ อธิบายไว้ในเอกสารนี้ได้ตลอดเวลาโดยไม่ต้องแจ้งให้ทราบล่วงหน้า

การอ้างอิงเว็บไซต์ที่ไม่ใช่ของ IBM นั้นเป็นไปเพื่อวัตถุประสงค์ด้านความสะดวกเท่านั้น และไม่ได้เป็นการรับรองเว็บไซต์เหล่านั้น เนื้อหาที่อยู่ในเว็บไซต์เหล่านั้น ไม่ถือว่าเป็นส่วนหนึ่งของเนื้อหาสำหรับผลิตภัณฑ์ของ IBM นี้ และคุณต้องรับผิดชอบต่อความเสี่ยงในการใช้งานเว็บไซต์ดังกล่าว

IBM อาจใช้งานหรือเผยแพร่ส่วนหนึ่งส่วนใดของข้อมูลที่คุณให้ไว้ไม่ว่าจะด้วยวิธีการใดเมื่อเห็นสมควร โดยไม่ก่อให้เกิดข้อผูกพันใดๆ แก่คุณ

หากผู้ที่ได้รับอนุญาตให้ใช้โปรแกรมนี้ต้องการทราบข้อมูลเกี่ยวกับโปรแกรมเพื่อจุดประสงค์ในการใช้งาน: (1) แลกเปลี่ยนข้อมูลระหว่างโปรแกรมที่ถูกสร้างขึ้นอย่างเป็นอิสระ และโปรแกรมอื่น (รวมถึงโปรแกรมนี้) และ (2) ใช้ข้อมูลร่วมกันซึ่งมีการแลกเปลี่ยน ควรติดต่อ:

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

ข้อมูลดังกล่าวอาจมีพร้อมให้ภายใต้ข้อกำหนดและเงื่อนไขที่เหมาะสม รวมถึงในบางกรณี ที่ต้องมีการชำระค่าธรรมเนียม

โปรแกรมไลเซนส์ที่อธิบายไว้ในเอกสารนี้ รวมถึงข้อมูลทั้งหมดที่มีสำหรับโปรแกรม ถูกจัดเตรียมโดย IBM ภายใต้เงื่อนไขของ IBM Customer Agreement, IBM International Program License Agreement, IBM License Agreement for Machine Code หรือข้อตกลงอื่นที่เทียบเท่า

ข้อมูลของประสิทธิภาพการทำงานใดในที่นี่ เป็นข้อมูลที่ได้จากสภาพแวดล้อมที่ถูกควบคุม ดังนั้น ผลที่ได้ในสภาพแวดล้อมอื่นอาจแตกต่างกันได้ ค่าที่วัดได้บางอย่างอาจทำขึ้นในขั้นตอนการพัฒนาและไม่มีประกันว่าผลที่ได้เหล่านี้จะเหมือนกับระบบที่วางจำหน่าย ยิ่งไปกว่านั้น เกณฑ์บางอย่างอาจได้มาจากการประมาณโดยผ่านกระบวนการ extrapolation ค่าที่ได้จริงอาจแตกต่างกันได้ ผู้ใช้เอกสารนี้ควรตรวจสอบข้อมูลที่ใช้ได้สำหรับสภาพแวดล้อมเฉพาะสำหรับผู้ใช้

ข้อมูลเกี่ยวกับผลิตภัณฑ์ที่ไม่ได้จัดทำโดย IBM ได้รับมาจากซัพพลายเออร์ของผลิตภัณฑ์เหล่านั้น, ประกาศที่เผยแพร่หรือแหล่งข้อมูลที่เปิดเผยต่อสาธารณะ IBM ไม่ได้ทดสอบผลิตภัณฑ์เหล่านั้น และไม่ได้ยืนยันความถูกต้องของประสิทธิภาพการทำงาน ความเข้ากันได้ หรือค่ากล่าวอ้างอื่นๆ เกี่ยวกับผลิตภัณฑ์ที่ไม่ได้จัดทำโดย IBM คำถามเกี่ยวกับความเข้ากันได้ของผลิตภัณฑ์ที่ไม่ได้ผลิตโดย IBM ควรแจ้งกับซัพพลายเออร์ของผลิตภัณฑ์เหล่านั้น

ข้อความใดๆ ที่เกี่ยวข้องกับทิศทางในอนาคตและเจตจำนงของ IBM อาจมีการเปลี่ยนแปลง หรือเพิกถอนได้โดยไม่ต้องแจ้งล่วงหน้า และนำเสนอเฉพาะเป้าหมายและวัตถุประสงค์เท่านั้น

ข้อมูลนี้ประกอบด้วยตัวอย่างข้อมูลและรายงานที่ใช้ในการดำเนินธุรกิจประจำวัน เพื่อแสดงให้เห็นอย่างสมบูรณ์ที่สุดเท่าที่เป็นไปได้ ตัวอย่างเหล่านี้จึงประกอบด้วยชื่อของบุคคล บริษัท ตราสินค้า และผลิตภัณฑ์ ชื่อทั้งหมดเหล่านี้เป็นชื่อสมมติ และการคล้ายคลึงในชื่อและที่อยู่หน่วยธุรกิจที่มีอยู่จริงเป็นความบังเอิญทั้งสิ้น

COPYRIGHT LICENSE:

ข้อมูลนี้ประกอบด้วยแอปพลิเคชันตัวอย่างในภาษาต้นฉบับ ซึ่งแสดงเทคนิคในการเขียนโปรแกรมบนแพลตฟอร์มปฏิบัติการที่หลากหลาย คุณอาจดัดลอก ดัดแปลง หรือเผยแพร่โปรแกรมตัวอย่างเหล่านี้ในรูปแบบใดๆ โดยไม่ต้องจ่ายเงินให้กับ IBM สำหรับวัตถุประสงค์ของการพัฒนา การใช้งาน การตลาด หรือการเผยแพร่โปรแกรมแอปพลิเคชันที่ใช้ application programming interface สำหรับแพลตฟอร์มปฏิบัติการที่โปรแกรมตัวอย่างได้ถูกพัฒนาขึ้น ตัวอย่างเหล่านี้ ไม่ได้ผ่านการทดสอบภายใต้ทุกสถานการณ์ ดังนั้น IBM ไม่สามารถรับประกัน หรือกล่าวเป็นนัยถึงความเชื่อถือ ความสามารถในการให้บริการ หรือฟังก์ชันการทำงานของโปรแกรมเหล่านี้ได้

แต่ละสำเนาหรือบางส่วนของโปรแกรมตัวอย่าง หรืองานใดๆ ที่มาจากโปรแกรมเหล่านี้ ต้องมีข้อความแสดงลิขสิทธิ์ ดังนี้:

© (ชื่อบริษัทของคุณ) (ปี) บางส่วนของโค้ดนี้ถูกพัฒนามาจากโปรแกรมตัวอย่างของ IBM Corp. © Copyright IBM Corp. _ปีอนปี_. All rights reserved.

เอกสาร การจัดการระบบงาน เล่มนี้ จัดทำส่วนโปรแกรมมิงอินเทอร์เน็ตเฟสซึ่งช่วยผู้ใช้ในการเขียนโปรแกรมติดต่อเซอวีส์ต่างๆ ของ IBM i5/OS

หากคุณกำลังอ่านข้อมูลนี้ในรูปแบบที่เป็น softcopy รูปภาพและภาพประกอบนี้อาจไม่ปรากฏขึ้น

เครื่องหมายการค้า

ชื่อต่อไปนี้ เป็นเครื่องหมายการค้าของ International Business Machines Corporation ในประเทศสหรัฐอเมริกา, ประเทศอื่น หรือทั้งสองกรณี:

DB2

Domino

i5/OS

IBM

IBM (logo)

IPDS

Lotus Notes

OS/400

SP

System i

System i/36

WebSphere

Adobe, สัญลักษณ์ Adobe, PostScript และสัญลักษณ์ PostScript เป็นเครื่องหมายการค้าจดทะเบียน หรือเครื่องหมายการค้าของ Adobe Systems Incorporated ในประเทศสหรัฐอเมริกา และ/หรือประเทศอื่นๆ

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

Microsoft, Windows, Windows NT และสัญลักษณ์ Windows เป็นเครื่องหมายการค้าของ Microsoft Corporation ในประเทศสหรัฐอเมริกา, ประเทศอื่น หรือทั้งสองกรณี

ชื่อบริษัทอื่น, ชื่อผลิตภัณฑ์อื่น และชื่อบริการอื่น อาจเป็นเครื่องหมายการค้าหรือเครื่องหมายการบริการของผู้อื่น

ข้อกำหนดและเงื่อนไข

คำอนุญาตในการใช้เอกสารเหล่านี้เป็นไปตามข้อกำหนดและเงื่อนไขต่อไปนี้

การใช้งานเป็นส่วนตัว: คุณสามารถจัดทำสำเนาของเอกสารเหล่านี้เพื่อใช้เป็นการส่วนตัว มิใช่เพื่อการพาณิชย์โดยมีเงื่อนไขว่าจะต้องคงข้อความประกาศความเป็นเจ้าของไว้โดยครบถ้วน คุณไม่สามารถแจกจ่าย แสดงหรือสร้างงานที่สืบเนื่องจากหนังสือเหล่านี้ หรือมาจากบางส่วนของหนังสือเหล่านี้ โดยไม่ได้รับความยินยอมอย่างชัดแจ้งจาก IBM

การใช้งานในเชิงพาณิชย์: คุณสามารถจัดทำสำเนา แจกจ่าย และแสดงเอกสารนี้ได้เฉพาะภายในองค์กรของคุณ โดยมีเงื่อนไขว่าจะต้องคงข้อความประกาศความเป็นเจ้าของไว้โดยครบถ้วน คุณไม่สามารถสร้างงานที่สืบเนื่องจากหนังสือเหล่านี้ หรือสร้างหนังสือเหล่านี้ใหม่ แจกจ่าย หรือแสดงหนังสือเหล่านี้ทั้งหมดหรือเป็นบางส่วนออกไปยังภายนอกองค์กรของคุณ โดยไม่ได้รับความยินยอมอย่างชัดแจ้งจาก IBM

นอกเหนือจากคำอนุญาตที่ได้แสดงไว้ในที่นี้ IBM ไม่ได้ให้อำนาจดำเนินการ โฆษณาสื่อหรือสิทธิ์อื่นใด ทั้งโดยชัดแจ้งและโดยนัยกับเอกสารเหล่านี้ รวมถึงสารสนเทศ ข้อมูล ซอฟต์แวร์ หรือทรัพย์สินทางปัญญาอื่นๆ ที่อยู่ภายในที่นี้

IBM ขอสงวนสิทธิ์ในการเพิกถอนคำอนุญาตที่ให้ไว้ในที่นี้ เมื่อใดก็ตามที่พิจารณาแล้วว่า การใช้เอกสารเหล่านี้ก่อให้เกิดความเสียหายต่อผลประโยชน์ของบริษัท หรือเมื่อ IBM ได้พิจารณาแล้วว่า ไม่มีการปฏิบัติตามข้อกำหนดข้างต้นอย่างเหมาะสม

คุณไม่สามารถดาวน์โหลด เอ็กซ์พอร์ตหรือทำการเอ็กซ์พอร์ตข้อมูลนี้เข้าได้ ยกเว้นจะได้ปฏิบัติตามกฎหมายและข้อบังคับที่กำหนดไว้ รวมไปถึงกฎหมายและข้อบังคับในการเอ็กซ์พอร์ตของสหรัฐอเมริกา

IBM ไม่ขอรับประกันเกี่ยวกับเนื้อหาของเอกสารเหล่านี้ เอกสารเหล่านี้นำเสนอเนื้อหาความ "ตามสภาพที่เป็น" โดยไม่มีการรับประกันใดๆ ไม่ว่าจะโดยชัดแจ้งหรือโดยนัย ตลอดจนไม่มีการรับประกันโดยนัยต่อความสามารถในการจำหน่าย การไม่ละเมิด และความเหมาะสมสำหรับวัตถุประสงค์อย่างใดอย่างหนึ่ง



พิมพีในสหรัฐอเมริกา