FUJITSU

Fujitsu Software Systemwalker Operation Manager V17.2.0

リファレンス集

UNIX/Windows(R)共通

J2X1-8185-09Z0(00)
2026年6月

まえがき

本書の目的

本書は、Systemwalker Operation ManagerのコマンドやJCLなどのリファレンス情報について説明しています。

本書の読者

本書は、Systemwalker Operation Managerを使用して、業務をスケジュール、操作、監視、管理する方を対象としています。

本書を読む場合、OSやGUIの一般的な操作方法をご理解の上でお読みください。

本書の表記について

固有記事の表記、略称表記については、“A.1 略称表記について”を参照してください。

登録商標について

登録商標については、“A.2 商標について”を参照してください。

輸出管理規制について

本ドキュメントを輸出または第三者へ提供する場合は、お客様が居住する国および米国輸出管理関連法規等の規制をご確認のうえ、必要な手続きをおとりください。

出版年月および版数

版数	マニュアルコード
2021年 8月 第1版	J2X1-8185-01Z0(00)/ J2X1-8185-01Z2(00)
2021年 9月 第2版	J2X1-8185-02Z0(00)/ J2X1-8185-02Z2(00)
2021年 10月 第3版	J2X1-8185-03Z0(00)/ J2X1-8185-03Z2(00)
2022年 10月 第4版	J2X1-8185-04Z0(00)/ J2X1-8185-04Z2(00)
2023年 10月 第5版	J2X1-8185-05Z0(00)/ J2X1-8185-05Z2(00)
2024年 2月 第6版	J2X1-8185-06Z0(00)/ J2X1-8185-06Z2(00)
2025年 2月 第7版	J2X1-8185-07Z0(00)/ J2X1-8185-07Z2(00)
2025年 8月 第8版	J2X1-8185-08Z0(00)/ J2X1-8185-08Z2(00)
2026年 6月 第9版	J2X1-8185-09Z0(00)/ J2X1-8185-09Z2(00)

著作権表示

Copyright 1995-2026 Fujitsu Limited

目　次

第1章 共通コマンド 1
1.1 soperationmgr サービス/デーモン起動コマンド 1
1.2 poperationmgr サービス/デーモン停止コマンド 2
1.3 swstop サービス/デーモン起動の抑止コマンド 3
1.4 swstart サービス/デーモン起動の抑止解除コマンド 5
1.5 swsilent サイレントインストールコマンド 6
1.6 jmmode 実行継続モード切り替えコマンド 9
1.7 mpbko バックアップコマンド 10
1.8 mprso リストアコマンド 14
1.9 omgrmonitor 監視ホスト情報取込みコマンド 17
1.10 createsubsystem.exe/sh サブシステム環境作成コマンド 20
1.11 deletesubsystem.exe サブシステム環境削除コマンド【Windows版】 21
1.12 swmove 移行用退避コマンド【UNIX版】 22
1.13 swtrans 移行用変換・登録コマンド【UNIX版】 23
1.14 mpsetcluster クラスタ情報の登録/削除コマンド 24
1.15 swpkginfo 製品情報表示コマンド(サーバ用)【UNIX版】 26
1.16 F3crfver 製品情報表示コマンド 27
1.17 setup Systemwalker Operation Managerインストールコマンド【Windows版】 35
1.18 swuset Systemwalker Operation Managerアンインストールコマンド【Windows版】 35
1.19 mpowebsetup Webサーバセットアップコマンド 36
1.20 mpowebconfig Webサーバ設定変更コマンド 37
1.21 jmsetauthinfo 共通基盤認証情報設定コマンド【Windows版】 40
1.21.1 共通基盤認証情報制御文 41

第2章 セキュリティコマンド 43
2.1 mpsetaclコマンド 43
2.2 mpchownコマンド 46
2.3 mkbatコマンド 48
2.4 mpaclclsコマンド 49
2.5 mpcssaveコマンド 51
2.6 mplstacluserコマンド 52
2.7 mpsetusermodeコマンド【UNIX版】 53
2.8 mpusermodeコマンド【UNIX版】 55
2.9 mpadduserコマンド【UNIX版】 55
2.10 mpdeluserコマンド【UNIX版】 57
2.11 mpmoduserコマンド【UNIX版】 58
2.12 mpsetpasswdコマンド【UNIX版】 60
2.13 mpusersコマンド【UNIX版】 61
2.14 mpsetupolmodeコマンド【UNIX版】 63
2.15 mpupolmodeコマンド【UNIX版】 64
2.16 mpprjcmdaclコマンド【UNIX版】 65
2.17 mpmkpwbatコマンド【UNIX版】 66
2.18 mpsetlogsend_omgrコマンド 68

第3章 カレンダ/電源制御コマンド 71
3.1 f3crheet 待ち合わせ終了通知コマンド 71
3.2 calregpath カレンダ自動反映の共用ディスクディレクトリ登録コマンド【Windows版】 72
3.3 calsetcluster カレンダ自動反映セットアップコマンド 73
3.4 calrecpl カレンダ反映先ホスト定義の再読み込みコマンド 75

第4章 ジョブスケジューラコマンド 77
4.1 コマンド一覧 77
4.2 jobschsetholコマンド 82
4.2.1 休日情報制御文 83
4.3 jobschsetcalコマンド 85
4.3.1 起動日雛形制御文 86

4.4 jobschcodコマンド ... 88
4.5 jobschsetprojectコマンド ... 89
4.5.1 プロジェクト情報制御文 ... 91
4.6 jobschgetoutputコマンド ... 93
4.7 jobschinitコマンド ... 96
4.8 stemSchDeleteコマンド ... 97
4.9 jobschsetauthinfoコマンド【Windows版】【Linux版】 ... 100
4.9.1 認証情報制御文【Windows版】【Linux版】 ... 102
4.10 jobschrefstatusコマンド【Windows版】【Linux版】 ... 111
4.10.1 状態情報ファイル ... 115
4.11 jobschctlprojectコマンド【Windows版】【Linux版】 ... 117
4.12 jobschsetnetコマンド ... 118
4.12.1 ジョブネット制御文 ... 123
4.12.2 ジョブ制御文 ... 138
4.12.3 起動日制御文 ... 152
4.12.4 メッセージ事象制御文 ... 157
4.12.5 一時変更制御文 ... 159
4.13 jobschsetgrpコマンド ... 161
4.13.1 グループ制御文 ... 163
4.13.2 グループ内ジョブネット制御文 ... 164
4.14 jobschctljobコマンド ... 165
4.15 jobschcontrolコマンド ... 167
4.16 jobschctlgrpコマンド ... 169
4.17 jobschoperateコマンド ... 171
4.18 jobschmoveコマンド ... 173
4.19 jobschmsgclearコマンド ... 174
4.20 jobschmsgeventコマンド ... 176
4.21 jobschcancelnetコマンド【Windows版】 ... 181
4.22 jobschdeftocsvコマンド ... 182
4.23 jobschcsvtodefコマンド ... 186
4.24 jobschchecktimeコマンド ... 192
4.25 jobschchecklogコマンド ... 195
4.26 chkfile / jobschchkfileコマンド ... 198
4.27 jobschnetmemoコマンド ... 203
4.28 jobschconvertコマンド ... 206
4.29 jobschchkmsgevtコマンド ... 207
4.30 jobschchknetvarコマンド ... 208
4.31 jobschsleepコマンド ... 211
4.32 jobschrestexecコマンド【Windows版】【Linux版】 ... 212

第5章 ジョブスケジューラコマンド(印刷用) ... 216
5.1 コマンド一覧 ... 216
5.2 jobschprint -j(-J)コマンド ... 218
5.3 jobschprint -n(-N)コマンド ... 222
5.4 jobschprint -a(-A)コマンド ... 225
5.5 jobschprint -e(-E)コマンド ... 234
5.6 jobschprint -f(-F)コマンド ... 236
5.7 jobschprint -g(-G)コマンド ... 239
5.8 jobschprint -i(-I)コマンド ... 242
5.9 jobschprint -l(-L)コマンド ... 251
5.10 jobschprint -m(-M)コマンド ... 253
5.11 jobschprint -p(-P)コマンド ... 255
5.12 jobschprint -q(-Q)コマンド【Windows版】 ... 258
5.13 jobschprint -Rコマンド ... 260
5.14 jobschprint -rコマンド ... 261
5.15 jobschprint -s(-S)コマンド ... 263
5.16 jobschprint -w(-W)コマンド ... 265

5.17 jobschprint -x(-X)コマンド 267
5.18 jobschprint -y(-Y)コマンド 270
5.19 jobschprint -bコマンド 272
5.20 jobschprint -cコマンド 273
5.21 jobschprint -dコマンド 275
5.22 jobschprint -hコマンド 276
5.23 jobschprint -uコマンド 278
5.24 jobschprint -vコマンド 279
5.25 jobschprint -o(-O)コマンド 280
5.26 jobschprint -rhコマンド 282
5.27 jobschprint -rspコマンド 283
5.28 jobschprintcsv -nコマンド 285
5.29 jobschprintcsv -gコマンド 287
5.30 jobschprintcsv -lコマンド 290
5.31 jobschprintcsv -fコマンド 292
5.32 stemprintprojectコマンド 295

第6章 マスタスケジュール管理コマンド 297
6.1 コマンド一覧 297
6.2 stemenvset 環境構築コマンド 298
6.3 stemManager スケジュール管理コマンド 299
6.4 stemSetPrjMaster プロジェクトマスタ設定コマンド 301
6.5 stemCopyProject 運用変更事前設定コマンド 303
6.6 stemMoveProject 運用変更リリース日変更コマンド 304
6.7 stemBackProject 運用変更取消コマンド 305
6.8 jobschmsgeventex メッセージ事象発生コマンド 306
6.9 stemprintnet 退避情報確認コマンド 309
6.10 stemConfig マスタスケジュール管理環境設定コマンド【UNIX版】 311
6.11 stemprint -f(-F)/-g(-G)コマンド 312
6.12 stemprint -n(-N)コマンド 316
6.13 stemschnet ジョブネット投入コマンド 319
6.14 stemsetdbpath クラスタ用DBディレクトリ変更コマンド【Windows版】 321
6.15 stemregservice クラスタ用サービス登録コマンド【Windows版】 323

第7章 ジョブ実行制御コマンド 325
7.1 コマンド一覧 325
7.2 mjstart 運用開始コマンド 326
7.3 mjstop 運用停止コマンド 327
7.4 mjsview ログファイル出力コマンド 328
7.5 qalter キューの属性変更コマンド 332
7.6 qjalter ジョブの属性変更コマンド 334
7.7 qstart キューの起動コマンド 336
7.8 qstop キューの停止コマンド 336
7.9 mjdjadd 分散実行の構成ホスト追加コマンド 337
7.10 mjdjdel 分散実行の構成ホスト削除コマンド 339
7.11 mjdjalt 分散実行の設定変更コマンド 341
7.12 qdel ジョブの削除コマンド 342
7.13 qhold ジョブの保留(ホールド)コマンド 343
7.14 qjstat ジョブ情報の表示コマンド 345
7.15 qmove ジョブのキュー間移動コマンド 348
7.16 qrls ジョブの保留解除(リリース)コマンド 350
7.17 qstat キューの状態表示コマンド 351
7.18 qsub ジョブの投入コマンド 353
7.18.1 ジョブ実行制御の環境変数設定ファイル 362
7.18.2 転送する入出力ファイルの指定方法 363
7.19 mjdjstat 分散実行の状態表示コマンド 365
7.20 mjsmdset 起動モード指定コマンド 366

7.21 mjrmtjob リモートマシン上のコマンド/シェルスクリプト実行コマンド....367
7.22 mjkeygen キーペア作成コマンド【Windows版】....371

第8章 アクション管理コマンド【Windows版】....375
8.1 Mp_SpAct アクション停止コマンド....375
8.2 Mp_SpSnd 音声停止コマンド....375

第9章 業務連携コマンド【Windows版】....377
9.1 サーバ業務連携コマンド....377
9.1.1 EXECASVR.EXE アプリケーション起動コマンド....377
9.1.2 FCOPYSVR.EXE ファイル転送コマンド....379
9.1.3 FCTRLSVR.EXE ファイル操作コマンド....383
9.1.4 FPRSSVR.EXE ファイル圧縮/伸長コマンド....387
9.2 クライアント業務連携コマンド....390
9.2.1 PONCLI.EXE 電源投入コマンド....390
9.2.2 POFFCLI.EXE 電源切断コマンド....392
9.2.3 EXECACLI.EXE アプリケーション起動コマンド....394
9.2.4 FCOPYCLI.EXE ファイル転送コマンド....396
9.2.5 FCTRLCLI.EXE ファイル操作コマンド....400
9.2.6 FPRSCLI.EXE ファイル圧縮/伸長コマンド....404
9.3 業務連携管理コマンド....406
9.3.1 LGMANAGE.EXE パスワード管理コマンド....406

第10章 業務連携コマンド【UNIX版】....409
10.1 サーバ業務連携コマンド....409
10.1.1 execasvr アプリケーション起動コマンド....409
10.1.2 fcopysvr ファイル転送コマンド....411
10.1.3 fctrlsvr ファイル操作コマンド....415
10.1.4 fprssvr ファイル圧縮/伸長コマンド....418
10.2 クライアント業務連携コマンド....421
10.2.1 poncli 電源投入コマンド....421
10.2.2 poffcli 電源切断コマンド....423
10.2.3 execacli アプリケーション起動コマンド....425
10.2.4 fcopycli ファイル転送コマンド....428
10.2.5 fctrlcli ファイル操作コマンド....432
10.2.6 fprscli ファイル圧縮/伸長コマンド....435
10.3 業務連携管理コマンド....439
10.3.1 lgmanage パスワード管理コマンド....439

第11章 ユーティリティコマンド【Windows版】....441
11.1 chksrv サービス状態確認コマンド....441
11.2 f3crhsst サービス開始コマンド....442
11.3 f3crhssp サービス停止コマンド....443

第12章 SSL環境設定コマンド....444
12.1 ahsregistupin パスフレーズ登録コマンド....444
12.2 openssl RSA秘密鍵の作成/CSRの作成/証明書のフィンガープリント表示/DHパラメータファイル作成/ECDSA秘密鍵の作成コマンド....445

第13章 保守コマンド....450
13.1 mpipcso IPC資源状態表示コマンド【UNIX版】....450
13.2 mppviewo プロセス動作状況表示コマンド....452
13.3 swcolinf 保守情報収集コマンド(サーバ用)【UNIX版】....454
13.4 swcolinf 保守情報収集コマンド....456
13.5 mpenvset 運用環境保守ウィザード起動コマンド【Windows版】....459

第14章 グローバルサーバジョブ連携コマンド....460
14.1 コマンド一覧....460
14.2 mfexecu 未定義ジョブ起動コマンド....460

14.3 mfexecd 定義済みジョブ起動コマンド 462
14.4 mfusradd 利用者登録コマンド 463
14.5 mfusrdel 利用者削除コマンド 463
14.6 mfusrmod 利用者変更コマンド 464
14.7 mfusrpasswd 利用者パスワード登録コマンド 464
14.8 mfjobadd 業務情報登録コマンド 465
14.9 mfjobdel 業務情報削除コマンド 466
14.10 mfjobmod 業務情報変更コマンド 466
14.11 mfjobpasswd 業務パスワード登録コマンド 466

第15章 JCL(業務手続き言語) 468
15.1 JCLの概要 468
15.1.1 JCLの機能 468
15.1.2 JCLの構成 468
15.1.3 JCLの規約 469
15.2 ジョブ全体に関する制御文 470
15.2.1 jobstart文 470
15.2.2 host文 472
15.2.3 user文 472
15.2.4 resource文 473
15.2.5 qprty文 474
15.2.6 dprty文 474
15.2.7 path文 475
15.2.8 file文 476
15.2.9 jobend文 478
15.3 ジョブステップに関する制御文 478
15.3.1 stepstart文 478
15.3.2 path文 480
15.3.3 file文 481
15.3.3.1 connectコマンド 484
15.3.3.2 sendコマンド 484
15.3.3.3 receiveコマンド 485
15.3.3.4 executeコマンド 485
15.3.3.5 disconnectコマンド 486
15.3.4 exec文 486
15.3.5 stepend文 487
15.4 JCLとプログラムの関係 488
15.4.1 path文とプログラムの関係 488
15.4.2 file文とプログラムの関係 488
15.4.3 exec文とプログラムの関係 489
15.5 JCL・ジョブステップの終了コード 489
15.5.1 JCLの終了コード 489
15.5.2 ジョブステップの終了コード 489

第16章 Systemwalkerスクリプト 490
16.1 Systemwalkerスクリプトの概要 490
16.1.1 Systemwalkerスクリプトの機能 490
16.1.2 Systemwalkerスクリプトの構成 490
16.1.3 Systemwalkerスクリプトの文法 491
16.1.4 正規表現 495
16.2 Systemwalkerスクリプトで使用するコマンド・制御文 498
16.2.1 after(一定時間処理を休止する) 498
16.2.2 break(ループを中止する) 499
16.2.3 catch(スクリプト行を実行し処理例外をトラップする) 499
16.2.4 cd(作業ディレクトリを変更する) 502
16.2.5 close(オープンされたファイルをクローズする) 502
16.2.6 continue(ループを次の繰り返しまでスキップする) 503

16.2.7 exec(サブプロセスを起動する)..... 504
16.2.8 exit(スクリプトを終了する)..... 511
16.2.9 expr(算術演算をする)..... 511
16.2.10 flush(バッファリングされた出力をフラッシュする)..... 512
16.2.11 gets(ファイルから1行読む)..... 513
16.2.12 if～else(条件分岐をする)..... 513
16.2.13 incr(変数に値を加算する)..... 515
16.2.14 open(ファイルをオープンする)..... 516
16.2.15 puts(ファイルに書き込む)..... 517
16.2.16 regexp(文字列と正規表現のマッチング検査と一致部分の切り出しをする)..... 518
16.2.17 regsub(正規表現のパターンマッチングに基づいて置換する)..... 519
16.2.18 set(変数に値を格納する)..... 520
16.2.19 string(文字列を操作する)..... 521
16.2.20 while(ループする)..... 523
16.3 デバッグで使用するコマンド..... 524
16.3.1 トレース拡張コマンド..... 524
16.3.1.1 sw_TcOpenTrace (トレースファイルオープンコマンド)..... 524
16.3.1.2 sw_TcWriteTrace (トレース情報出力コマンド)..... 526
16.3.1.3 sw_TcCloseTrace (トレースファイルクローズコマンド)..... 527
16.3.2 swotclsh(スクリプト実行コマンド)..... 528
16.4 スクリプトの例..... 530
16.5 スクリプトのデバッグ例..... 535

第17章 API情報..... 538
17.1 カレンダAPI..... 538
17.1.1 カレンダ名一覧取得API..... 540
17.1.2 カレンダ登録API(Mp_AddCalendar2)..... 542
17.1.3 カレンダ更新API(Mp_SetCalendar2)..... 545
17.1.4 カレンダ更新API(Mp_SetCalendar3)..... 547
17.1.5 カレンダ情報取得API..... 551
17.1.6 日変わり時刻取得API..... 554
17.2 ジョブスケジューラAPI..... 555
17.2.1 ジョブネット操作API..... 556
17.2.2 ジョブネット操作API/EE..... 560
17.2.3 ジョブネット操作(起動パラメタ指定)API..... 564
17.2.4 ジョブネット操作(起動パラメタ指定)API/EE..... 569
17.2.5 グループ操作API..... 574
17.2.6 グループ操作API/EE..... 578
17.2.7 ジョブネット起動API【Windows版】..... 581
17.2.8 ジョブネット起動API/EE【Windows版】..... 584
17.2.9 ジョブネット起動時刻変更API..... 587
17.2.10 ジョブネット起動時刻変更API/EE..... 592
17.3 ジョブ実行制御API..... 598
17.3.1 導入/稼働情報獲得API【Windows版】..... 599
17.3.2 導入/稼働情報獲得API/EE【Windows版】..... 600
17.3.3 ジョブ投入API..... 602
17.3.4 ジョブ保留API【Windows版】..... 605
17.3.5 ジョブ保留API/EE【Windows版】..... 607
17.3.6 ジョブ保留解除API【Windows版】..... 609
17.3.7 ジョブ保留解除API/EE【Windows版】..... 611
17.3.8 ジョブ削除API..... 613
17.3.9 ジョブ削除API/EE..... 616
17.3.10 ジョブ移動API【Windows版】..... 619
17.3.11 ジョブ移動API/EE【Windows版】..... 621
17.3.12 ジョブ状態獲得API..... 623
17.3.13 ジョブ状態獲得API/EE..... 627
17.3.14 ジョブ一覧情報獲得API【Windows版】..... 631

17.3.15 ジョブ一覧情報獲得API/EE【Windows版】 634
17.3.16 キュー一覧情報獲得API【Windows版】 638
17.3.17 キュー一覧情報獲得API/EE【Windows版】 641
17.3.18 エラーメッセージ獲得API 645
17.4 アクション管理API【Windows版】 645
17.4.1 アクション実行API 646
17.4.1.1 音声通知API 647
17.4.1.2 音声停止API 650
17.4.1.3 ポップアップメッセージ送信(指定ユーザ)API 651
17.4.1.4 ポップアップメッセージ送信(ドメイン)API 654
17.4.1.5 ポップアップメッセージ送信(全ユーザ)API 655
17.4.1.6 MS-Mail送信API 657
17.4.1.7 E-Mail送信API 661
17.4.1.8 ショートメールへのメッセージ送信API 663
17.4.2 アクション実行API(旧バージョン互換) 667
17.4.2.1 音声通知API 667
17.4.2.2 ポップアップメッセージ送信(指定ユーザ)API 670
17.4.2.3 ポップアップメッセージ送信(ドメイン)API 672
17.4.2.4 ポップアップメッセージ送信(全ユーザ)API 674
17.4.3 アクション停止API 675
17.4.3.1 アクション停止API 675
17.4.4 アクション状態通知API 677
17.4.4.1 アクション状態通知API 677
17.4.4.2 アクション情報領域の解放API 681
17.4.5 アクション状態通知API(旧バージョン互換) 682
17.4.5.1 アクション状態通知API 682
17.4.5.2 アクション情報領域の解放API 687

第18章 Web API【Windows版】【Linux版】 688
18.1 Web APIの概要 688
18.2 APIリファレンス 688
18.2.1 API形式 688
18.2.2 API一覧 689
18.2.3 共通仕様 689
18.2.3.1 データ型 689
18.2.3.2 リクエストの形式 690
18.2.3.3 レスポンスの形式 691
18.2.3.4 ユーザ認証 693
18.2.3.5 サブシステム番号 693
18.3 各APIのリファレンス 694
18.3.1 projectsリソース 694
18.3.1.1 プロジェクト情報の取得 694
18.3.1.2 プロジェクトの操作 696
18.3.1.3 プロジェクト情報の一覧取得 697
18.3.2 jobnetsリソース 698
18.3.2.1 ジョブネット情報の取得 701
18.3.2.2 ジョブネットの操作 702
18.3.2.3 ジョブネット情報の一覧取得 705
18.3.3 jobsリソース 705
18.3.3.1 ジョブ情報の取得 707
18.3.3.2 ジョブの操作 708
18.3.3.3 ジョブ情報の一覧取得 710
18.3.4 queuesリソース 711
18.3.4.1 キュー情報の取得 712
18.3.4.2 キューの定義の一時変更 714
18.3.4.3 キューの操作 715
18.3.4.4 キュー情報の一覧取得 717

18.3.5 contained-hostsリソース ... 718
18.3.5.1 構成ホスト情報の取得 ... 719
18.3.5.2 構成ホストの定義の一時変更 ... 720
18.3.5.3 構成ホストの削除 ... 722
18.3.5.4 構成ホストの追加 ... 723
18.3.5.5 構成ホスト情報の一覧取得 ... 725

第19章 出口情報 ... 727
19.1 ジョブネット正常終了/異常終了出口 ... 727
19.2 ジョブネット異常終了拡張出口 ... 730
19.3 ジョブ終了出口 ... 734
19.4 ログファイル切換え出口 ... 737
19.5 日変わり時刻到来出口 ... 739
19.6 正常終了/異常終了シャットダウン出口【Windows版】 ... 741
19.7 ジョブネット起動遅延出口 ... 742
19.8 ジョブネット終了遅延出口 ... 744
19.9 サポート外のショートメールのメッセージ変換用出口【Windows版】 ... 746
19.10 デマンドジョブ終了出口 ... 747
19.11 出口情報(災対用)【Windows版】【Linux版】 ... 749
19.11.1 ジョブ開始出口(災対用) ... 750
19.11.2 ジョブ終了出口(災対用) ... 752
19.11.3 ジョブネット開始出口(災対用) ... 754
19.11.4 ジョブネット終了出口(災対用) ... 757
19.11.5 メッセージ事象発生出口(災対用) ... 759

第20章 ファイルリファレンス ... 762
20.1 ログファイル ... 762
20.1.1 ジョブ実行制御の稼働実績情報ファイル ... 764
20.1.1.1 サービスレコード ... 765
20.1.1.2 ジョブレコード ... 765
20.1.1.3 ステップレコード ... 768
20.1.2 監査ログのファイル ... 769
20.1.2.1 監査ログファイル ... 769
20.2 メッセージ事象発生履歴ファイル ... 773
20.2.1 メッセージ事象発生履歴ファイル ... 773
20.3 ジョブネット変数情報ファイル ... 775
20.3.1 ジョブネット変数情報ファイル ... 775
20.4 実行ユーザ制限のファイル【UNIX版】 ... 778
20.4.1 実行ユーザ制限リスト ... 778
20.5 インポート/エクスポート実行結果ファイル ... 779
20.5.1 インポート実行結果ファイル ... 779
20.5.2 インポート実行結果ファイル(V13.9.0以前のサーバに接続した場合) ... 780
20.5.3 エクスポート実行結果ファイル ... 781
20.5.4 エクスポート実行結果ファイル(V13.9.0以前のサーバに接続した場合) ... 782
20.6 クラスタ運用の定義ファイル ... 784
20.6.1 calcphost.def カレンダ反映先ホスト定義ファイル ... 784
20.6.2 cluster_ip.csv 定義ファイル ... 785
20.7 Webサーバポートの定義ファイル ... 786
20.7.1 Webサーバポート定義ファイル ... 786
20.8 実行サブシステム名の定義ファイル ... 787
20.8.1 実行サブシステム名定義ファイル ... 787
20.9 クラウドサービス連携の定義ファイル【Windows版】【Linux版】 ... 789
20.9.1 接続情報ファイル ... 789
20.9.1.1 AWS Batchサービスと連携する例 ... 808
20.9.1.2 AWS Lambdaサービスと連携する例 ... 809
20.9.1.3 AWS Step Functionsサービスと連携する例 ... 810
20.9.1.4 Amazon Simple Storage Service (Amazon S3)サービスと連携する例 ... 811

20.9.1.5 AWS Backupサービスと連携する例 816
20.9.1.6 Amazon Relational Database Service (Amazon RDS)サービスと連携する例 817
20.9.1.7 Amazon Elastic Compute Cloud (Amazon EC2)サービスと連携する例 818
20.9.1.8 Amazon Elastic Kubernetes Service (Amazon EKS)サービスと連携する例 819
20.9.1.9 Azure Batchサービスと連携する例 821
20.9.1.10 Azure Functionsサービスと連携する例 822
20.9.1.11 Azure Logic Appsサービスと連携する例 822
20.9.1.12 Azure Blob Storageサービスと連携する例 823
20.9.1.13 Azure Backupサービスと連携する例 828
20.9.1.14 Azure SQL Managed Instanceサービスと連携する例 832
20.9.1.15 Azure Virtual Machinesサービスと連携する例 833
20.9.1.16 Azure Kubernetes Service (AKS)サービスと連携する例 834
20.9.1.17 Oracle Cloud Infrastructure Functions(OCI Functions)サービスと連携する例 835
20.9.1.18 Oracle Cloud Infrastructure Compute(OCI Compute)サービスと連携する例 836
20.9.1.19 Oracle Cloud Infrastructure Kubernetes Engine(OKE)サービスと連携する例 837
20.9.1.20 Oracle Autonomous Databaseサービスと連携する例 839
20.9.1.21 Oracle Cloud Infrastructure Object Storage(OCI Object Storage)サービスと連携する例 840
20.10 PATH環境変数定義ファイル 846

第21章 イベントログ【Windows版】 849
21.1 イベントログの内容 849

第22章 ジョブの終了コード 851
22.1 ジョブスケジューラの終了コード 851
22.2 ジョブ実行制御の終了コード 853

付録A 本書の表記、商標について 857
A.1 略称表記について 857
A.2 商標について 858

第1章 共通コマンド

本章では、Systemwalker Operation Managerが提供する共通コマンドについて説明します。なお、swsilentコマンド以外のコマンドは、Systemwalker Operation Managerのサーバ機能がインストールされているサーバで起動してください。

注意

WindowsのUAC機能を有効にしている環境でコマンドを実行する場合、OSにログインしたユーザの権限によって権限の昇格を求めるダイアログが表示され、対処が必要となります。

詳細は、“Systemwalker Operation Manager 運用ガイド”の、“Windowsでコマンドを実行する場合の注意事項【Windows版】”を参照してください。

1.1 soperationmgr サービス/デーモン起動コマンド

記述形式

【Windows版】

```
soperationmgr [/a]
```

【UNIX版】

```
soperationmgr [-a]
```

機能説明

Systemwalker Operation Managerに含まれる機能のサービス/デーモンを一括して起動します。

オプション

/a【Windows版】 -a【UNIX版】

Systemwalker Operation ManagerとSystemwalker Centric Managerが共存する環境で、両方のサービス/デーモンを同時に起動したい時に指定します。本オプションを省略した場合は、共通のサービス/デーモンを含めたSystemwalker Operation Managerのサービス/デーモンのみ起動します。

Systemwalker Centric Managerがインストールされていない場合は、本オプションの指定に関係なくSystemwalker Operation Managerのサービス/デーモンを起動します。

コマンド格納場所

Windows	Systemwalker Operation Managerインストールディレクトリ¥MpWalker.JM¥bin
Solaris HP-UX AIX Linux	/opt/systemwalker/bin

実行に必要な権限

システム管理者(Administratorsグループに属するユーザ/スーパーユーザ)権限が必要です。

注意事項

- 本コマンドはフルパスで使用します。

- Systemwalker Operation ManagerとSystemwalker Centric Managerで共通のサービス/デーモンは、本コマンドで起動要求した時にすでにSystemwalker Centric Managerで起動されていても、多重起動は無視されるため、起動要求を行います。
- サービス/デーモン起動中にエラーが発生した場合は、エラーに対処してから、再度soperationmgrコマンドを実行してサービス/デーモンを起動してください。
- 本コマンドで起動するサービスは、[コントロールパネル]の[サービス]で、自動になっているものが対象となります。あらかじめ[コントロールパネル]の[サービス]で自動を設定しておいてください。【Windows版】
- V5.0L10以前のバージョンレベルのSystemWalker/CentricMGRがインストールされている場合も、本コマンドで起動の対象となります。【Windows版】
- 本コマンドをSystemwalker Operation Managerのジョブから実行した場合、Systemwalker Operation Manager/Systemwalker Centric Managerの動作は保証されません。詳細については、“Systemwalker Operation Manager 運用ガイド”の“ジョブとして実行してはいけないコマンド”を参照してください。
- Solaris版 Systemwlaker Centric Manager V17.0.0/V17.0.0Aと共存し、LD_LIBRARY_PATH_64環境変数が設定されている場合は、/opt/systemwalker/lib を LD_LIBRARY_PATH_64環境変数に追加し、本コマンドを実行してください。【Solaris版】

1.2 poperationmgr サービス/デーモン停止コマンド

記述形式

【Windows版】

```
poperationmgr [/a]
```

【UNIX版】

```
poperationmgr [-s|-a]
```

機能説明

Systemwalker Operation Managerに含まれる機能のサービス/デーモンを一括して停止します。

オプション

/a【Windows版】 -a【UNIX版】

Systemwalker Operation ManagerとSystemwalker Centric Managerが共存する環境で、両方のサービス/デーモンを同時に停止したい時に指定します。本オプションを省略した場合は、Systemwalker Centric Managerの運用に影響しないSystemwalker Operation Managerのサービス/デーモンのみ停止します。

Systemwalker Centric Managerがインストールされていない場合は、本オプションの指定に関係なくSystemwalker Operation Managerのサービス/デーモンを停止します。

-s【UNIX版】

Systemwalker Operation ManagerとSystemwalker Centric Managerが同じマシンにインストールされている場合に、ACLマネージャのデーモンも停止したい時に指定します。本オプションを省略した場合は、ACLマネージャのデーモンは停止されません。

Systemwalker Centric Managerがインストールされていない場合は、本オプションの指定に関係なくACLマネージャのデーモンを停止します。

-aオプションとの併用はできません。

コマンド格納場所

Windows	Systemwalker Operation Managerインストールディレクトリ¥MpWalker.JM¥bin
Solaris HP-UX AIX Linux	/opt/systemwalker/bin

実行に必要な権限

システム管理者(Administratorsグループに属するユーザ/スーパーユーザ)権限が必要です。

注意事項

- 本コマンドはフルパスで使用します。
- サービス/デーモンを停止する場合は、実行中のジョブネットがないことを確認してください。
- サービス/デーモン停止中にエラーが発生した場合は、エラーに対処してから再度poperationmgrコマンドを実行してサービス/デーモンを停止してください。
- V5.0L10以前のバージョンレベルのSystemWalker/CentricMGRがインストールされている場合も、本コマンドで停止の対象となります。【Windows版】
- Systemwalker Operation ManagerとSystemwalker Centric Managerが混在するシステムでは、-sオプションを指定してACLマネージャデーモンを停止している間、以下のSystemwalker Centric Managerの機能は使用できません。【UNIX版】

  ー GUIによる操作

  ー 資源配付

  また、ACLマネージャを停止した場合、Systemwalker Centric ManagerとSystemwalker Operation Managerの両製品の共通機能であるACLマネージャのデーモンプロセス(MpFwsec)も停止します。このため、Systemwalker Centric Managerのプロセス監視機能が以下のイベントを出力しますが問題ありません。対処は不要です。

  [メッセージ]

Systemwalker Centric Managerのプロセス(MpFwsec)が正常に動作しているか確認してください。

- 本コマンドをSystemwalker Operation Managerのジョブから実行した場合、Systemwalker Operation Manager/Systemwalker Centric Managerの動作は保証されません。詳細については、“Systemwalker Operation Manager 運用ガイド”の“ジョブとして実行してはいけないコマンド”を参照してください。
- Solaris版 Systemwlaker Centric Manager V17.0.0/V17.0.0Aと共存し、LD_LIBRARY_PATH_64環境変数が設定されている場合は、/opt/systemwalker/lib を LD_LIBRARY_PATH_64環境変数に追加し、本コマンドを実行してください。【Solaris版】

1.3 swstop サービス/デーモン起動の抑止コマンド

記述形式

```
swstop
```

機能説明

Systemwalker Operation Managerのインストール後にシステムの再起動を行った場合に、Systemwalker Operation Managerのサービスの起動を抑止します。Systemwalker Centric Manager V11.0L10/ V11.0以降と共存している環境の場合、Systemwalker Centric Managerのサービスの起動も抑止します。

オプション

なし

復帰値

復帰値	意 味
0	正常終了
0以外	異常終了

コマンド格納場所

Windows	Systemwalker Operation Managerインストールディレクトリ¥mpwalker.jm¥mpcmtool¥bin
Solaris HP-UX AIX Linux	/opt/FJSVftlo/bin

実行に必要な権限

システム管理者(Administratorsグループに属するユーザ/スーパーユーザ)権限が必要です。

注意事項

- 本コマンドは、サーバで実行できます。
- **【UNIX版】**
  すべてのデーモンの自動起動を抑止するには、本コマンドの実行に加えて、以下の手順で起動シェルスクリプトを無効にしてください。

  1. スーパーユーザになります。
  2. Systemwalker Operation Managerをpoperationmgrコマンドで停止します。
  3. デーモンの自動起動の抑止を設定します。

     【Solaris版】

     以下のシェルスクリプトのファイル名を変更(先頭の大文字の“S”を小文字の“s”に変更)します。

変更前ファイル名	変更後ファイル名	備考
/etc/rc2.d/ S90mpfwsec	/etc/rc2.d/s90mpfwsec	
/etc/rc3.d/S28JMCAL	/etc/rc3.d/s28JMCAL	注

     注）クラスタシステムで自動反映機能を有効にしている場合のみ

     【Linux版】

     以下のコマンドを実行し、systemdからの自動起動を無効にします。

```
systemctl disable mpfwsec.service
systemctl disable JMCAL.service (注)
```

     注）クラスタシステムで自動反映機能を有効にしている場合のみ

  上記の手順の実施により、次回のサーバ起動のタイミングで、デーモンが起動抑止されます。

- 本コマンドは、Systemwalker Centric Manager V11.0L10/V11.0以降で提供しているため、Systemwalker CentricMGR V10.0L21/V10.1以前と共存している環境の場合、Systemwalker Centric Managerのサービス/デーモンの起動を抑止することができません。
- Systemwalker Centric Manager V11.0L10/11.0以降と共存している環境の場合、Systemwalker Operation Manager のインストール/アンインストール前に、 swstart（サービス/デーモン起動の抑止解除コマンド）で起動抑止の設定を解除してください。詳細については、“1.4 swstart サービス/デーモン起動の抑止解除コマンド”を参照してください。

記述例

【Windows版】

```
d:¥win32app¥mpwalker.jm¥mpcmtool¥bin¥swstop
```

【UNIX版】

```
/opt/FJSVftlo/bin/swstop
```

1.4　swstart サービス/デーモン起動の抑止解除コマンド

記述形式

```
swstart
```

機能説明

Systemwalker Operation Managerのサービス/デーモンが起動するように制御し、Systemwalker Operation Managerのサービス/デーモンを起動します。Systemwalker Centric Manager V11.0L10/11.0以降と共存している環境の場合、Systemwalker Centric Managerのサービス/デーモンも起動するように制御し、Systemwalker Centric Managerのサービス/デーモンを起動します。

オプション

なし

復帰値

復帰値	意 味
0	正常終了
0以外	異常終了

コマンド格納場所

Windows	Systemwalker Operation Managerインストールディレクトリ¥mpwalker.jm¥mpcmtool¥bin
Solaris HP-UX AIX Linux	/opt/FJSVftlo/bin

実行に必要な権限

システム管理者(Administratorsグループに属するユーザ/スーパーユーザ)権限が必要です。

注意事項

- 本コマンドは、サーバで実行できます。

  本コマンドをSystemwalker Operation Managerのジョブから実行した場合、Systemwalker Operation Manager/Systemwalker Centric Managerの動作は保証されません。詳細については、“Systemwalker Operation Manager 運用ガイド”の“ジョブとして実行してはいけないコマンド”を参照してください。

- **【UNIX版】**
  すべてのデーモンの自動起動抑止を解除するには、本コマンドの実行に加えて、以下の手順で起動シェルスクリプトを有効にしてください。

  1. スーパーユーザになります。
  2. Systemwalker Operation Managerをpoperationmgrコマンドで停止します。
  3. デーモンの自動起動の抑止解除を設定します。

     【Solaris版】

     以下のシェルスクリプトのファイル名を変更(先頭の小文字の“s”を大文字の“S”に変更)します。

変更前ファイル名	変更後ファイル名	備考
/etc/rc2.d/s90mpfwsec	/etc/rc2.d/S90mpfwsec	
/etc/rc3.d/s28JMCAL	/etc/rc3.d/S28JMCAL	注

     注）クラスタシステムで自動反映機能を有効にしている場合のみ

     【Linux版】

     以下のコマンドを実行し、自動起動を有効にします。

     ```
     systemctl enable JMCAL.service (注)
     systemctl enable mpfwsec.service
     ```

     注）クラスタシステムで自動反映機能を有効にしている場合のみ

  4. soperationmgrコマンドでデーモンを起動します。

- **【Solaris版】**
  Systemwlaker Centric Manager V17.0.0/V17.0.0Aと共存し、LD_LIBRARY_PATH_64環境変数が設定されている場合は、/opt/systemwalker/lib を LD_LIBRARY_PATH_64環境変数に追加し、本コマンドを実行してください。

記述例

【Windows版】

```
d:¥win32app¥mpwalker.jm¥mpcmtool¥bin¥swstart
```

【UNIX版】

```
/opt/FJSVftlo/bin/swstart
```

1.5　swsilent サイレントインストールコマンド

記述形式

【Windows版】

```
Swsilent.exe -i File [-y]
```

【UNIX版】

［サーバ］

```
swsilent.sh -i File [-y]
```

［クライアント］

```
swsilent.exe -i File [-y]
```

機能説明

Systemwalker Operation Managerのサイレントインストールを行います。

オプション

-i *File*

サイレントインストール情報ファイルをフルパスで指定します。途中に空白を含む場合は、""で囲って指定してください。

-y

サイレントインストール後に自動的にシステムの再起動を行います。

復帰値

復帰値	意 味
0	正常終了
0以外	異常終了

コマンド格納場所

Windows x86	<ドライブ>¥MAIN¥tool¥sscmd
Windows x64(サーバ用)	<ドライブ>¥MAIN¥tool¥sscmd
Windows x64(クライアント用)	<ドライブ>¥Client¥tool¥sscmd
Solaris	<ドライブ>/Solaris/tool_unx/sscmd
Linux	<ドライブ>/Linux/tool_unx/sscmd
HP-UX AIX	<ドライブ>/tool_unx/sscmd

実行に必要な権限

- システム管理者(Administratorsグループに属するユーザ/スーパーユーザ)権限が必要です。
- Windows OS上で本コマンドを実行する場合、管理者権限で起動したコマンドプロンプト上で本コマンドを実行してください。

コマンド実行結果について

- 本コマンドはサイレントのため、コマンドのオプションエラーの場合は、以下のログに出力されます。

  【Windows版】

  ```
  Windowsディレクトリ¥swsilent.log
  ```

【Solaris版】

［サーバ］

```
/var/tmp/swsilent.log
```

［クライアント］

```
Windowsディレクトリ¥swsilent.log
```

【HP-UX版/AIX版/Linux版】

［サーバ］

```
/tmp/swsilent.log
```

［クライアント］

```
Windowsディレクトリ¥swsilent.log
```

- インストールの結果は、インストール情報ファイル作成時に設定したインストール結果出力先に、下記ファイル名で出力されます。必要に応じて参照してください。

  正常終了の場合：swinst.success
  異常終了の場合：swinst.err

注意事項

- 本コマンドは、サーバ、クライアントで実行できます。
- Windowsにインストールする場合、特定条件において、インストール完了時に自動的にSystemwalker Operation Managerが起動されます。

  以下のどれかの条件に当てはまる場合は、インストール完了時にSystemwalker Operation Managerは起動されません。

  - Systemwalker Centric Managerがインストールされている場合
  - 新規インストールでない場合
  - 製品がバンドルしているMicrosoft Visual C++ 再頒布可能パッケージよりも古いバージョンがインストールされており、再起動が必要な条件に一致した場合

記述例

指定したインストール情報ファイルでサイレントインストールを行い、インストール後に自動的にシステムの再起動を行います。

【Windows x86版】

```
f:¥MAIN¥tool¥sscmd¥swsilent.exe -i c:¥swinst.ini -y
```

【Solaris版】

```
/dev/cd0/Solaris/tool_unx/sscmd/swsilent.sh -i /work/swinst.ini -y
```

【Linux版】

```
/dev/cd0/Linux/tool_unx/sscmd/swsilent.sh -i /work/swinst.ini -y
```

1.6　jmmode 実行継続モード切り替えコマンド

記述形式

```
jmmode {continue|cancel|check} [-func {JOB|WRT}]
[-sys {all|サブシステム番号}](注)
```

注)

“[-sys { all | サブシステム番号}]”はSystemwalker Operation Manager EEのオプションです。

機能説明

以下に示す2つの高信頼機能に対して、必要に応じてそれぞれ有効にするかどうかを定義します。

- スケジュールサーバのシステムダウン時におけるネットワークジョブの実行継続。
- スケジュール情報ファイルの二重化(バックアップファイルへの書込み)。

オプション

{continue|cancel|check}

高信頼機能を有効にするかどうかを選択します。

continue

-funcオプションで指定された高信頼機能を有効にします。-funcオプションが省略されている場合は、すべての高信頼機能を有効にします。

cancel

-funcオプションで指定された高信頼機能を無効にします。-funcオプションが省略されている場合は、すべての高信頼機能を無効にします。

なお、初期値はJOB/WRT共にcancelとなります。

check

現在の高信頼機能の状態を表示します。本オプションを指定した場合、他のオプションの指定はすべて無視され、状態表示のみが行われます。

-func {JOB|WRT}

どの高信頼機能を有効または無効にするかを指定します。有効/無効の指定は、continueまたはcancelオプションで行います。本オプションを省略した場合、すべての高信頼機能が有効または無効となります。

JOB

ネットワークジョブの実行継続を指定します。

本状態は、スケジュールサーバと実行サーバにおいて統一する必要があります。本状態が統一されていない場合、ジョブは正しく起動されません。

continueオプションで有効が指定されていた場合は、スケジュールサーバのシステムダウン時にネットワークジョブの実行を継続します。

cancelオプションで無効が指定されていた場合は、スケジュールサーバのシステムダウン時にネットワークジョブを強制終了します。

WRT

スケジュール情報ファイルの二重化を指定します。

continueオプションで有効が指定されていた場合は、スケジュール情報ファイルを二重化し、スケジュール情報ファイルをオリジナルファイルおよびバックアップファイルに書き込みます。

cancelオプションで無効が指定されていた場合は、スケジュール情報ファイルのバックアップファイルへの書込みは行われません。ただし、以下の場合には、無効が指定されてもバックアップファイルへの書込みは行われます。

－ 登録情報(ジョブネット/グループ情報)を追加/更新/削除する場合

EE GEE **-sys {all|サブシステム番号}**

複数サブシステム運用をしているシステムで、操作対象のサブシステムを指定するためのオプションです。なお、このオプションを省略した場合、サブシステム番号0が操作対象となります。

all

すべてのサブシステムを操作対象とします。

サブシステム番号

操作対象とするサブシステム番号を、0から9の範囲で指定します。

復帰値

復帰値	意 味
0	正常終了
0以外	異常終了。エラーの内容はエラーメッセージを確認してください。

実行に必要な権限

システム管理者(Administratorsグループに属するユーザ/スーパーユーザ)権限が必要です。

表示例

checkオプションを指定した場合の表示例を示します。有効な場合はon、無効な場合はoffと表示されます。

```
JOB on
WRT off
```

EE GEE EE版の場合の表示例を以下に示します。

```
SYS 0
JOB on
WRT off

SYS 1
JOB off
WRT off
```

注意事項

“continue”、“cancel”オプションの指定時は、サービス/デーモンを停止させておく必要があります。[Systemwalker Operation Manager 環境設定]ウィンドウより、以下のサービスを停止、および起動させてください。

- ジョブスケジューラ
- ジョブ実行制御

1.7 mpbko バックアップコマンド

記述形式

【Windows版】

```
mpbko ドライブ名:ディレクトリ名 [/Ca | /Cs] [/d] [/SN | /p]
```

【UNIX版】

```
mpbko -b Directory [-d] [-SN | -p]
```

機能説明

Systemwalker Operation Managerサーバ上の環境定義情報やジョブのスケジュール情報といった運用管理情報を、指定したディレクトリにバックアップします。

mpbkoコマンドは、Systemwalker Operation Managerのサービス/デーモンを停止してバックアップを行い、バックアップ後にサービス/デーモンを起動します。また、Systemwalker Operation Managerが停止状態でのバックアップの場合も、バックアップ後にSystemwalker Operation Managerのサービス/デーモンを起動します。

オプション

ドライブ名:ディレクトリ名【Windows版】

バックアップ先を指定します。ドライブ名:ディレクトリ名は、半角英数字32文字以内で指定します。空白を含むことはできません。

ドライブ名

書込み可能なバックアップ先のハードディスクのドライブ名を指定します。

ディレクトリ名

バックアップ先のディレクトリを指定します。

-b *Directory*【UNIX版】

バックアップ先のディレクトリ名をフルパスで指定します。指定したディレクトリが存在しない場合は、自動的に作成されます。バックアップ先のディレクトリ名は、半角英数字で指定し、空白を含むことはできません。

EE GEE

/Ca | /Cs【Windows版】

本オプションは、クラスタシステムでバックアップするときに使用します。クラスタ運用でない通常のシステムでバックアップする場合は本オプションを省略します。本オプションは、“ドライブ名:ディレクトリ名”よりも後に指定します。

/Ca

本オプションは、運用中のノードでバックアップするときに指定します。運用中のノード上の情報と共有ディスク上の情報がバックアップされます。

なお、1:1運用待機(サブシステムあり、一部クラスタ運用)の場合は、運用系、待機系にかかわらず、必ず指定してください。

/Cs

本オプションは、待機中のノードでバックアップするときに指定します。待機中のノード上の情報がバックアップされます。

/d【Windows版】 -d【UNIX版】

バックアップ時に確認メッセージを一切出力せずに、サイレントモードで資産をバックアップさせる場合に指定します。本オペランド/オプションを指定する場合は、バックアップ先ディレクトリを空にしておく必要があります。バックアップ先ディレクトリにファイルが存在していた場合、コマンドは、エラー終了します。

/SN | /p 【Windows版】
-SN | -p 【UNIX版】

バックアップ時のSystemwalker Operation Managerのサービス/デーモンの起動/停止状態を制御するためのオプションです。

/SN -SN

Systemwalker Operation Managerのサービス/デーモンを停止せずにバックアップするときに指定します。

/p -p

バックアップ後にSystemwalker Operation Managerのサービス/デーモンを起動したくない場合に指定します。

復帰値

復帰値	意 味
0	正常終了
1	異常終了

コマンド格納場所

Windows	Systemwalker Operation Managerインストールディレクトリ¥MpWalker.JM¥MpCmtool¥backup¥bin
Solaris HP-UX AIX Linux	/opt/systemwalker/bin

実行に必要な権限

システム管理者(Administratorsグループに属するユーザ/スーパーユーザ)権限が必要です。

注意事項

- 本コマンドはフルパスで使用してください。
- 本コマンドを/SNまたは-SNオプションなしで実行すると、起動中のサービスが停止します。Systemwalker Centric Managerと共存する環境の場合は、Systemwalker Centric Managerのサービス/デーモンも同時に停止します。
- Windows版の場合、操作ミス・コマンド入力ミスなどを避けるため、運用環境保守ウィザードの運用環境の退避でバックアップを行うことを推奨します。手順の詳細については、“Systemwalker Operation Manager 導入ガイド”の“運用環境をバックアップ・リストアする”を参照してください。
  ただし、運用環境保守ウィザードの運用環境の退避でバックアップを行う場合、バックアップ後に起動しない指定ができません。バックアップ後に起動したくない場合は、本コマンドの/pオプション指定によりバックアップを行ってください。コマンド実行する場合は、操作ミス・コマンド入力ミスなどに注意してください。
  【Windows版】
- 本コマンドを実行する前に、必ず以下を確認してください。
  - バックアップ先のディレクトリにアクセス権があるか。
  - バックアップ先のディレクトリに十分な容量があるか。必要な容量は、“バックアップ容量”を参考にしてください。
- バックアップが正常に行われなかった場合は、以下を確認して必要な対処をしたうえで、もう一度mpbkoコマンドを実行してください。
  - バックアップ先のディレクトリにアクセス権があるか。

ー バックアップ先のディレクトリに十分な容量があるか。必要な容量は、“バックアップ容量”を参考にしてください。

- バックアップは、ジョブ/ジョブネット/グループが実行していない時間帯に行うことを推奨します。
- ジョブ/ジョブネット/グループが実行中にバックアップを行い、バックアップが正常に行われなかった場合（エラーメッセージが出力された場合）は、もう一度mpbkoコマンドを実行してください。
- ジョブ/ジョブネット/グループの実行中にバックアップを行い、その情報をリストアした場合、実行中だったジョブ/ジョブネット/グループは、終了コード239で異常終了になります。
- ネットワークドライブ上にあるディレクトリ名を指定することはできません。

バックアップ容量

【Windows版】

バックアップ先に必要な容量は、環境や登録データ量により変わります。容量は、以下の値を基に見積もります。可変部の容量を算出し、固定部の容量に加算してください。

固定部

固定部は、586Kバイトです。

可変部

可変部は以下のとおりです。

ー カレンダ機能を使用している場合

- カレンダ数×6Kバイト

ー イベント監視機能を使用している場合

- 定義イベント行×2Kバイト

  定義イベント行の行数は、[イベント監視の条件定義]ウィンドウに表示される一覧の行数と同じです。

ー ジョブスケジューラを使用している場合

- 登録プロジェクト数×1.5Kバイト
- ジョブネット数×8Kバイト
- ジョブ数×2Kバイト
- グループ数×2.4Kバイト
- グループ内ジョブネット数×12Kバイト
- 各種出口総容量
- 各ジョブの出力(stdout/stderr)容量の総和
- log指定サイズ×3バイト
- 監視ホスト定義数×2Kバイト＋(ホスト数＋フォルダ数）×1Kバイト

ー ジョブ実行制御を使用している場合

- ジョブフォルダ数×1Kバイト
- ジョブ所有者数×0.5Kバイト
- キューの数×4Kバイト
- システムに存在し得る最大ジョブ数×40Kバイト
- debug.logファイルの大きさ(不定)

【UNIX版】

バックアップ先に必要な容量については、以下の式を参考にしてください。

“/etc/opt/FJSV*”配下の容量の合計値 ＋“/var/opt/FJSV*”の配下の容量の合計値

FJSV*：製品に含まれるコンポーネントのパッケージ名

FJSV*はOSにより異なります。以下に読み替えて見積もってください。

Solaris版	：	FJSV*
Linux版	：	FJSV*
HP-UX版	：	FHP*
AIX版	：	FAIX*

記述例

【Windows版】

Systemwalker Operation Managerが“c:¥win32app”配下にインストールされている場合、Systemwalker Operation Managerサーバの環境定義情報やジョブのスケジュール情報といった運用管理情報やログ情報を、“d:¥tmp¥OMGRBack”ディレクトリにバックアップするときの指定例を以下に示します。

```
cd c:¥win32app¥MpWalker.JM¥MpCmtool¥backup¥bin
mpbko d:¥tmp¥OMGRBack
```

【UNIX版】

バックアップ先ディレクトリを“/backup”にして、バックアップするときの指定例を以下に示します。

```
/opt/systemwalker/bin/mpbko -b /backup
```

参考

ジョブとして登録する

mpbkoコマンドをジョブとしてスケジュールすることができます。ジョブとして登録する場合は、必ず以下を行ってください。

- コマンドに/SNまたは-SNオプションを指定し、バックアップ時にサービス/デーモンが停止しないようにする
- コマンドに/dまたは-dオプションを指定し、バックアップ時に確認メッセージを一切出力しないようにする
- バックアップを行う前に、バックアップ先のディレクトリが空になっていることを確認する

1.8　mprso リストアコマンド

記述形式

【Windows版】

```
mprso ドライブ名:ディレクトリ名 [/c] [/Ln | /Lo | /Le] [/Ca | /Cs]
```

【UNIX版】

```
mprso -b Directory [-c]
```

機能説明

mpbkoコマンドでバックアップした、Systemwalker Operation Managerサーバの環境定義情報や運用管理情報を、同じバージョンレベルのSystemwalker Operation Managerにリストアします。

オプション

ドライブ名:ディレクトリ名【Windows版】

mpbkoコマンドによってバックアップされた情報の格納先を指定します。ドライブ名:ディレクトリ名は、半角英数字32文字以内で指定します。空白を含むことはできません。

ドライブ名

mpbkoコマンドによってバックアップされた情報の格納先ハードディスクのドライブ名を指定します。

ディレクトリ名

mpbkoコマンドによってバックアップされた情報の格納先のディレクトリ名を指定します。

-b *Directory* 【UNIX版】

資産がバックアップされているディレクトリ名をフルパスで指定します。ディレクトリ名は、半角英数字で指定し、空白を含むことはできません。

/c 【Windows版】

本オプションは、バックアップ時の環境定義情報や運用管理情報に戻すため、これらを初期化してからリストアする場合に指定します。

/Lnと同時に指定した場合、定義情報を初期化してからリストアします。ログ情報は初期化もリストアもされません。
/Loと同時に指定した場合、ログ情報を初期化してからリストアします。定義情報は初期化もリストアもされません。
それ以外は、ログ情報と定義情報の両方を初期化してからリストアします。

本オプションは、“ドライブ名:ディレクトリ名”よりも後に指定します。

-c 【UNIX版】

本オプションは、バックアップ時の環境定義情報や運用管理情報に戻すため、これらを初期化してからリストアする場合に指定します。

/Ln|/Lo|/Le 【Windows版】

本オプションは、ログ情報と定義情報のどちらをリストアするかを選択するときに指定します。本オプションを省略すると、/Leが指定されたものとみなされます。本オプションは、“ドライブ名:ディレクトリ名”よりも後に指定します。

/Ln

定義情報だけをリストアします。ログ情報はリストアされません。

/Lo

ログ情報だけをリストアします。定義情報はリストアされません。

/Le

ログ情報と定義情報の両方をリストアします。

EE GEE

/Ca|/Cs 【Windows版】

本オプションは、クラスタシステムでリストアするときに使用します。クラスタ運用でない通常のシステムでリストアする場合は本オプションを省略します。本オプションは、/cオプション、/Lnオプション、/Loオプションおよび/Leオプションよりも後に指定します。

/Ca

本オプションは、運用中のノードでリストアするときに指定します。運用中のノードと共有ディスクに情報がリストアされます。mpbkoコマンドで/Csオプションを指定して採取したバックアップをリストアするときは、本オプションは指定できません。

なお、1:1運用待機(サブシステムあり、一部クラスタ運用)の場合は、運用系、待機系にかかわらず、必ず指定してください。

/Cs

本オプションは、待機中のノードでリストアするときに指定します。待機中のノードに情報がリストアされます。共有ディスクには情報はリストアされません。

復帰値

復帰値	意 味
0	正常終了
1	異常終了

参照

mpbko

コマンド格納場所

Windows	Systemwalker Operation Managerインストールディレクトリ¥MpWalker.JM¥MpCmtool¥backup¥bin
Solaris HP-UX AIX Linux	/opt/systemwalker/bin

実行に必要な権限

システム管理者(Administratorsグループに属するユーザ/スーパーユーザ)権限が必要です。

注意事項

- 本コマンドはフルパスで使用してください。
- Windows版の場合、操作ミス・コマンド入力ミスなどを避けるため、運用環境保守ウィザードの運用環境の復元でリストアを行ってください。手順の詳細については、“Systemwalker Operation Manager 導入ガイド”の“運用環境をバックアップ・リストアする”を参照してください。
  ただし、運用環境保守ウィザードの運用環境の復元でリストアを行う場合、リストア先の環境定義情報や運用管理情報を削除する指定ができません。削除したい場合は、本コマンドの/cオプション指定によりリストアを行ってください。コマンド実行する場合は、操作ミス・コマンド入力ミスなどに注意してください。【Windows版】
- 本コマンドを実行すると起動中のサービス/デーモンが停止します。Systemwalker Centric Managerと共存する環境で本コマンドを実行すると、Systemwalker Centric Managerのサービス/デーモンも同時に停止します。
- 本コマンドを実行する前に必ず以下を確認してください。
  - バックアップ先ディレクトリおよびリストア先ディレクトリへのアクセスが可能かどうか。
  - リストア先ディレクトリに十分な空き容量があるかどうか。
- リストアすることができるのは、本バージョンレベルのバックアップコマンドでバックアップされた資産に限ります。
- 異なるOS種別、インストール種別、コード系、インストールディレクトリのバックアップデータは、リストアできません。

・ ネットワークドライブ上にあるディレクトリ名を指定することはできません。

本コマンドをSystemwalker Operation Managerのジョブから実行した場合、Systemwalker Operation Manager/Systemwalker Centric Managerの動作は保証されません。詳細については、“Systemwalker Operation Manager 運用ガイド”の“ジョブとして実行してはいけないコマンド”を参照してください。

・ /c【Windows版】および -c【Linux版】オプション指定により初期化される情報は、以下のとおりです。

ー カレンダ/電源制御の定義ファイル

“Systemwalker Operation Manager 導入ガイド”の“カレンダおよび電源制御に関する情報”を参照してください。

ー プロジェクト情報、ジョブネット情報、グループ情報、起動日雛形情報、ジョブのパラメタ情報、ジョブネット変数情報、ログ情報、ジョブ定義変数情報

“Systemwalker Operation Manager 導入ガイド”の“ジョブスケジューラに関する情報”を参照してください。

ー プロジェクト制御文(登録済)

“Systemwalker Operation Manager 導入ガイド”の“マスタスケジュール管理機能に関する情報”を参照してください。

EE GEE ・ リストア先の環境がバックアップ時の環境と同じでない場合は、同じ環境にしてください。

リストアする場合には、以下がバックアップ時の環境と同じである必要があります。

ー 複数サブシステム環境の構築（サブシステムの構成）

ー クラスタ環境の構築

ー マスタスケジュール管理機能の構築

これらの環境構築がされていた場合は、リストア先でもリストア前に同じ環境にしておく必要があります。

記述例

【Windows版】

Systemwalker Operation Managerが“c:¥win32app”配下にインストールされている場合、“d:¥tmp¥OMGRBack”にバックアップされているSystemwalker Operation Managerの環境定義情報、運用管理情報およびログ情報をリストアするときの指定例を以下に示します。

```
cd c:¥win32app¥MpWalker.JM¥MpCmtool¥backup¥bin
mprso d:¥tmp¥OMGRBack
```

【UNIX版】

“/backup”にバックアップされているサーバ資産をリストアするときの指定例を以下に示します。

```
/opt/systemwalker/bin/mprso -b /backup
```

1.9 omgrmonitor 監視ホスト情報取込みコマンド

記述形式

【Windows版】

```
omgrmonitor {-export|-get|-import} file_name [-name 監視ホスト定義名]
```

【UNIX版】

```
omgrmonitor {-export|-import} file_name [-name 監視ホスト定義名]
```

機能説明

Systemwalker Operation Managerの監視ホスト情報をCSV形式のファイルに出力します。

また、CSV形式のファイルをSystemwalker Operation Managerの監視ホスト情報として取り込むことができます。

なお、本コマンドでは、サブシステムの情報は扱えません。

Systemwalker Centric Managerの運用管理サーバが導入されている場合は、Systemwalker Centric Managerの構成情報データベースに登録されている基本ツリー情報をCSV形式のファイルに出力することができます。【Windows版】

オプション

-export *file_name*

監視ホスト情報を、指定されたファイル名にCSV形式で出力します。

*file_name*にはSJISのコードで出力されます。【UNIX版】

-get *file_name* 【Windows版】

Systemwalker Centric Managerの構成情報データベースに登録されている基本ツリー情報を、指定されたファイル名にCSV形式で出力します。

-import *file_name*

指定されたCSV形式のファイルを、Systemwalker Operation Managerの監視ホスト情報として取り込みます。

*file_name*は、SJISのコードで記述してある必要があります。【UNIX版】

-name *監視ホスト定義名*

出力または取込み対象とする監視ホスト定義名を指定します。省略した場合には、“DEFAULT”の情報を出力または取込み対象とします。

注意事項

- Systemwalker Centric Managerの構成情報データベースに登録されている基本ツリー情報を、-getオプションを指定してCSV形式に出力する場合は、Systemwalker Centric Managerの運用管理サーバ上で本コマンドを実行してください。【Windows版】

基本ツリー情報の出力例

```
フォルダ番号,先行フォルダ番号,フォルダの表示名,ホストの表示名,ホスト名,IPアドレス
1,,"root",,,
2,1,"astroNT",,,
3,2,"自部門1",,,
4,3,"10.11.11.0",,,
,4,,"運用管理サーバ","astroNT",10.11.111.111
5,1,"自部門2",,,
6,5,"10.111.111.0",,,
,6,,"111","111",10.111.111.111
,6,,"222","222",10.111.111.112
,6,,"333","333",10.111.111.113
7,5,"10.111.222.0",,,
```

上記のファイルを表計算ソフトで表示した例を、次に示します。

基本ツリー情報の表示例

	フォルダ番号	先行フォルダ番号	フォルダの表示名	ホストの表示名	ホスト名	IPアドレス
2	1		root			
3	2	1	astroNT			
4	3	2	自部門１			
5	4	3	10.11.11.0			
6		4		運用管理サーバ	astroNT	10.11.11.111
7	5	1	自部門２			
8	6	5	10.111.111.0			
9		6		111	111	10.111.111.111
10		6		222	222	10.111.111.112
11		6		333	333	10.111.111.113
12	7	5	10.111.222.0			

CSV形式のファイルで監視ホスト情報を作成する場合の注意事項

1行目のタイトル行は必須項目です。変更しないでください。

2行目のrootフォルダは必須項目です。変更しないでください。

3行目以降は、以下の手順に従って記述してください。

フォルダを指定する場合

1. フォルダ番号を指定します。他のフォルダ番号と重複しないように注意してください。
2. 先行フォルダ番号を指定します。
3. フォルダの表示名を指定します。

フォルダ番号、先行フォルダ番号およびフォルダの表示名の指定は必須です。

その他の情報は指定しないでください。

ホストを指定する場合

1. 先行フォルダ番号を指定します。
2. ホストの表示名(接続ホスト名コンボボックスでの表示名)を指定します。128バイト以内で指定します。
3. ホスト名(DNS名)を指定します。“ホスト名”または“ホスト名.ドメイン名”の形式で指定します。
4. IPアドレスを指定します。

先行フォルダ番号、ホストの表示名、ホスト名およびIPアドレスは必須です。

その他の情報は指定しないでください。

注意

CSV形式のファイルを表計算ソフトで読み込む時、数字だけの文字列が以下のように変換される場合があります。

CSVファイル上での表記:0010 → 表計算ソフトでの表記:10

これは、表計算ソフトが数字だけの文字列を、文字列ではなく数値データとして誤って認識するためにおこる現象です。このような場合は、セルの書式設定により該当セルが文字列として扱われるように変更したのち、CSVファイル上での表記にあわせて文字列を訂正してください。

1.10 createsubsystem.exe/sh サブシステム環境作成コマンド

記述形式

【Windows版】

```
createsubsystem.exe -sys サブシステム番号
```

【UNIX版】

```
createsubsystem.sh -sys サブシステム番号
```

機能説明

サブシステム環境を作成します。

オプション

-sys サブシステム番号

作成するサブシステム番号を1から9の範囲で指定します。

復帰値

【Windows版】

復帰値	意 味
0	正常終了
1	引数エラーまたは権限エラーで失敗しました。
2	ジョブスケジューラでサブシステム環境の作成に失敗しました。
3	ジョブ実行制御でサブシステム環境の作成に失敗しました。
4	ジョブスケジューラとジョブ実行制御の環境作成に失敗しました。

【UNIX版】

復帰値	意 味
0	正常終了
1	引数エラーまたは権限エラーで失敗しました。

参照

【Windows版】

deletesubsystem.exe

コマンド格納場所

Windows	Systemwalker Operation Managerインストールディレクトリ¥MpWalker.JM¥bin
Solaris Linux	/opt/FJSVJMCMN/bin/
HP-UX	/opt/FHPJMCMN/bin/
AIX	/opt/FAIXJMCMN/bin/

実行に必要な権限

Administratorsグループに所属したユーザで、“オペレーティングシステムの一部として機能”の権利を有するユーザ/スーパーユーザが必要です。

使用例

【Windows版】

```
createsubsystem.exe -sys 1
```

EE

1.11 deletesubsystem.exe サブシステム環境削除コマンド【Windows版】

記述形式

```
deletesubsystem.exe -sys サブシステム番号
```

機能説明

サブシステム環境を削除します。

オプション

-sys サブシステム番号

削除するサブシステム番号を1から9の範囲で指定します。

復帰値

復帰値	意 味
0	正常終了
1	引数エラーまたは権限エラーで失敗しました。
2	ジョブスケジューラでサブシステム環境の作成に失敗しました。
3	ジョブ実行制御でサブシステム環境の作成に失敗しました。
4	ジョブスケジューラとジョブ実行制御の環境作成に失敗しました。

参照

createsubsystem.exe

コマンド格納場所

Windows	Systemwalker Operation Managerインストールディレクトリ¥MpWalker.JM¥bin

実行に必要な権限

システム管理者(Administratorsグループに属するユーザ)権限が必要です。

注意事項

本コマンドは、削除するサブシステムに対応するジョブスケジューラおよびジョブ実行制御のサービスを停止した後に実行してください。

使用例

```
deletesubsystem.exe -sys 1
```

1.12 swmove 移行用退避コマンド【UNIX版】

記述形式

```
swmove -b ディレクトリ
```

機能説明

旧版のSystemwalker Operation Managerの資産を本バージョンへ移行するために、環境を退避します。移行対象となる資産が指定された退避先ディレクトリに退避されます。

オプション

-b ディレクトリ

退避先のディレクトリ名をフルパスで指定します。指定したディレクトリが存在しない場合は、自動的に作成されます。ディレクトリ名は、半角英数字で指定し、空白を含むことはできません。

復帰値

復帰値	意 味
0	正常終了
1	異常終了

参照

rmpackage.inf

Systemwalker Operation Managerを移行する時に、削除が必要な旧パッケージ名の一覧を格納するファイルです。移行の途中に行われる削除が失敗した場合は、利用者がこのファイルを参照してパッケージを削除します。このファイルは、-bオプションで指定した退避先ディレクトリの直下に作成されます。

コマンド格納場所

Solaris	<ドライブ>/マウントポイント/Solaris/unx/tool
Linux	<ドライブ>/マウントポイント/Linux/unx/tool
HP-UX AIX	<ドライブ>/マウントポイント/unx/tool

実行に必要な権限

システム管理者(スーパーユーザ)権限が必要です。

注意事項

- 必ず“Systemwalker Operation Manager 移行ガイド”に記載されている手順に従って資産の退避を行うようにしてください。
- 本コマンドを実行する前に必ず以下を確認してください。
  - 退避先ディレクトリへの書込みが可能かどうか。
  - 退避先ディレクトリに十分な空き容量があるかどうか。

エラー時の対処

容量不足時には、不要なファイルを削除または別ディレクトリを指定するなど、利用者の手で十分な空き領域を確保してコマンドを再実行してください。旧パッケージの削除途中でエラーが発生した場合も、コマンドを再実行してください。また、rmpackage.infを見て利用者の責任で旧パッケージを削除することもできます。

- ネットワークドライブ上にあるディレクトリ名を指定することはできません。
- 本コマンドを実行すると、Systemwalker Operation Managerが自動的に停止されます。また、Systemwalker Centric Managerと共存している場合、Systemwalker Centric Managerの各デーモンも自動的に停止されます。

使用例

“/work/move”へ移行データを退避します。

【Solaris版】

```
<ドライブ>/マウントポイント/Solaris/unx/tool/swmove -b /work/move
```

【Linux版】

```
<ドライブ>/マウントポイント/Linux/unx/tool/swmove -b /work/move
```

1.13 swtrans 移行用変換・登録コマンド【UNIX版】

記述形式

```
swtrans -b ディレクトリ
```

機能説明

swmoveコマンドで退避した旧版のSystemwalker Operation Managerの資産を本バージョンへ移行します。コマンドを実行すると、指定された退避先ディレクトリにある移行資産が復元されます。

オプション

-b ディレクトリ

復元する移行資産が格納(退避)されているディレクトリ名をフルパスで指定します。ディレクトリ名は、半角英数字で指定し、空白を含むことはできません。

復帰値

復帰値	意 味
0	正常終了
1	異常終了

コマンド格納場所

Solaris	<ドライブ>/マウントポイント/Solaris/unx/tool
Linux	<ドライブ>/マウントポイント/Linux/unx/tool
HP-UX AIX	<ドライブ>/マウントポイント/unx/tool

実行に必要な権限

システム管理者(スーパーユーザ)権限が必要です。

注意事項

- 必ず“Systemwalker Operation Manager 移行ガイド”に記載されている手順に従って資産の移行を行うようにしてください。
- 退避元マシンと復元先マシンのOSの種別、インストール種別、インストールディレクトリおよびコード系は、同じでなければなりません。
- 本コマンドを実行する前に必ず以下を確認してください。
  - 退避先ディレクトリおよびインストールディレクトリへのアクセスが可能かどうか。
  - 復元先マシンのインストールディレクトリに十分な空き容量があるかどうか。
- ネットワークドライブ上にあるディレクトリ名を指定することはできません。
- 本コマンドを実行すると、Systemwalker Operation Managerが自動的に停止されます。また、Systemwalker Centric Managerと共存している場合、Systemwalker Centric Managerの各デーモンも自動的に停止されます。
- 本コマンドを実行後、以下のメッセージが出力されます。

  ```
  May I shift by this backup data? [y or n](Default : n)
  ```

  指定した資産の復元を実施してもよい場合には、“y”を指定してください。

使用例

“/work/move”に退避された移行データを復元します。

【Solaris版】

```
<ドライブ>/マウントポイント/Solaris/unx/tool/swtrans -b /work/move
```

【Linux版】

```
<ドライブ>/マウントポイント/Linux/unx/tool/swtrans -b /work/move
```

1.14 mpsetcluster クラスタ情報の登録/削除コマンド

記述形式

【Windows版】

```
mpsetcluster.exe
```

【UNIX版】

```
mpsetcluster -c|-d
```

機能説明

Systemwalker Operation Managerのクラスタ情報を登録または削除します。

オプション

-c【UNIX版】

クラスタ情報を登録します。

-d【UNIX版】

クラスタ情報を削除します。

コマンド格納場所

Windows	Systemwalker Operation Managerインストールディレクトリ ¥mpwalker.jm¥mpcmtool¥bin
Solaris HP-UX AIX Linux	/opt/systemwalker/bin

実行に必要な権限

システム管理者(Administratorsグループに属するユーザ/スーパーユーザ)権限が必要です。

注意事項

- 本コマンドでクラスタ情報を登録していない場合、バックアップ・リストアコマンドは共有ディスクの情報を参照できないため正常に動作しません。
- クラスタ情報を登録したマシンでSystemwalker Operation Managerをアンインストールする場合、本コマンドでクラスタ情報を削除してからアンインストールしてください。

使用例

【Windows版】

```
mpsetcluster.exe
```

本コマンドを実行すると、[クラスタ情報の登録/削除]ウィンドウが表示されます。



[登録]:

クラスタ情報を登録します。

[削除]:

クラスタ情報を削除します。

【UNIX版】

クラスタ情報を登録します。

```
mpsetcluster -c
```

クラスタ情報を削除します。

```
mpsetcluster -d
```

1.15　swpkginfo 製品情報表示コマンド(サーバ用)【UNIX版】

記述形式

```
swpkginfo [-l]
```

機能説明

本コマンドは、インストールされているSystemwalker Centric ManagerおよびSystemwalker Operation Managerの製品情報を簡単な方法で確認します。

Linux版およびSolaris版の場合は、インストールに使用した媒体が32bitか64bitかを確認することができます。

オプション

-l

インストール済みパッケージの一覧を出力するオプションです。

コマンド格納場所

Solaris HP-UX AIX Linux	/opt/FJSVftlo/bin/swpkginfo

注意事項

誤ったオプションが指定された場合、コマンドの説明(USAGE)が出力されます。

使用例

```
/opt/FJSVftlo/bin/swpkginfo
```

実行結果

オプションを省略した場合

```
***********************************************
Systemwalker Operation Manager Information
***********************************************
PRODUCTNAME   : Systemwalker Operation Manager Enterprise Edition (32bit)
VERSION       : Vxx.x.x
VENDER        : COPYRIGHT FUJITSU LIMITED 1995-20XX
CODE          : UTF-8
INSTALLDATE   : 20XX.XX.XX
PLATFORM      : Linux
INSTALLCLASS  : Server
INSTALLPATH   : /opt
                /etc/opt
                /var/opt
```

-lを指定した場合

```
***********************************************
Systemwalker Operation Manager Information
***********************************************
PRODUCTNAME   : Systemwalker Operation Manager Enterprise Edition (32bit)
VERSION       : Vxx.x.x
VENDER        : COPYRIGHT FUJITSU LIMITED 1995-20XX
CODE          : UTF-8
INSTALLDATE   : 20XX.XX.XX
PLATFORM      : Linux
INSTALLCLASS  : Server
INSTALLPATH   : /opt
                /etc/opt
                /var/opt

PACKAGES      : Install of Packages as follows.
                FJSVsclr
                 Securecrypto Library RunTime
                FJSVsmee
                 S/MIME & EE Certificate Management Package
                FJSVswaic
                 Systemwalker Authentication and ID Management Client
                FJSVftlo
                 Systemwalker Operation Manager - FrameWork Tool Operation Manager
                FJSVfwtco
                 Systemwalker Operation Manager - Script Base
                FJSVJMCMN
                 Systemwalker Operation Manager - Operation Manager Common
                FJSVfwseo
                 Systemwalker Operation Manager - ACL Manager
                FJSVJOBSC
                 Systemwalker Operation Manager - JOB(command/shell script) Scheduling
                FJSVMJS
                 Job Execution Control Server
                FJSVsnjss
                 Systemwalker Operation Manager - NeTJobServe for Operation Manager Server
                FJSVjmcal
                 Systemwalker Operation Manager - Calendar Services
                FJSVstem
                 Systemwalker Operation Manager - Solution Template
                FJSVjmweb
                 Systemwalker Operation Manager - Web Console
                FJSVjmman
                 Systemwalker Operation Manager - Web Manual
```

1.16 F3crfver 製品情報表示コマンド

記述形式

```
F3crfver.exe [/OPER | -f実行結果出力先ファイル名]
```

機能説明

本コマンドは、インストールされているSystemwalker Operation Managerの製品情報を簡単な方法で確認します。

オプション

指定なし

Systemwalker Operation Managerの製品情報の概要をダイアログに表示します。

Systemwalker Centric Managerがインストールされていた場合は、Systemwalker Centric Managerの製品情報の概要も同時に表示します。

/OPER

Systemwalker Operation Managerの製品情報の詳細をダイアログに表示します。

-f実行結果出力先ファイル名

製品情報の概要をファイルに出力する場合に指定します。

実行結果出力ファイル名は、“-f”とファイル名の間に空白を入れず、フルパスで指定してください。

存在しないディレクトリを含むパスは指定しないでください。

コマンド格納場所

Windows	Systemwalker Operation Managerインストールディレクトリ ¥mpwalker.jm¥mpcmtool¥bin¥F3crfver.exe

注意事項

インストール種別がクライアントの場合は、エディションを表示しません。

使用例

Systemwalker Operation Managerの製品情報の詳細を表示する場合

```
Systemwalker Operation Managerインストールディレクトリ
¥mpwalker.jm¥mpcmtool¥bin¥F3crfver.exe /OPER
```

実行結果/出力形式

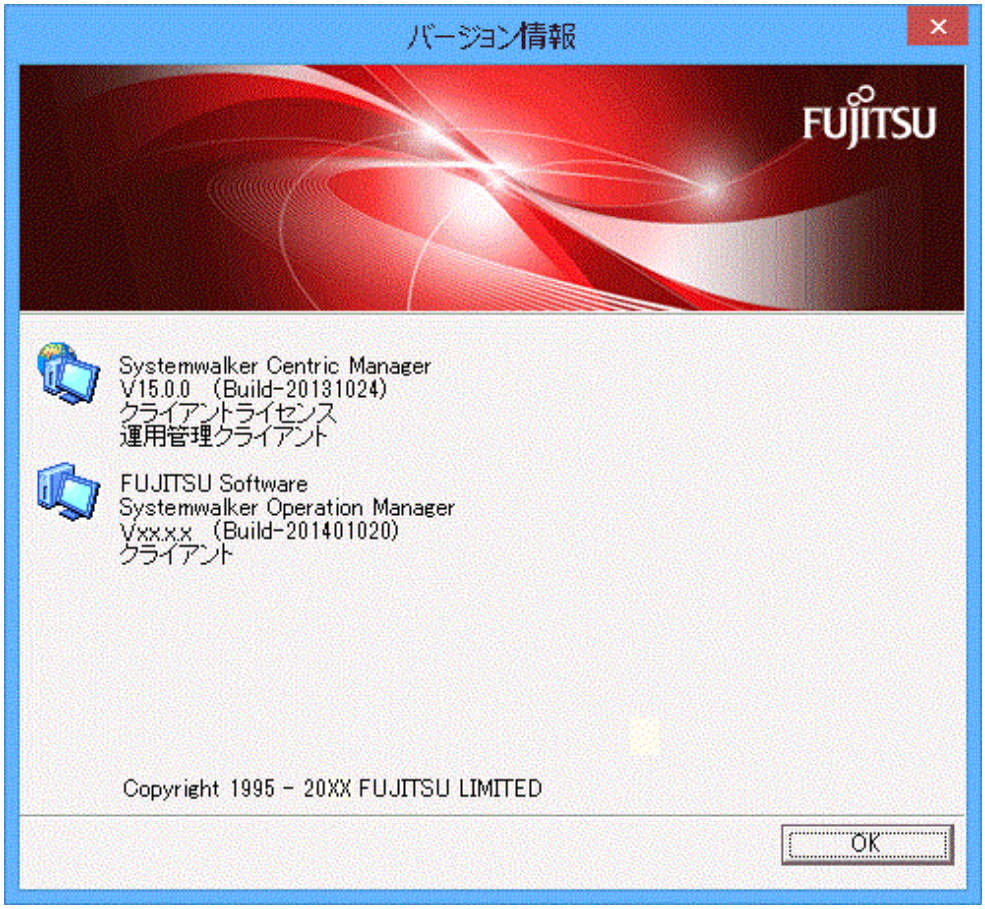
本コマンドを実行した場合の実行結果を以下に示します。

・ オプションを省略した場合

【Windows版】

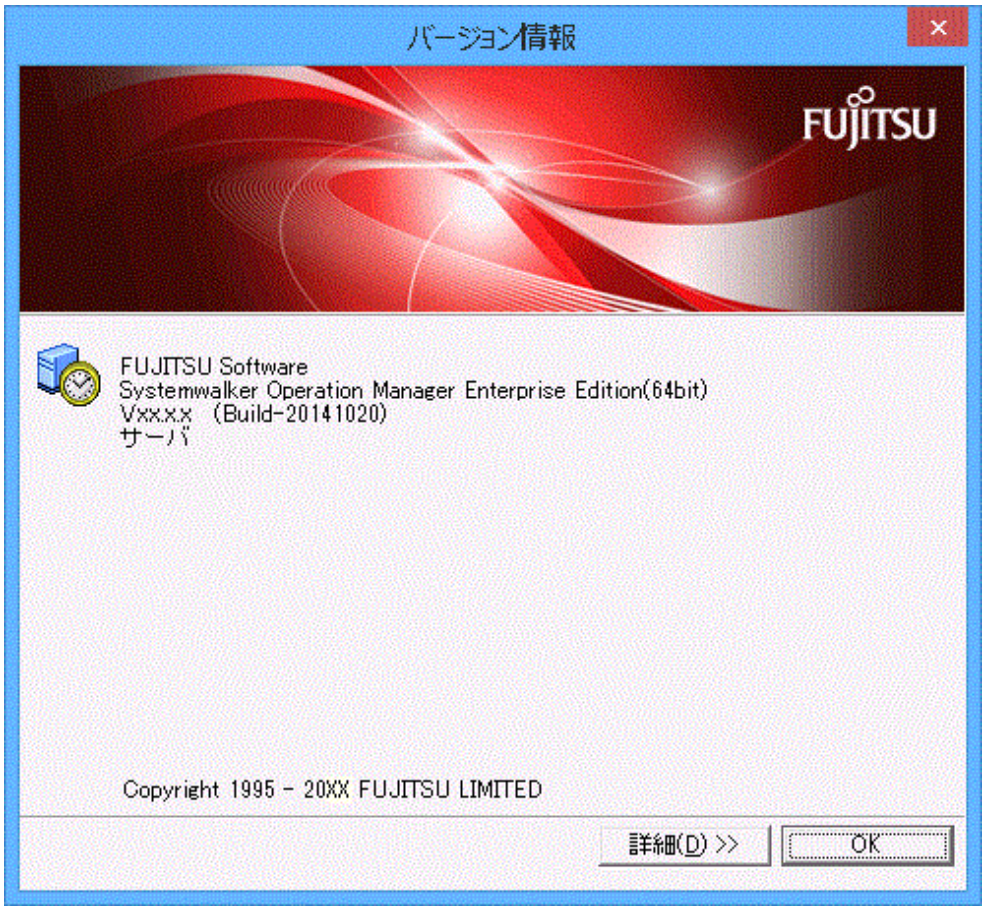


【UNIX版】

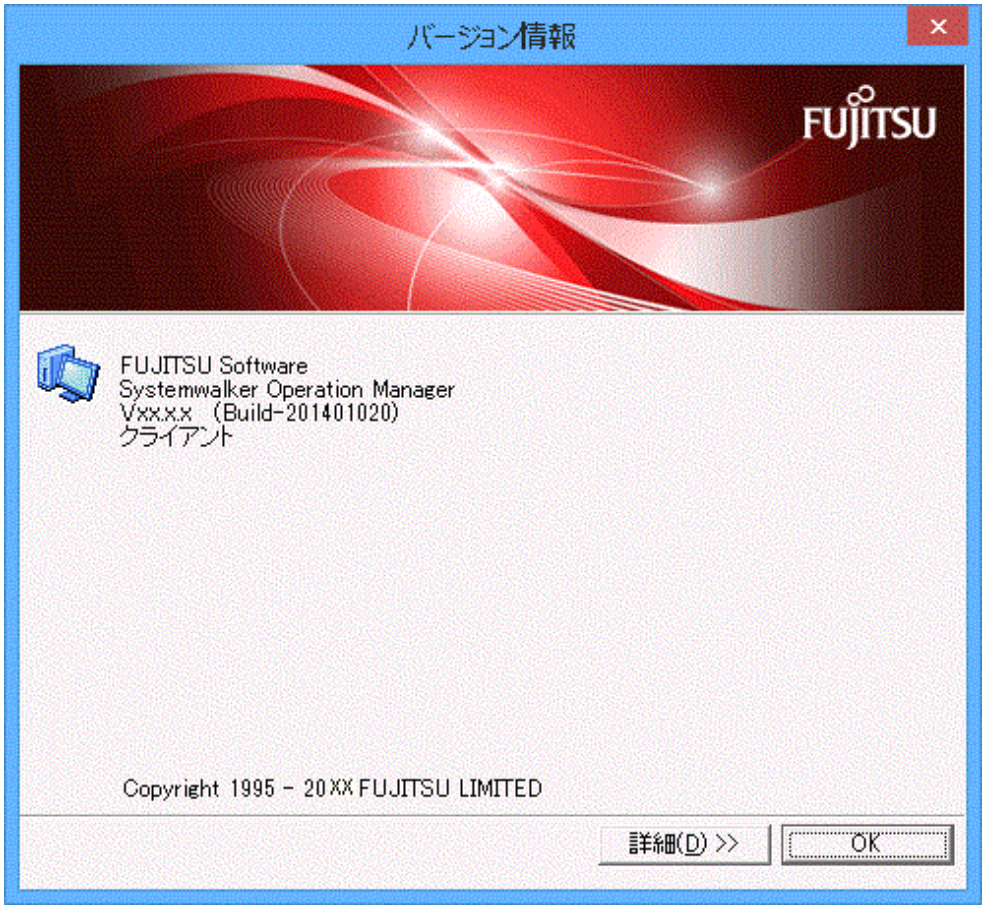

バージョン情報
FUJITSU
Systemwalker Centric Manager
V15.0.0 (Build-20131024)
クライアントライセンス
運用管理クライアント
FUJITSU Software
Systemwalker Operation Manager
Vxx.x.x (Build-201401020)
クライアント
Copyright 1995 - 20XX FUJITSU LIMITED
OK

- /OPERを指定した場合

【Windows版】



【UNIX版】

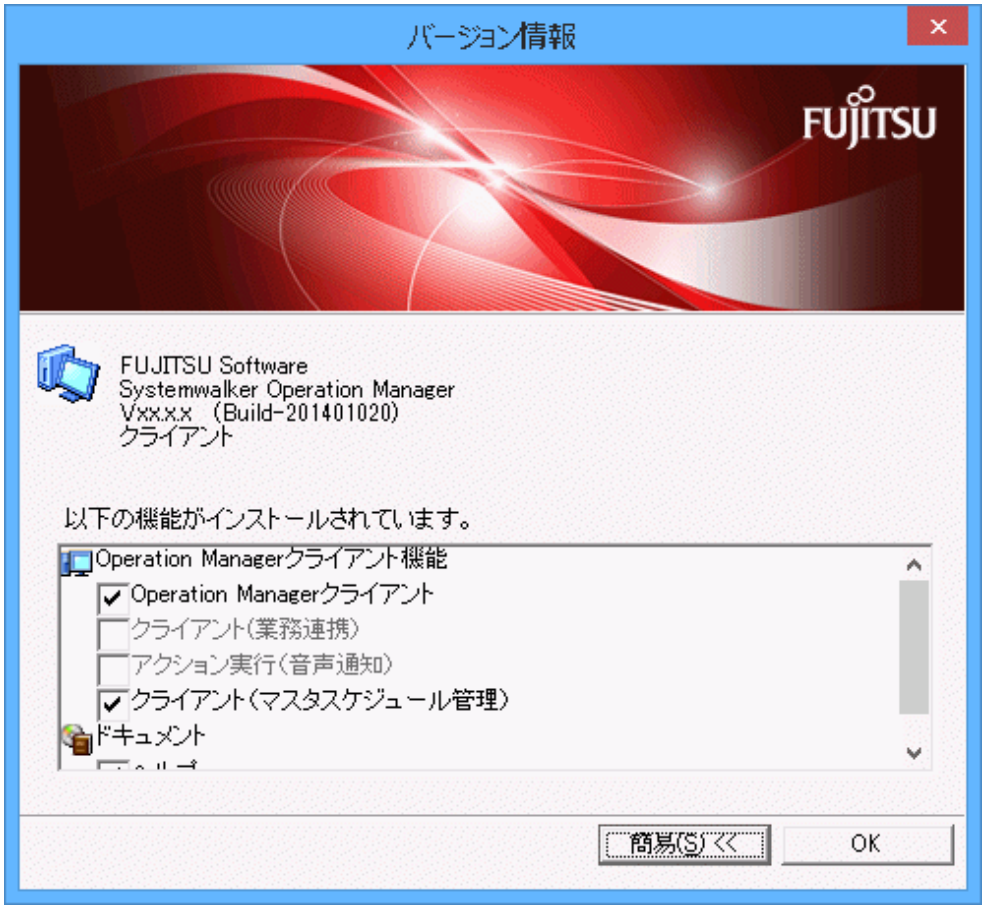


詳細ボタンをクリックすると、インストールされている機能が確認できます。

【Windows版】


バージョン情報
FUJITSU
FUJITSU Software
Systemwalker Operation Manager Enterprise Edition(64bit)
Vxx.x.x (Build-20141020)
サーバ
以下の機能がインストールされています。
Operation Managerサーバ機能
サーバ(基本)
Operation Managerクライアント機能
Operation Managerクライアント
クライアント(業務連携)
アクション実行(音声通知)
簡易(S)<<
OK

【UNIX版】



- -fを指定した場合

【Windows版】

```
Windows版 Systemwalker Centric Manager Enterprise Edition(64bit)
V15.0.0 (Build-20131024)
エージェントライセンス
業務サーバ

FUJITSU Software
Systemwalker Operation Manager Enterprise Edition(64bit)
Vxx.x.x (Build-20141020)
サーバ
```

【UNIX版】

```
Systemwalker Centric Manager
V15.0.0 (Build-20131024)
クライアントライセンス
運用管理クライアント

FUJITSU Software
Systemwalker Operation Manager
Vxx.x.x (Build-20141020)
クライアント
```

インストール種別がクライアントの場合、エディションは表示されません。

1.17 setup Systemwalker Operation Managerインストールコマンド【Windows版】

記述形式

setup.exe

機能説明

Systemwalker Operation Managerのインストーラを起動します。コマンドを実行すると[Systemwalker セットアップ]画面が表示されます。画面に従って各種設定をすることでSystemwalker Operation Managerのインストールを行うことができます。

オプション

なし

復帰値

コマンドを実行するとGUIが起動し、コマンドはすぐに復帰します。復帰値には任意の値が返りますが、エラーなどの判定には使用しないでください。

コマンド格納場所

Windows x86	<ドライブ>¥MAIN¥win32
Windows x64(サーバ用)	<ドライブ>¥MAIN¥win32
Windows x64(クライアント用)	<ドライブ>¥Client¥win32

実行に必要な権限

システム管理者(Administratorsグループに属するユーザ)権限が必要です。

注意事項

本コマンドでサイレントインストールはできません。サイレントインストールに関しては、“Systemwalker Operation Manager 導入ガイド”の“サイレントインストール”を参照してください。

使用例

【Windows x86版】

<ドライブ>¥MAIN¥win32¥setup.exe

1.18 swuset Systemwalker Operation Managerアンインストールコマンド【Windows版】

記述形式

swuset.exe

機能説明

Systemwalker Operation Managerをアンインストールします。

オプション

なし

復帰値

コマンドを実行するとGUIが起動し、コマンドはすぐに復帰します。復帰値には任意の値が返りますが、エラーなどの判定には使用しないでください。

コマンド格納場所

Windows	Systemwalker Operation Managerインストールディレクトリ¥mpwalker.jm¥jmunins

実行に必要な権限

システム管理者(Administratorsグループに属するユーザ)権限が必要です。

注意事項

注意事項などについては、“Systemwalker Operation Manager 導入ガイド”の、“アンインストール前の注意事項”、“アンインストール前の作業”および“Windowsからのアンインストール”を参照してください。

使用例

```
Systemwalker Operation Managerインストールディレクトリ¥mpwalker.jm¥jmunins¥swuset.exe
```

1.19 mpowebsetup Webサーバセットアップコマンド

記述形式

【Windows版】

```
mpowebsetup.bat [-u]
```

【UNIX版】

```
mpowebsetup [-u]
```

機能説明

本コマンドは、Webサーバ(WebコンソールおよびWeb API)をセットアップします。

オプション

-u

Webサーバをアンセットアップします。

復帰値

復帰値	意 味
0	正常終了
1以上	異常終了

コマンド格納場所

Windows	Systemwalker Operation Managerインストールディレクトリ¥mpwalker.jm¥mpjmweb¥bin
Solaris Linux	/opt/FJSVjmweb/bin

実行に必要な権限

システム管理者(Administratorsグループに属するユーザ/スーパーユーザ)権限が必要です。

注意事項

- 製品インストール直後など、すでにセットアップが実施されている状態で“mpowebsetup”が実行された場合、セットアップを行わずに正常終了します。また、アンセットアップが行われた状態で“mpowebsetup -u”が実行された場合、アンセットアップを行わずに正常終了します。
- 本コマンドの実行前に、poperationmgrコマンドでSystemwalker Operation Managerを停止してください。本コマンド実行後、soperationmgr コマンドでSystemwalker Operation Managerを起動すると、設定が反映されます。

使用例

Webサーバ(WebコンソールおよびWeb API)をセットアップします。

【Windows版】

```
mpowebsetup.bat
```

【UNIX版】

```
mpowebsetup
```

Webサーバ(WebコンソールおよびWeb API)をアンセットアップします。

【Windows版】

```
mpowebsetup.bat -u
```

【UNIX版】

```
mpowebsetup -u
```

実行結果/出力形式

正常終了時は、標準出力に以下のメッセージが出力されます。

```
mpowebsetup コマンドは正常に終了しました。
```

異常終了時は、標準エラー出力に以下のメッセージが出力されます。

```
Webサーバのセットアップに失敗しました。
```

1.20 mpowebconfig Webサーバ設定変更コマンド

記述形式

【Windows版】

```
mpowebconfig.bat -o filename|-i filename
```

【UNIX版】

```
mpowebconfig -o filename|-i filename
```

機能説明

本コマンドは、コマンドを実行したシステムのWebサーバで使用するポート番号の情報を更新します。

オプション

-o filename

Webサーバに現在設定されているポート番号の定義をXML形式のWebサーバポート定義ファイルに出力します。filename には出力するファイル名をフルパスで指定します。ファイル名のパスには、半角英数字、“/”、“:”、“¥”、“.”のみ指定可能です。

-i filename

XML形式のWebサーバポート定義ファイルに記載されたポート番号の定義情報をWebサーバに反映させます。filenameにはファイル名をフルパスで指定します。ファイル名のパスには、半角英数字、“/”、“:”、“¥”、“.”のみ指定可能です。

復帰値

復帰値	意 味
0	正常終了
1以上	異常終了

参照

Webサーバポート定義ファイル

Webサーバポート定義ファイルについては、“20.7 Webサーバポートの定義ファイル”を参照してください。

コマンド格納場所

Windows	Systemwalker Operation Managerインストールディレクトリ¥mpwalker.jm¥mpjmweb¥bin
Solaris Linux	/opt/FJSVjmweb/bin

実行に必要な権限

システム管理者(Administratorsグループに属するユーザ/スーパーユーザ)権限が必要です。

注意事項

- Webサーバがセットアップされている状態で、-iオプションによりポート番号の更新を実行する場合は、事前にpoperationmgrコマンドでSystemwalker Operation Managerを停止してください。ポート番号の更新を有効にするには、soperationmgrコマンドでSystemwalker Operation Managerを起動してください。
- -iオプションによるポート番号の更新時にWebサーバがセットアップされていない場合は、本コマンドの実行後に“mpowebsetup”(Webサーバセットアップコマンド)を実行して、Webサーバのセットアップを行ってください。

使用例

Webサーバに設定されている現在のポート番号の定義を、XML形式のWebサーバポート定義ファイルに出力します。

【Windows版】

```
mpowebconfig.bat -o C:¥tmp¥port.xml
```

【Solaris版】

```
mpowebconfig -o /var/tmp/port.xml
```

【Linux版】

```
mpowebconfig -o /tmp/port.xml
```

XML形式のWebサーバポート定義ファイルに記載されたポート番号の定義情報をWebサーバに反映させます。

【Windows版】

```
mpowebconfig.bat -i C:¥tmp¥port.xml
```

【Solaris版】

```
mpowebconfig -i /var/tmp/port.xml
```

【Linux版】

```
mpowebconfig -i /tmp/port.xml
```

実行結果/出力形式

正常終了時は、標準出力に以下のメッセージが出力されます。

```
mpowebconfigコマンドは正常に終了しました。
```

異常終了時は、標準エラー出力に以下のメッセージが出力されます。

- [実行結果1]

  ```
  入力ファイル(%1)のオープンに失敗しました。
  ```

  %1:入力されたファイルのパス

- [実行結果2]

  ```
  入力ファイル(%1)の形式が不正です。
  ```

  %1:入力されたファイルのパス

- [実行結果3]

  ```
  入力ファイル(%1)の中で値(%2)が重複しています。
  ```

  %1:入力されたファイルのパス
  %2: 重複しているポート番号の値

- [実行結果4]

  ```
  出力ファイル(%1)のオープンに失敗しました。
  ```

  %1:ファイルの出力先として指定されたファイルパス

1.21 jmsetauthinfo 共通基盤認証情報設定コマンド【Windows版】

記述形式

[登録の場合]

jmsetauthinfo -add 定義ファイル

[削除の場合]

jmsetauthinfo -del 認証情報名

[参照の場合]

jmsetauthinfo -show [認証情報名]

機能説明

Systemwalker Operation Manager共通基盤、カレンダ機能、電源制御機能が利用する認証情報を登録/削除/参照します。

オプション

-add

定義ファイルの情報をもとに認証情報を登録します。なお、同一の認証情報名の認証情報がすでに存在する場合は上書きされます。

定義ファイル

共通基盤認証情報制御文(JSON形式)を格納してファイル名を指定します。

複数の認証情報を登録するときは、認証情報をJSONのリスト形式で複数記述します。

“共通基盤認証情報制御文”については、“1.21.1 共通基盤認証情報制御文”を参照してください。

-del

指定された認証情報名の認証情報の削除を行います。

認証情報名

削除する認証情報の認証情報名を指定します。

-show

指定された認証情報名の認証情報の参照を行います。認証情報名の指定がなければ、登録されている認証情報の一覧を表示します。

以下の情報は値が***で表示されます。

- PowerChuteユーザーのパスワード

復帰値

復帰値	意 味
0	正常終了
0以外	異常終了

コマンド格納場所

Windows	Systemwalker Operation Managerインストールディレクトリ¥MPWALKER.JM¥bin

実行に必要な権限

システム管理者(Administratorsグループに属するユーザ)権限が必要です。

注意事項

- 定義ファイルの内容に誤りがあった場合、コマンド実行時にエラーになります。
- 本コマンドで登録した認証情報は、ポリシー情報の抽出・配付の対象外となります。

使用例

登録の場合

```
jmsetauthinfo -add c:¥tmp¥authinfo.json
```

削除の場合

```
jmsetauthinfo -del powerchute
```

参照の場合

```
jmsetauthinfo -show powerchute
```

実行結果/出力形式

- -add、-delオプションを指定した場合の出力例

  標準出力なし

- -showオプションを指定した場合の出力例

```
jmsetauthinfo -show powerchute
{
  "name": "powerchute",
  "authType": "powerchute",
  "pcUser": "testuser1",
  "pcPasswd": "***"
}
```

1.21.1 共通基盤認証情報制御文

Systemwalker Operation Manager共通基盤、カレンダ機能、電源制御機能が利用する認証情報を設定する制御文です。

設定可能な認証種別は以下です。

- PowerChute認証

記述形式

JSON形式、文字コードはUTF-8で記載します。データの型はすべてstring型で指定します。

リスト形式にすることで複数の認証情報をまとめて登録することもできます。

以下に、認証種別ごとの記述形式について説明します。

PowerChute認証の場合の制御文

電源制御機能で、電源制御ソフトウェアとして“PowerChute(R) Business Edition 10.0”を利用する際に、本認証種別の制御文を登録してください。

本認証種別で設定したユーザー名・パスワードを利用して、電源制御ソフトウェアの電源操作が行われます。

```
{
  ”name”: ”認証情報名”,                                  ------(1)（※）
  ”authType”: ”認証種別”,                                ------(2)（※）
  ”pcUser”: ”PowerChuteユーザーのユーザー名”,             ------(3)（※）
  ”pcPasswd”: ”PowerChuteユーザーのパスワード”            ------(4)（※）
}
```

(*)：必ず指定してください。

(1) name 認証情報名

powerchuteを指定します。

(2) authType 認証種別

powerchuteを指定します。

(3) pcUser PowerChuteユーザーのユーザー名

PowerChuteユーザーのユーザー名を、1024バイト以内で指定してください。

以下の文字は使用できません。

¥ " %

(4) pcPasswd PowerChuteユーザーのパスワード

PowerChuteユーザーのパスワードを、1024バイト以内で指定してください。

以下の文字は使用できません。

¥ " %

第2章 セキュリティコマンド

本章では、Systemwalker Operation Managerが提供するセキュリティの設定のためのコマンドについて説明します。なお、コマンドは、Systemwalker Operation Managerのサーバ機能がインストールされているサーバで起動してください。

注意

WindowsのUAC機能を有効にしている環境でコマンドを実行する場合、OSにログインしたユーザの権限によって権限の昇格を求めるダイアログが表示され、対処が必要となります。

詳細は、“Systemwalker Operation Manager 運用ガイド”の、“Windowsでコマンドを実行する場合の注意事項【Windows版】”を参照してください。

2.1 mpsetaclコマンド

記述形式

【Windows版】

[設定／変更の場合]
mpsetacl -a -t *Target* {-u *User* | -g *Group* | -w *Wellknown* } [-s *Domain*] -p {w | c | x | r}
[削除の場合]
mpsetacl -d -t *Target* [[{-u *User* | -g *Group* | -w *Wellknown* }] [-s *Domain*]]

【UNIX版】

[設定／変更の場合]
 mpsetacl -a -t *Target* {-u *User* | -g *Group* } -p {w | c | x | r}
[削除の場合]
 mpsetacl -d -t *Target* [-u *User* | -g *Group*]

機能説明

Targetで指定したプロジェクトにユーザまたはグループのアクセス権を設定/変更/削除します。

オプション

-a

プロジェクトにユーザまたはグループのアクセス権を設定/変更する場合に指定します。

-t *Target*

アクセス権を設定/変更/削除するプロジェクト名を指定します。

プロジェクトは、以下の形式で指定してください。

【Windows版】

```
mpjobsch¥schedule¥プロジェクト名
```

EE

複数サブシステム運用をしている場合、プロジェクトは以下の形式で指定してください。(注)

```
mpjobsch¥schedule*n*¥プロジェクト名
```
n：サブシステム番号(1～9)

注)

サブシステム番号0のプロジェクトは、複数サブシステム運用をしていない場合と同じ形式で指定してください。

【UNIX版】

```
mpjobsch/schedule/プロジェクト名
```

EE GEE

複数サブシステム運用をしている場合、プロジェクトは以下の形式で指定してください。(注)

```
mpjobsch/schedulen/プロジェクト名
n：サブシステム番号(1～9)
```

注)

サブシステム番号0のプロジェクトは、複数サブシステム運用をしていない場合と同じ形式で指定してください。

-u *User*

アクセスを許可するユーザ名を指定します。

ユーザ名に空白を含む場合は、ユーザ名全体をダブルクォーテーション(")で囲ってください。

なお、ユーザプリンシパル名を指定することはできません。【Windows版】

-g *Group*

アクセスを許可するグループ名を指定します。

なお、グループ名に空白を含む場合は、グループ名全体をダブルクォーテーション(")で囲ってください。

-w *Wellknown*【Windows版】

アクセスを許可する既知グループ名を指定します。既知グループ名とは、“Everyone”や“NETWORK”といったあらかじめシステムに登録されているグループを指します。

なお、既知グループ名に空白を含む場合は、既知グループ名全体をダブルクォーテーション(")で囲ってください。

-s *Domain*【Windows版】

“-u”で指定したユーザまたは“-g”で指定したグループのドメインを指定します。省略した場合は、本コマンドを実行したコンピュータ上のユーザまたはグループのドメイン名になります。

-p

許可するアクセス権を指定します。wcxrのうちどれかを必ず指定してください。

更新権は登録権、操作権および参照権を、登録権と操作権は参照権を含みます。

w	更新権
c	登録権
x	操作権
r	参照権

-d

プロジェクトのユーザまたはグループのアクセス権を削除する場合に指定します。

本オプションを指定して、ユーザ(-uオプション)またはグループ(-gオプション)を指定しなかった場合、Targetで指定されたプロジェクトに対するすべてのアクセス権が削除されるので注意してください。ユーザまたはグループの指定がある場合は、Targetで指定したプロジェクトから指定ユーザまたはグループのアクセス権を削除します。

コマンド格納場所

Windows	Systemwalker Operation Managerインストールディレクトリ¥mpwalker¥bin
Solaris HP-UX AIX Linux	Systemwalker Operation Managerインストールディレクトリ/FJSVfwseo/bin

実行に必要な権限

本コマンドは、システム管理者(Administratorsグループに属するユーザ/スーパーユーザ)のみ実行できます。

注意事項

- “-a”指定の場合、ユーザ(-uオプション)、グループ(-gオプション)または既知グループ(-wオプション)の指定は必須です。
- クラスタ環境では、本コマンドは運用系(共有ディスクがマウントされているノード)でのみ実行してください。

  本コマンドを使用して運用系で行った設定は、フェールオーバ時に待機系に反映されます。運用系か待機系かの確認は、使用しているクラスタシステムで実施してください。

記述例

アクセス権設定時

プロジェクト“proj1”に設定されているユーザ“user1”のアクセス権として更新権を設定する場合の記述例を以下に示します。

【Windows版】

```
mpsetacl -a -t mpjobsch¥schedule¥proj1 -u user1 -p w
```

【UNIX版】

```
mpsetacl -a -t mpjobsch/schedule/proj1 -u user1 -p w
```

複数サブシステム運用をしている場合のアクセス権設定時

サブシステム番号2のプロジェクト“proj1”に設定されているユーザ“user1”のアクセス権として、更新権を設定する場合の記述例を以下に示します。

【Windows版】

```
mpsetacl -a -t mpjobsch¥schedule2¥proj1 -u user1 -p w
```

【UNIX版】

```
mpsetacl -a -t mpjobsch/schedule2/proj1 -u user1 -p w
```

アクセス権削除時

プロジェクト“proj1”に設定されているユーザ“user1”のアクセス権を削除する場合の記述例を以下に示します。

【Windows版】

```
mpsetacl -d -t mpjobsch¥schedule¥proj1 -u user1
```

【UNIX版】

```
mpsetacl -d -t mpjobsch/schedule/proj1 -u user1
```

プロジェクト削除時

ユーザ名(または、グループ名)を省略した場合、プロジェクト“proj1”に設定されているすべてのアクセス権が削除されます。

【Windows版】

```
mpsetacl -d -t mpjobsch¥schedule¥proj1
```

【UNIX版】

```
mpsetacl -d -t mpjobsch/schedule/proj1
```

2.2 mpchownコマンド

記述形式

【Windows版】

```
mpchown -t Target [-d Domain] -o Owner [-g|-u|-w]
```

【UNIX版】

```
mpchown -t Target -o Owner [-g | -u]
```

機能説明

プロジェクトの所有者にアクセス権(更新権)を設定します。

オプション

-t *Target*

所有者のアクセス権(更新権)を設定するプロジェクト名を指定します。

プロジェクトは、以下の形式で指定してください。

【Windows版】

```
mpjobsch¥schedule¥プロジェクト名
```

EE

複数サブシステム運用をしている場合、プロジェクトは以下の形式で指定してください。(注)

```
mpjobsch¥schedulen¥プロジェクト名
n：サブシステム番号(1～9)
```

注)

サブシステム番号0のプロジェクトは、複数サブシステム運用をしていない場合と同じ形式で指定してください。

【UNIX版】

```
mpjobsch/schedule/プロジェクト名
```

EE GEE

複数サブシステム運用をしている場合、プロジェクトは以下の形式で指定してください。(注)

```
mpjobsch/schedulen/プロジェクト名
n：サブシステム番号(1～9)
```

注)

サブシステム番号0のプロジェクトは、複数サブシステム運用をしていない場合と同じ形式で指定してください。

-d *Domain*【Windows版】

“-o”で指定した所有者のドメイン名を指定します。省略した場合は、本コマンドを実行したコンピュータ上のユーザまたはグループのドメイン名になります。

-o *Owner*

所有者名を指定します。

なお、ユーザプリンシパル名を指定することはできません。

-g

“-o”で指定した所有者名をグループとして設定する場合に指定します。

-u

“-o”で指定した所有者名をユーザとして設定する場合に指定します。

-w【Windows版】

“-o”で指定した所有者名を既知グループとして設定する場合に指定しま

す。既知グループとは、“Everyone”や“NETWORK”といったあらかじめシステムに登録されており、変更や削除ができないグループを指します。

コマンド格納場所

Windows	Systemwalker Operation Managerインストールディレクトリ¥mpwalker¥bin
Solaris HP-UX AIX Linux	Systemwalker Operation Managerインストールディレクトリ/FJSVfwseo/bin

実行に必要な権限

本コマンドは、システム管理者(Administratorsグループに属するユーザ/スーパーユーザ)のみ実行できます。

注意事項

- “-g(グループ)”、“-u(ユーザ)”、“-w(既知グループ)【Windows版】”のいずれも省略した場合、“-o”で指定した所有者名がユーザまたはグループとしてOSに登録されていれば、登録されている方の所有者のアクセス権が設定されます。

  ただし、“-d(ドメイン)”を省略し、かつ既知グループ名と同じユーザまたはグループ名が指定された場合は、既知グループのアクセス権が設定されます。【Windows版】

  OSにグループおよびユーザの両方で登録されている場合は、ユーザのアクセス権が設定されます。【UNIX版】

- mpchownコマンドは、ポリシーの抽出/配付によりスケジュール情報を配付した際に、プロジェクトの所有者に対して更新権を設定するコマンドとなります。本コマンドは、ポリシー適用時、自動的に内部で実行されます。

  プロジェクトの所有者となるユーザを変更する場合には、jobschsetprojectコマンドを使用してください。詳細については、“4.5 jobschsetprojectコマンド”を参照してください。

・ クラスタ環境では、本コマンドは運用系(共有ディスクがマウントされているノード)でのみ実行してください。

本コマンドを使用して運用系で行った設定は、フェールオーバ時に待機系に反映されます。運用系か待機系かの確認は、使用しているクラスタシステムで実施してください。

2.3 mkbatコマンド

記述形式

【Windows版】

```
mkbat -f file [-j] [-b]
```

【UNIX版】

```
mkbat -f file [-j]
```

機能説明

すでにプロジェクトに設定されているアクセス制御情報からアクセス権設定コマンド(mpsetacl)、所有者のアクセス権設定コマンド(mpchown)のコマンド列を作成します。本コマンドは、サーバ間で同一の定義を使用する場合などに利用します。アクセス権設定コマンド(mpsetacl)の“-u”オプションに指定したユーザ名および“-g”オプションに指定したグループ名と、所有者のアクセス権設定コマンド(mpchown)の“-o”オプションに指定したユーザ名またはグループ名がサーバごとに異なる場合は、出力したバッチファイル/シェルスクリプトの編集が必要です。

オプション

-f *file*

作成するバッチファイル名/シェルスクリプト名を指定します。

なお、既存ファイルを指定した場合は、上書きされます。

-j

-jを指定した場合は、Systemwalker Operation Managerのアクセス権の定義情報をコマンド列で作成します。Systemwalker Centric Managerがインストールされていない場合は、-jを指定して実行してください。

-jを省略した場合は、Systemwalker Operation ManagerおよびSystemwalker Centric Managerの両方のアクセス権定義情報をコマンド列で作成します。

-b【Windows版】

-jオプションで作成したアクセス権定義情報を自コンピュータ用にバックアップする場合に指定します。

コマンド格納場所

Windows	Systemwalker Operation Managerインストールディレクトリ¥mpwalker¥bin
Solaris HP-UX AIX Linux	Systemwalker Operation Managerインストールディレクトリ/FJSVfwseo/bin

実行に必要な権限

本コマンドは、システム管理者(Administratorsグループに属するユーザ/スーパーユーザ)のみ実行できます。

注意事項

HP-UX版の場合、本コマンドにより出力されたシェルスクリプトを実行するときは、以下に注意してください。

- システムの日本語環境がシフトJISコードの場合は、環境変数LANGに“ja_JP.SJIS”が設定されていることを確認してから実行してください。

Systemwalker Centric Managerのアクセス権定義情報も一緒に作成する場合は、以下に注意してください。

- Systemwalker Centric Managerのセキュリティロールの変更にともない、V13.3.1以前のSystemwalker Centric ManagerとV13.4.0以降のSystemwalker Centric Managerの間で、mkbatコマンドにより、アクセス権定義情報の作成および適用を実施する場合は、以下のことが必要です。
  - アクセス権定義情報の作成および適用先のサーバでSystemwalker Centric Managerのセキュリティロール“SecurityAdmin”または“SecurityAuditor”にユーザが登録されている場合は、“SecurityAdmin”と“SecurityAuditor”に登録されているユーザをすべて削除した後、mkbatコマンドでセキュリティ情報の抽出および適用を実施してください。
  - mkbatコマンドでアクセス権定義情報の作成または適用後、必要な場合は、“SecurityAdmin”または“SecurityAuditor”にユーザを登録してください。

クラスタ環境では、本コマンドで作成したバッチファイル／シェルスクリプトは、運用系でのみ実行してください。運用系で行った設定は、フェールオーバ時に待機系に反映されます。運用系か待機系かの確認は、使用しているクラスタシステムで実施してください。

2.4 mpaclclsコマンド

記述形式

【Windows版】

```
mpaclcls [-u [-s サブシステム番号]] | [[-s サブシステム番号 [-w]] 共有ディスクのディレクトリ ]
```

【UNIX版】

```
mpaclcls [-u ] [-s サブシステム番号]
```

機能説明

クラスタ環境において、プロジェクトに設定されているアクセス権情報を退避するための共有ディスクの情報を設定します。

また、アクセス権情報を退避するためのディレクトリを共有ディスク上に作成します。【Windows版】

オプション

-u

クラスタ環境を解除する場合に指定します。サブシステム番号が指定されている場合は、指定されたサブシステム番号のクラスタ環境を解除します。

-s サブシステム番号

サブシステム番号を指定します。本オプションは、1:1運用待機(サブシステムあり、一部クラスタ運用)または2ノード相互待機運用の場合に使用します。

1:1運用待機(サブシステムあり、一部クラスタ運用)または2ノード相互待機運用の場合、本オプションを省略するとエラーになります。また、1:1運用待機(サブシステムあり、一部クラスタ運用)または2ノード相互待機運用でない場合、本オプションを指定するとエラーになります。【UNIX版】

-w【Windows版】

待機系ノードのサブシステム番号を設定する場合に指定します。本オプションを指定する場合は、“-s”オプションの指定が必要です。

復帰値

復帰値	意 味
0	正常終了
0以外	異常終了

コマンド格納場所

Windows	Systemwalker Operation Managerインストールディレクトリ¥mpwalker¥bin
Solaris HP-UX Linux	Systemwalker Operation Managerインストールディレクトリ/FJSVfwseo/bin

実行に必要な権限

本コマンドは、システム管理者(Administratorsグループに属するユーザ/スーパーユーザ)のみ実行できます。

記述例

【Windows版】

1:1運用待機(サブシステムなし)および1:1運用待機(サブシステムあり)の運用において、共有ディスク“d:¥share”を設定します。

```
mpaclcls d:¥share
```

2ノード相互待機の運用において、サブシステム1で運用しているノードの共有ディスクの“d:¥omgr1”を設定します。

```
mpaclcls -s 1 d:¥omgr1
```

1:1運用待機(サブシステムなし) および1:1運用待機(サブシステムあり)の運用において、共有ディスク“d:¥share”の設定を解除します。

```
mpaclcls -u d:¥share
```

2ノード相互待機の運用において、サブシステム1で運用しているノードの共有ディスク“d:¥omgr1”の設定を解除します。

```
mpaclcls -u -s 1 d:¥omgr1
```

2ノード相互待機の運用において、サブシステム2で運用しているノードの共有ディスク“e:¥omgr2”を待機系ノードで設定します。

```
mpaclcls -s 1 -w e:¥omgr2
```

【UNIX版】

1:1運用待機(サブシステムなし) および1:1運用待機(サブシステムあり)の運用において、共有ディスクの情報を設定します。

```
mpaclcls
```

1:1運用待機(サブシステムあり、一部クラスタ運用)の運用において、サブシステム1で運用しているノードの共有ディスクの情報を設定します

```
mpaclcls -s 1
```

1:1運用待機(サブシステムなし) および1:1運用待機 (サブシステムあり)の運用において、共有ディスクの情報を解除します。

```
mpaclcls -u
```

1:1運用待機(サブシステムあり、一部クラスタ運用)の運用において、サブシステム1で運用しているノードの共有ディスクの情報を解除します。

```
mpaclcls -u -s 1
```

実行結果/出力形式

異常終了時は、標準エラー出力にメッセージが出力されます。

EE GEE 2.5 mpcssaveコマンド

記述形式

```
mpcssave [-s サブシステム番号]
```

機能説明

クラスタ環境において、プロジェクトに設定されているアクセス権情報およびOperation Managerユーザ情報を、クラスタの共有ディスクに退避します。

本コマンドは運用系でのみ実行してください。

オプション

-s サブシステム番号

サブシステム番号を0から9の範囲で指定します。指定したサブシステム番号のアクセス権情報がクラスタの共有ディスクに退避されます。

本オプションは、1:1運用待機(サブシステムあり、一部クラスタ運用)または2ノード相互待機運用の場合には必須です。

本オプションを省略した場合、全サブシステムが操作対象となります。

復帰値

復帰値	意 味
0	正常終了
0以外	異常終了

コマンド格納場所

Windows	Systemwalker Operation Managerインストールディレクトリ¥mpwalker¥bin
Solaris HP-UX Linux	Systemwalker Operation Managerインストールディレクトリ/FJSVfwseo/bin

実行に必要な権限

本コマンドは、システム管理者(Administratorsグループに属するユーザ/スーパーユーザ)のみ実行できます。

注意事項

【Windows版】

- イベントログに以下のメッセージが出力された場合に実行してください。

  “退避できなかったアクセス権情報があります。詳細は、*ファイル名* を参照してください。”

【UNIX版】

- SYSLOGに以下のメッセージが出力された場合に実行してください。

  "The system could not backup the access control information for some functions. For details, see [ファイル名].

記述例

プロジェクトのアクセス権情報を共有ディスクへ退避します。

```
mpcssave
```

サブシステム2で運用しているサーバで、プロジェクトのアクセス権情報を共有ディスクへ退避します。

```
mpcssave -s 2
```

2.6 mplstacluserコマンド

記述形式

```
mplstacluser [-s サブシステム番号]
```

機能説明

プロジェクトにアクセス権が設定されているアカウントの一覧を表示します。

オプション

EE GEE

-s サブシステム番号

サブシステム番号を指定します。指定されたサブシステムに登録されているプロジェクトに、アクセス権が設定されたユーザー一覧を表示します。

省略時は、サブシステム番号0のプロジェクトが対象となります。

復帰値

復帰値	意 味
0	正常終了
0以外	異常終了

コマンド格納場所

Windows	Systemwalker Operation Managerインストールディレクトリ¥mpwalker¥bin
Solaris HP-UX	Systemwalker Operation Managerインストールディレクトリ/FJSVfwseo/bin

AIX Linux	

実行に必要な権限

本コマンドは、システム管理者(Administratorsグループに属するユーザ/スーパーユーザ)のみ実行できます。

注意事項

既知グループ名は、プロジェクトにアクセス権が設定されていても表示されません。既知グループ名とは、“Everyone”や“NETWORK”といったあらかじめシステムに登録されているグループを指します。【Windows版】

Operation Managerユーザを使用している場合は、Operation Managerユーザに割り当てられているOSユーザも表示対象になります。【UNIX版】

使用例

プロジェクトにアクセス権が設定されているユーザー一覧を表示します。

```
mplstacluser
```

サブシステム1のプロジェクトにアクセス権が設定されているユーザー一覧を表示します。

```
mplstacluser -s 1
```

実行結果/出力形式

【Windows版】

```
[USER]     :  コンピュータ名¥Administrator
[USER]     :  コンピュータ名¥swuser1
[USER]     :  コンピュータ名¥swuser2
[USER]     :  ドメイン名¥domuser1
[USER]     :  ドメイン名¥domuser2
[GROUP]    :  コンピュータ名¥swgrp1
[GROUP]    :  コンピュータ名¥swgrp2
```

【UNIX版】

```
[USER]     :  root
[USER]     :  swuser1
[USER]     :  swuser2
[EX-USER]  :  omgruser1
[EX-USER]  :  omgruser2
[GROUP]    :  swgrp1
[GROUP]    :  swgrp2
```

備考.
  USER   ：OSユーザ
  GROUP   ：OSグループ
  EX-USER ：Operation Managerユーザ(拡張ユーザ管理機能が有効となっている場合のみ)

2.7 mpsetusermodeコマンド【UNIX版】

記述形式

```
mpsetusermode -s {on|off}
```

機能説明

拡張ユーザ管理機能を有効/無効にします。

オプション

-s on

拡張ユーザ管理機能を有効とするときに指定します。

-s off

拡張ユーザ管理機能を無効とするときに指定します。インストール直後は、offの状態になっています。

復帰値

復帰値	意 味
0	正常終了
0以外	異常終了

参照

mpusermode, mpadduser, mpdeluser, mpmoduser, mpsetpasswd, mpusers

コマンド格納場所

Solaris HP-UX AIX Linux	Systemwalkerインストールディレクトリ/ FJSVfwseo/bin

実行に必要な権限

システム管理者(スーパーユーザ)の権限が必要です。

注意事項

- Systemwalker Operation Managerのデーモンの動作中に、拡張ユーザ管理機能の有効/無効を変更することはできません。Systemwalker Operation Managerのデーモンを停止してから実行してください。
- OSユーザでプロジェクトを作成した後に、拡張ユーザ管理機能を有効とした場合、作成済のプロジェクトに対しては、管理者のみ更新権が設定された状態となります。必要に応じてプロジェクトにアクセス権を設定してください。所有者情報はそのまま引き継がれます。
- Operation Managerユーザでプロジェクトを作成した後に、拡張ユーザ管理機能を無効とした場合、作成済のプロジェクトに対しては、システム管理者(スーパーユーザ)およびプロジェクトの所有者のみプロジェクトに対して更新権を持った状態となります。必要に応じてプロジェクトにアクセス権を設定してください。所有者情報はそのまま引き継がれます。

記述例

拡張ユーザ管理機能を有効とします。

```
mpsetusermode -s on
```

拡張ユーザ管理機能を無効とします。

```
mpsetusermode -s off
```

実行結果/出力形式

正常終了　　なし

異常終了　　エラーが発生した旨のメッセージが表示されます。

2.8　mpusermodeコマンド【UNIX版】

記述形式

```
mpusermode
```

機能説明

拡張ユーザ管理機能が有効となっているか無効となっているかを表示します。

復帰値

復帰値	意 味
0	正常終了
0以外	異常終了

参照

mpsetusermode

コマンド格納場所

Solaris HP-UX AIX Linux	Systemwalkerインストールディレクトリ/ FJSVfwseo/bin

実行に必要な権限

システム管理者(スーパーユーザ)の権限が必要です。

記述例

拡張ユーザ管理機能が有効となっているか無効となっているかを表示します。

```
mpusermode
```

実行結果/出力形式

拡張ユーザ管理機能が有効となっている場合

```
Operation Manager user mode is ON.
```

拡張ユーザ管理機能が無効となっている場合

```
Operation Manager user mode is OFF.
```

2.9　mpadduserコマンド【UNIX版】

記述形式

mpadduser -o *OSユーザ名* [-d *説明*] [-a] *username*

機能説明

Systemwalker Operation Manager上に、Operation Managerユーザを登録します。

オプション

-o *OSユーザ名*

Operation Managerユーザに割り当てるOSユーザを指定します。管理者を登録する場合は、システム管理者(スーパーユーザ)名を指定します。非管理者を登録する場合は、一般ユーザ名を指定します。-aオプションを指定して管理者を登録する場合、本オプションで一般ユーザ名を指定しても、登録できません。また、-aオプションを指定しない(非管理者を登録する)場合は、本オプションでシステム管理者(スーパーユーザ)名を指定しても、登録できません。

-d *説明*

登録するOperation Managerユーザについての説明を記述します。省略可能です。

-a

Operation Managerユーザを管理者として登録する場合に指定します。管理者は、すべてのプロジェクトに対し、更新権を持ちます。管理者に割り当てられるOSユーザは、システム管理者(スーパーユーザ)となります。

本オプションの指定がない場合、一般ユーザとなります。

username

登録するOperation Managerユーザ名を20バイト以内で指定します。

rootという名前のOperation Managerユーザは、インストール時に作成されるため、指定できません。

復帰値

復帰値	意 味
0	正常終了
0以外	異常終了

参照

mpdeluser, mpmoduser, mpsetpasswd, mpusers

コマンド格納場所

Solaris HP-UX AIX Linux	Systemwalkerインストールディレクトリ/ FJSVfwseo/bin

実行に必要な権限

システム管理者(スーパーユーザ)の権限が必要です。

注意事項

- 拡張ユーザ管理機能が無効となっている場合は、登録したOperation Managerユーザは使用できません。
- Operation Managerユーザ名として使用できる文字は、以下のとおりです。

カテゴリ	文字
英字	A～Z a～z
数字	0～9
記号	_

- クラスタ環境では、本コマンドは運用系(共有ディスクがマウントされているノード)でのみ実行してください。

  本コマンドを使用して運用系で行った設定は、フェールオーバ時に待機系に反映されます。運用系か待機系かの確認は、使用しているクラスタシステムで実施してください。

- コマンド実行時にusageが表示された場合は、以下を確認してください。

  - コマンドに指定するオプションに誤りがないか
  - コマンドの各オプションに指定する値に改行文字が含まれていないか

記述例

“swroot”という名前のOperation Managerユーザを管理者として登録します。

```
mpadduser -o root -d “Operation Manager Administrator” -a swroot
```

“swuser”という名前のOperation Managerユーザを登録します。OS上のユーザ権限はuserとします。

```
mpadduser -o user swuser
```

実行結果/出力形式

正常終了:　なし

異常終了:　エラーが発生した旨のメッセージが表示されます。

2.10　mpdeluserコマンド【UNIX版】

記述形式

```
mpdeluser username [username ・・・] | -all
```

機能説明

Systemwalker Operation Manager上に登録されているOperation Managerユーザを削除します。

オプション

username

削除するOperation Managerユーザ名を指定します。複数指定できます。

-all

Systemwalker Operation Manager上に登録されているOperation Managerユーザをすべて削除します。

Systemwalker Operation Managerのデーモン動作中に本オプションを指定することはできません。本オプションを指定する場合は、Systemwalker Operation Managerのデーモンを停止してからコマンドを実行してください。

復帰値

復帰値	意 味
0	正常終了
0以外	異常終了

参照

mpusermode, mpadduser, mpmoduser, mpsetpasswd, mpusers

コマンド格納場所

Solaris HP-UX AIX Linux	Systemwalkerインストールディレクトリ/ FJSVfwseo/bin

実行に必要な権限

システム管理者(スーパーユーザ)の権限が必要です。

注意事項

- -allオプションを指定する場合は、Systemwalker Operation Managerのデーモンが停止している必要があります。
- Systemwalker Operation Managerを利用しているユーザを、Systemwalker Operation Manager運用中に削除しないでください。削除したユーザは、すべてのプロジェクトに対してアクセス権がないものとして動作します。なお、削除したユーザにマスタプロジェクトのアクセス権が設定されている場合は、再度プロジェクトに対してマスタ設定を行う必要があります。
- クラスタ環境では、本コマンドは運用系(共有ディスクがマウントされているノード)でのみ実行してください。

  本コマンドを使用して運用系で行った設定は、フェールオーバ時に待機系に反映されます。運用系か待機系かの確認は、使用しているクラスタシステムで実施してください。

記述例

Systemwalker Operation Manager上に登録されたOperation Managerユーザ“swroot”を削除します。

```
mpdeluser swroot
```

Systemwalker Operation Manager上に登録されたOperation Managerユーザ“swroot”と“swuser” を削除します。

```
mpdeluser swroot swuser
```

Systemwalker Operation Manager上に登録されたOperation Managerユーザをすべて削除します。

```
mpdeluser -all
```

実行結果/出力形式

正常終了　　なし

異常終了　　エラーが発生した旨のメッセージが表示されます。

2.11　mpmoduserコマンド【UNIX版】

記述形式

mpmoduser {-o *OSユーザ名* | -d *説明* } *username*

機能説明

Systemwalker Operation Manager上に登録された Operation Managerユーザに割り当てるOSユーザ名を設定します。

オプション

-o *OSユーザ名*

mpadduserコマンドで、Operation Managerユーザに割り当てたOSユーザを変更する場合に指定します。

Operation Managerユーザに対応づけられたOSのユーザ名がシステム管理者の場合は、システム管理者のOSユーザ名への変更のみできます。Operation Managerユーザに対応づけられたOSのユーザ名がシステム管理者以外の場合は、システム管理者以外のOSユーザ名への変更のみできます。

-d *説明*

Operation Managerユーザについての説明を変更する場合に指定します。

username

属性を変更するOperation Managerユーザ名を指定します。

復帰値

復帰値	意 味
0	正常終了
0以外	異常終了

参照

mpadduser, mpsetpasswd, mpusers

コマンド格納場所

Solaris HP-UX AIX Linux	Systemwalkerインストールディレクトリ/ FJSVfwseo/bin

実行に必要な権限

システム管理者(スーパーユーザ)の権限が必要です。

注意事項

- クラスタ環境では、本コマンドは運用系(共有ディスクがマウントされているノード)でのみ実行してください。

  本コマンドを使用して運用系で行った設定は、フェールオーバ時に待機系に反映されます。運用系か待機系かの確認は、使用しているクラスタシステムで実施してください。

- コマンド実行時にusageが表示された場合は、以下を確認してください。

  ー コマンドに指定するオプションに誤りがないか

  ー コマンドの各オプションに指定する値に改行文字が含まれていないか

記述例

Operation Managerユーザ“swuser”に割り当てられているOSユーザを“user2”に変更します。

```
mpmoduser -o user2 swuser
```

Operation Managerユーザ“swuser”の説明を、“Operation Manager User”に変更します。

```
mpmoduser -d “Operation Manager User” swuser
```

実行結果/出力形式

正常終了　なし

異常終了　エラーが発生した旨のメッセージが表示されます。

2.12 mpsetpasswdコマンド【UNIX版】

記述形式

```
mpsetpasswd [-d] username
```

機能説明

Systemwalker Operation Manager上に登録されたOperation Managerユーザのパスワードを設定/変更/削除します。パスワードが削除されたOperation Managerユーザはログインできません。rootという名前のOperation Managerユーザのパスワードも削除できます。

オプション

-d

パスワードを削除する場合に指定します。なお、パスワードが削除されたユーザはログインできません。

username

パスワードを設定/変更/削除するOperation Managerユーザ名を指定します。

復帰値

復帰値	意 味
0	正常終了
0以外	異常終了

参照

mpadduser, mpmoduser, mpusers

コマンド格納場所

Solaris HP-UX AIX Linux	Systemwalkerインストールディレクトリ/ FJSVfwseo/bin

実行に必要な権限

システム管理者(スーパーユーザ)の権限が必要です。

注意事項

- パスワードとして使用できる文字は、以下のとおりです。

カテゴリ	文字
英字	A～Z a～z
数字	0～9
記号	! " # % & ' ( ) * + , - . / : ; < = > ? [¥] ^ _ { | } ~
空白	‘　‘

- パスワードに設定できるのは、半角英数文字5～50文字以内です。

  Linux版の場合

  　パスワードとして有効な長さは、先頭から半角英数文字8文字までとなります。

  Solaris版の場合

  　パスワードとして設定した半角英数文字5～50文字の範囲で、Operation Managerユーザの暗号化アルゴリズムで許可されている文字数までが、パスワードとして有効となります。

  　例：
  　従来のUNIX暗号化アルゴリズム(DES)

  　　パスワードとして有効な長さは、先頭から半角英数文字8文字までとなります。

  　従来のUNIX暗号化アルゴリズム以外

  　　半角英数文字5～50文字の範囲で入力した文字列すべてが、パスワードとして有効となります。

- クラスタ環境では、本コマンドは運用系(共有ディスクがマウントされているノード)でのみ実行してください。

  本コマンドを使用して運用系で行った設定は、フェールオーバ時に待機系に反映されます。運用系か待機系かの確認は、使用しているクラスタシステムで実施してください。

記述例

swuser という名前のOperation Managerユーザのパスワードを設定/変更します。

```
mpsetpasswd swuser
Password: ← 設定/変更するパスワードを入力します。
          (エコーバックされません)
Retype: ← 設定/変更するパスワードをもう一度入力します。
          (エコーバックされません)
```

rootという名前のOperation Managerユーザのパスワードを削除します。

```
mpsetpasswd -d root
```

実行結果/出力形式

正常終了　　なし

異常終了　　エラーが発生した旨のメッセージが表示されます。

2.13　mpusersコマンド【UNIX版】

記述形式

```
mpusers [-a]
```

機能説明

Systemwalker Operation Manager上に登録されているOperation Managerユーザを一覧表示します。

オプション

-a

名前に加え、登録されているOperation Managerユーザの以下の属性についても表示します。

- 説明
- 割り当てられているOSユーザ名
- 管理者であるか否か

本指定を省略した場合は、ユーザ名のみ表示されます。

復帰値

復帰値	意 味
0	正常終了
0以外	異常終了

参照

mpadduser, mpdeluser, mpmoduser

コマンド格納場所

Solaris HP-UX AIX Linux	Systemwalkerインストールディレクトリ/ FJSVfwseo/bin

実行に必要な権限

システム管理者(スーパーユーザ)の権限が必要です。

記述例

Systemwalker Operation Manager上に登録されているOperation Managerユーザ名を一覧表示します。

```
mpusers
```

Systemwalker Operation Manager上に登録されているOperation Managerユーザを一覧表示します。ユーザの属性も表示します。

```
mpusers -a
```

実行結果/出力形式

-aオプションを指定しなかった場合

```
# mpusers
swuser1
swuser2
```

```
swuser3
#
```

-aオプションを指定した場合

```
# mpusers -a
NAME      OS USERNAME   ADMIN  DESCRIPTION
swroot1   root          Yes    Operation Manager Administrator
swroot2   root          Yes    Operation Manager Administrator
swuser1   user          No     Operation Manager User
```

出力項目の説明

出力項目について説明します。

-aオプションを指定しなかった場合は、登録されているOperation Managerユーザ名のみが表示されます。

NAME

登録されている、Operation Managerユーザ名が表示されます。

OS USERNAME

Operation Managerユーザに割り当てられているOSユーザ名が表示されます。

ADMIN

管理者の場合は“Yes”、非管理者の場合は、“No”と表示されます。

DESCRIPTION

登録されている説明が表示されます。

2.14 mpsetupolmodeコマンド【UNIX版】

記述形式

```
mpsetupolmode -u {clear|keep}
```

機能説明

Operation Managerユーザ情報をポリシー配付する場合、配付先のサーバ上に登録されているユーザ情報を削除してから配付するか、上書きで配付するかを設定します。本コマンドは、ポリシー抽出時に、ポリシー抽出元のサーバで使用します。

オプション

-u clear

ポリシー配付時に、配付先のサーバ上に登録されているOperation Managerユーザ情報を削除してから配付を行う場合に指定します。

-u keep

ポリシー配付時に、配付先のサーバ上に登録されているOperation Managerユーザ情報を削除せずに配付を行う場合に指定します。同じ名前のOperation Managerユーザが存在する場合は、上書きされます。

復帰値

復帰値	意 味
0	正常終了
0以外	異常終了

参照

mpusermode, mpadduser, mpdeluser, mpmoduser, mpusers

コマンド格納場所

Solaris HP-UX AIX Linux	Systemwalkerインストールディレクトリ/ FJSVfwseo/bin

実行に必要な権限

システム管理者(スーパーユーザ)の権限が必要です。

注意事項

- 本設定は、ポリシー抽出元のサーバで、ポリシー抽出前に行ってください。以下の場合、本コマンドを実行しても、設定が有効となりません。
  - ー ポリシー抽出元のサーバで、ポリシー抽出後に設定した場合
  - ー ポリシー配付先のサーバで設定した場合
- インストール直後は、keep(削除しない)が設定されています。

記述例

Operation Managerユーザ情報のポリシー抽出前に、配付先のOperation Managerユーザの情報を削除してから配付する設定にします。本設定を行った後、ポリシー抽出/配付を実施すると、配付先のOperation Managerユーザの情報は削除されます。

```
mpsetupolmode -u clear
```

Operation Managerユーザ情報のポリシー抽出前に、配付先のOperation Managerユーザの情報を残して配付する設定にします。本設定を行った後、ポリシー抽出/配付を実施すると、配付先のOperation Managerユーザの情報は削除されません。同じOperation Managerユーザが存在する場合は、上書きされます。

```
mpsetupolmode -u keep
```

実行結果/出力形式

正常終了　　なし

異常終了　　エラーが発生した旨のメッセージが表示されます。

2.15　mpupolmodeコマンド【UNIX版】

記述形式

```
mpupolmode
```

機能説明

Operation Managerユーザ情報をポリシー配付する際に、配付先のOperation Managerユーザ情報を削除する設定となっているか削除しない設定となっているかを表示します。本コマンドは、ポリシー抽出元のサーバで使用します。

復帰値

復帰値	意 味
0	正常終了
0以外	異常終了

参照

mpsetupolmode

コマンド格納場所

Solaris HP-UX AIX Linux	Systemwalkerインストールディレクトリ/ FJSVfwseo/bin

実行に必要な権限

システム管理者(スーパーユーザ)の権限が必要です。

記述例

配付先のOperation Managerユーザ情報を削除してから配付するか、削除せずに配付する設定となっているか、ポリシー抽出元のサーバで表示します。

```
mpupolmode
```

実行結果/出力形式

ポリシー配付先のユーザ情報を削除しない設定の場合

```
Operation Manager user policy distribution mode is KEEP.
```

ポリシー配付先のユーザ情報を削除する設定の場合

```
Operation Manager user policy distribution mode is CLEAR.
```

2.16 mpprjcmdaclコマンド【UNIX版】

記述形式

```
mpprjcmdacl [-s サブシステム番号] プロジェクト名
```

機能説明

プロジェクトに対するOSユーザのアクセス権情報を表示します。

オプション

EE GEE

-s サブシステム番号

複数サブシステム運用をしているシステムで、表示対象のサブシステムを指定するためのオプションです。表示対象とするサブシステム番号を、0から9の範囲で指定します。このオプションを省略した場合、サブシステム番号0が表示対象となります。

プロジェクト名

OSユーザのアクセス権一覧を表示するプロジェクト名を指定します。

復帰値

復帰値	意 味
0	正常終了
0以外	異常終了

参照

mpsetusermode, mpusermode

コマンド格納場所

Solaris HP-UX AIX Linux	Systemwalkerインストールディレクトリ/ FJSVfwseo/bin

実行に必要な権限

システム管理者(スーパーユーザ)の権限が必要です。

記述例

プロジェクト“project”のOSユーザのアクセス権一覧を表示します。

```
mpprjcmdacl project
```

実行結果/出力形式

```
# mpprjcmdacl project
OS USERNAME   ACCESS RIGHTS
root          Update    Right
user          Change    Right
user2         Operation Right
user3         Reference Right
```

出力項目の説明

出力項目について説明します。

OS USERNAME

OSユーザ名が表示されます。

ACCESS RIGHTS

アクセス権が表示されます。

Update Right　: 更新権

Change Right　: 登録権

Operation Right : 操作権

Reference Right : 参照権

2.17　mpmkpwbatコマンド【UNIX版】

記述形式

```
mpmkpwbat [-c] [-u username] -f filename
```

機能説明

Systemwalker Operation Manager上に登録されているOperation Managerユーザのパスワード情報をファイルに抽出します。

オプション

-c

パスワードが設定されていない、または削除されているユーザのパスワード情報も抽出対象とする場合に指定します。本オプションを指定してパスワード情報を抽出し、スクリプトとして実行すると、実行先に登録されている同じ名前のOperation Managerユーザのパスワード情報が削除されます。本オプションを指定しない場合は、パスワードが設定されていないユーザのパスワード情報は抽出されません。

-u *username*

指定されたユーザのパスワード情報のみ抽出します。本指定を省略した場合は、すべてのユーザのパスワード情報を抽出します。

-f *filename*

抽出先のファイル名を指定します。既存のファイル名を指定すると、既存のファイルに上書きします。

本ファイルをスクリプトとして実行することで、本コマンド実行サーバ上のOperation Managerユーザのパスワード情報を、抽出元サーバに登録されているOperation Managerユーザのパスワードに変更できます。本ファイルをスクリプトとして実行できるのは、システム管理者(スーパーユーザ)のみです。

復帰値

復帰値	意 味
0	正常終了
0以外	異常終了

参照

mpsetusermode, mpusermode

コマンド格納場所

Solaris HP-UX AIX Linux	Systemwalkerインストールディレクトリ/ FJSVfwseo/bin

実行に必要な権限

システム管理者(スーパーユーザ)の権限が必要です。

注意事項

- パスワード情報は暗号化された状態で抽出されます。
- 本コマンドにより抽出したファイルには、ユーザのパスワード情報が含まれています。取り扱いには十分注意してください。

記述例

Systemwalker Operation Manager上に登録されているOperation Managerユーザのパスワード情報をファイル“users.sh”に抽出します。

```
mpmkpwbat -f users.sh
```

Systemwalker Operation Manager上に登録されている“swuser1”という名前のOperation Managerユーザのパスワード情報をファイル“swuser1.sh”に抽出します。

```
mpmkpwbat -u swuser1 -f swuser1.sh
```

実行結果/出力形式

正常終了　　なし

異常終了　　エラーが発生した旨のメッセージが表示されます。

2.18　mpsetlogsend_omgrコマンド

記述形式

```
mpsetlogsend_omgr
mpsetlogsend_omgr {-n|-y}
mpsetlogsend_omgr [{-n|-y}] -f <出力先ファイル> -k {<保存日数>|unlimit}
mpsetlogsend_omgr -d
```

機能説明

Systemwalker Operation Managerが出力する監査ログの出力設定を行います。

オプション

なし

現在の監査ログ出力設定を出力します。出力する情報は以下のとおりです。

```
LOG NAME          : Systemwalker Operation Manager audit log
LOG OUTPUT        : ON
LOG KEEP DAYS     : 31 day(s)
OUTPUT FILE       : /var/opt/FJSVftlo/audit/log/mp_omgr_auditYYMMDD.log
```

出力内容は、以下のとおりです。

LOG NAME	対象となる監査ログの名称です。
LOG OUTPUT	監査ログを出力する設定になっている場合には"ON"、出力しない設定になっている場合は"OFF"が表示されます。
LOG KEEP DAYS	過去に出力された監査ログの保存日数を表示します。表示される内容は、"1 day(s)"、"2 day(s)"・・・"999 day(s)"、"unlimited"(無制限)です。
OUTPUT FILE	監査ログの出力先ファイルを表示します。表示される内容のうち、"YYMMDD"の部分は、実際には日時(YY:西暦下2桁、MM:月、DD:日)に置き換えられます。
LOG COLLECTION (注)	Systemwalker Centric Managerの監査ログ管理機能で、監査ログを収集する設定になっている場合は"ON"、収集しない設定になっている場合は"OFF"が表示されます。
COLLECTION FILE (注)	Systemwalker Centric Manager監査ログ管理機能で、収集対象となっているファイルを出力します。ログ収集設定が一度も行われていない場合は、"-"を出力します。

注) Systemwalker Centric Manager V13.2.0以降の運用管理サーバ/資産管理サーバ/部門管理サーバ/業務サーバ/運用管理クライアントが共存している場合のみ利用できます。

-f <出力先ファイル>

監査ログの出力先をフルパスで指定します。

出力先として指定できる文字数は、246バイト以内です。日本語を含むパスを指定することも可能ですが、言語コードにより指定できる文字数が異なるため注意してください。

ログファイル名は、-fオプションで指定したファイルの末尾に、"YYMMDD.log"(YYMMDDは日付を表す)が付加されます。

例)

-fオプションに、"/var/audit/audit_log"を指定した場合、2006年5月19日に出力されるログファイルは、"/var/audit/audit_log060519.log"となります。

-k <保存日数>

監査ログの保存期間を指定します。-kの後に、1～999(単位:日)の数字、または"unlimit"(無制限)を指定します。1000以上の値を指定した場合、パラメタ異常で終了します。"unlimit"を指定した場合、出力した監査ログファイルが削除されずに残ります。

-d

監査ログ出力を解除したい場合に指定します。本オプションを指定した場合、監査ログが出力されなくなります。

また、同時にSystemwalker Centric Managerの監査ログ管理機能による収集も停止します。監査ログ管理機能による収集のみを停止する場合は、-nオプションを指定します。(注)

-y (注)

Systemwalker Centric Managerの監査ログ管理機能で、監査ログファイルを運用管理サーバへ収集する場合に指定します。本オプションを指定した場合、出力した監査ログは、自動的に運用管理サーバへ収集されます。

また、監査ログ管理機能で収集対象となるファイルの収集設定も、本オプションで更新されます。

-n (注)

Systemwalker Centric Managerの監査ログ管理機能で、監査ログファイルの運用管理サーバへの収集を一時中止する場合に指定します。本オプションを指定した場合、出力した監査ログは、自動的に運用管理サーバへ収集されなくなります。再度、監査ログを収集するには、-yオプションを指定します。

注)

Systemwalker Centric Manager V13.2.0以降の運用管理サーバ/資産管理サーバ/部門管理サーバ/業務サーバ/運用管理クライアントが共存している場合のみ利用できます。

復帰値

復帰値	意味
0	正常終了
1	異常終了

コマンド格納場所

Windows	Systemwalker Operation Managerインストールディレクトリ¥mpwalker.jm¥bin
Solaris HP-UX	/usr/bin

AIX Linux	

実行に必要な権限

本コマンドは、システム管理者(Administratorsグループに属するユーザ/スーパーユーザ)のみ実行できます。

注意事項

- 監査ログ出力設定を変更した後、Systemwalker Centric Managerの監査ログ管理機能の収集設定を更新していない場合(-yオプションを指定していない場合)、監査ログ出力先(OUTPUT FILE)の内容と運用管理サーバに収集された内容が異なる場合があります。ログ収集を正しく行いたい場合は、-yオプションを指定してコマンドを実行してください。
- 監査ログの出力先を変更した場合、出力先ディレクトリのアクセス権を再設定する必要があります。詳細については、“Systemwalker Operation Manager 導入ガイド”の“利用者制限の定義”を参照してください。
- Linux x64上で、以下の表のとおりSystemwalker Operation ManagerがSystemwalker Centric Managerと混在する場合、ACLマネージャの監査ログは、2つのファイルに分けて出力されます。

		Centric Manager	
		V13.4.0以降 (Linux x86版)	**V13.4.0以降 (Linux x64版)**
Operation Manager	**V13.2.0 ～ V13.3.0**	問題なし	Centric ManagerとOperation Managerのインストール順番に関係なく、ACLマネージャの監査ログは、2つのファイルに分けて出力される
	V13.4.0以降 (Linux x86版)	問題なし	Operation Managerよりも後にCentric Managerをインストールした場合に、ACLマネージャの監査ログは、2つのファイルに分けて出力される
	V13.4.0以降 (Linux x64版)	Operation Managerよりも後にCentric Managerをインストールした場合に、ACLマネージャの監査ログは、2つのファイルに分けて出力される	問題なし

出力先のファイル名は、以下のとおりです。

- mpsetlogsend_omgr(監査ログ設定コマンド)で表示される出力先
- 上記ファイルの日付の前に「_acl」を付加したファイル
  (例：/var/opt/FJSVftlo/audit/log/mp_omgr_audit_aclYYMMDD.log)

「_acl」を付加したファイルは、mpsetlogsend_omgr(監査ログ設定コマンド)の実行結果として表示されませんが、2つのファイルの監査ログ出力設定のON/OFF、保存期間および収集対象かどうかの設定は同じになります。

第3章 カレンダ/電源制御コマンド

本章では、Systemwalker Operation Managerが提供するカレンダコマンドおよび電源制御コマンドについて説明します。なお、コマンドは、Systemwalker Operation Managerのサーバ機能がインストールされているサーバで起動してください。

注意

WindowsのUAC機能を有効にしている環境でコマンドを実行する場合、OSにログインしたユーザの権限によって権限の昇格を求めるダイアログが表示され、対処が必要となります。

詳細は、“Systemwalker Operation Manager 運用ガイド”の、“Windowsでコマンドを実行する場合の注意事項【Windows版】”を参照してください。

3.1 f3crheet 待ち合わせ終了通知コマンド

記述形式

```
f3crheet 待ち合わせ終了通知
```

機能説明

シャットダウン/電源切断処理を特定の処理やアプリケーションの終了まで延期する場合に、本コマンドで待ち合わせ終了通知を行います。

本コマンドで待ち合わせ終了通知を行う場合は、あらかじめ[待ち合わせ終了通知定義]ダイアログボックスに通知する文字列(本コマンドのオプション)を定義しておく必要があります。

なお、本コマンドはWindows版、Solaris版およびLinux版でのみサポートされています。

オプション

待ち合わせ終了通知

特定の処理やアプリケーションが終了したことを示す文字列を指定します。

半角で100文字まで指定できます。文字列の中に空白およびコンマ(,)を含むことはできません。

他の処理やアプリケーションおよび他ノードで指定する“待ち合わせ終了通知”と一致しない文字列を指定してください。

使用方法(ジョブネットの終了後にシャットダウンを行う方法)

運用上、最後に起動・終了するジョブとして、本コマンドを使用すると、ジョブネットの終了後に待ち合わせ終了通知が出力されます。この待ち合わせ終了通知を[待ち合わせ終了通知定義]ダイアログボックスで定義しておくと、本コマンドで指定した待ち合わせ終了通知が出力されるまでシャットダウン処理が延期されます。シャットダウン処理が完了した後、電源制御によってシステムの電源は切断されます。

標準提供されるジョブスケジューラの終了処理ジョブネットのシャットダウン出口では、本コマンドが設定されています。【Windows版】

ジョブスケジューラの終了処理ジョブネットのシャットダウン出口以外でも本コマンドを発行すれば、ジョブネットの終了以外の事象も同時に待ち合わせてシステムをシャットダウンできるので便利です。【Windows版】

復帰値

復帰値	意味
0	正常に終了しました。

復帰値	意味
10	何らかのエラーが発生しました。主な原因を以下に示します。 ・オプションに誤りがあります。 ・すでに待ち合わせを完了しています。
20	ノード間において通信エラーが発生しました。

コマンド格納場所

Windows	Systemwalker Operation Managerインストールディレクトリ¥mpwalker¥bin
Solaris	/opt/FJSVjmcal/bin
Linux	/opt/FJSVjmcal/bin

EE

3.2 calregpath カレンダ自動反映の共用ディスクディレクトリ登録コマンド【Windows版】

記述形式

```
calregpath -add {-sys サブシステム番号 | -all} [-pm] -p 共用ディスクのディレクトリ
calregpath -del {-sys サブシステム番号 | -all}
calregpath -ref
```

機能説明

本コマンドは、クラスタ運用を行う際、共用ディスク上のディレクトリのパス情報をカレンダサービスに登録、または、カレンダサービスに登録された情報を削除するときに使用します。登録する情報は、クラスタの運用形態により異なります。

クラスタを構成する各ノード上でそれぞれ実行する必要があります。

オプション

-add

共用ディスク上のディレクトリを登録します。

本オプションを指定した場合、-sysオプションと-allオプションのいずれか、および、-pオプションを指定する必要があります。

-allオプションと-sysオプションは同時に指定できません。

-sys *サブシステム番号*

クラスタシステムの形式が1:1運用待機(サブシステムあり、一部クラスタ運用)または以下の相互待機系運用のときに、ジョブスケジューラでクラスタ対象としているサブシステムの番号を指定します。サブシステム番号のパス情報が編集されます。

- 2ノード相互待機
- フェールオーバリング
- ランダム

本オプションはサブシステムの数だけ実行する必要があります。

-delオプション指定時に指定するときは、指定のサブシステム番号のパス情報が、初期状態に戻ります。

-all

クラスタシステムの形式が以下の待機系運用のときに指定します。

- 1:1運用待機(サブシステムなし)および1:1運用待機(サブシステムあり)
- N:1運用待機
- カスケード

-delオプション指定時に指定するときは、登録したすべてのサブシステム番号のパス情報が初期状態に戻ります。

-pm

-addオプション指定時に使用します。

本オプションを指定した場合、変更前のディレクトリに格納されているデータを-pオプションで指定した先のディレクトリへ移行します。

クラスタ環境設定時に、データを共有ディスクへ移行する場合などに使用してください。省略した場合は、ディレクトリの参照先情報のみ更新されます。

本オプションを指定する場合は、現在設定されている参照先ディレクトリと-pオプションで指定されたディレクトリ先が共に参照可能な状態である必要があります。

-p *共用ディスクのディレクトリ名*

クラスタ導入時に作成した、共用ディスク上のディレクトリを指定します。ディレクトリのパス名は255バイトまでの長さで指定します。-addオプションを指定した場合には必ず指定します。

-del

現在登録されているパス情報を初期状態に戻します。

-sysオプション、-allオプションのいずれかと組み合わせて使用します。

-ref

現在、登録されている共用ディレクトリのパス情報を標準出力に一覧表示します。

初期状態では、サブシステムごとに以下の初期情報が表示されます。

- Systemwalkerインストールディレクトリ¥MPWALKER.JM¥mpjmcal¥post¥sys(*n*)

(*n*):サブシステム番号

復帰値

復帰値	意味
0	正常終了
1	異常終了。エラーの内容はエラーメッセージを確認してください。

コマンド格納場所

Windows	Systemwalker Operation Managerインストールディレクトリ¥MpWalker.JM¥bin

注意事項

- 本コマンドは、システム管理者(Administratorsグループに属するユーザ)のみ実行できます。
- 登録、削除を行う場合は、カレンダサービスを停止した後に実行してください。

3.3 calsetcluster カレンダ自動反映セットアップコマンド

記述形式

```
calsetcluster [-p] [-a] {-type クラスタシステムの形式}
calsetcluster -d
calsetcluster -v
```

機能説明

クラスタ導入時にカレンダ情報の自動反映機能を有効にするコマンドです。

本コマンドは、calregpathコマンドで共用ディレクトリの設定を行った後に実行します。【Windows版】

以下の情報が自動反映されます。

- システムの日変わり時刻
- カレンダ休日情報
- SYSTEM_CALENDARのスケジュール情報、電源スケジュール情報、終了監視情報(設定時に選択可能)
- サービス・アプリケーション/アプリケーション起動情報(設定時に選択可能)

オプション

-p

本オプションを指定すると、SYSTEM_CALENDARのスケジュール情報、電源スケジュール情報、終了監視情報が反映の対象から外れます。省略した場合は自動反映の対象となります。

これらの情報を反映させるには、更新を行った反映元のサーバと反映先のサーバで共にこれらの情報の更新を許可する設定(-pオプションなし)になっている必要があります。

注意

情報反映先の全サーバ間で、電源制御設定の内容が統一されていない場合は、必ず本オプションを反映対象の全サーバで指定してください。

-a

本オプションを指定すると、サービス・アプリケーション/アプリケーション起動の情報が反映の対象から外れます。省略した場合は自動反映の対象となります。

サービス・アプリケーション/アプリケーション起動の情報を反映させるには、更新を行った反映元のサーバと反映先のサーバで共にこの情報の更新を許可する設定(-aオプションなし)になっている必要があります。

-type *クラスタシステムの形式*

本オプションを指定すると、自動反映機能が有効になります。運用を行うクラスタ形式に合わせて下記のように指定します。

[-type s]

1:1 運用待機

[-type n]

N:1 運用待機

[-type e]

2ノード相互待機

[-type r]

フェールオーバリング

[-type d]

ランダム

[-type c]

カスケード

-d

本オプションを指定すると、自動反映機能を無効にします。

注意

クラスタシステムでSystemwalker Operation Managerをアンインストールする場合は、アンインストールを行う前に自動反映機能を無効にしてください。自動反映機能を無効にせずにアンインストールを行った場合は、以下のシンボリックリンクが残るので、手動で削除してください。【UNIX版】

```
Solaris版 ： /etc/rc3.d/S28JMCAL
HP-UX版： /sbin/rc3.d/S28JMCAL
Linux版： シンボリックリンクは作成されません。
AIX版： シンボリックリンクは作成されません
```

-v

現在の設定内容を表示します。

復帰値

復帰値	意味
0	正常終了
1	異常終了。エラーの内容はエラーメッセージを確認してください。

コマンド格納場所

Windows	Systemwalker Operation Managerインストールディレクトリ¥MpWalker.JM¥bin
Solaris	/opt/FJSVjmcal/bin
HP-UX	/opt/FHPjmcal/bin
Linux	/opt/FJSVjmcal/bin
AIX	/opt/FAIXjmcal/bin

注意事項

- 本コマンドは、システム管理者(Administratorsグループに属するユーザ/スーパーユーザ)のみ実行できます。
- 本コマンドは、カレンダサービス/デーモンを停止した後に実行してください。
- サブシステムの日変わり時刻については個別に反映する必要はありません。

3.4 calrecpl カレンダ反映先ホスト定義の再読み込みコマンド

記述形式

```
calrecpl
```

機能説明

クラスタ運用時に使用するコマンドです。

反映先ホスト定義を変更した場合、本コマンドを実行することでカレンダサービス/デーモンを停止させることなく、定義の変更が有効になります。

オプション

なし

復帰値

復帰値	意味
0	正常終了
1	異常終了。エラーの内容はエラーメッセージを確認してください。

コマンド格納場所

Windows	Systemwalker Operation Managerインストールディレクトリ¥MpWalker.JM¥bin
Solaris	/opt/FJSVjmcal/bin
HP-UX	/opt/FHPjmcal/bin
Linux	/opt/FJSVjmcal/bin
AIX	/opt/FAIXjmcal/bin

注意事項

- 本コマンドは、システム管理者(Administratorsグループに属するユーザ/スーパーユーザ)のみ実行できます。
- 本コマンド実行時には、カレンダサービス/デーモンが起動している必要があります。

第4章 ジョブスケジューラコマンド

本章では、Systemwalker Operation Managerが提供するジョブスケジューラコマンドについて説明します。コマンドは、Systemwalker Operation Managerのサーバ機能がインストールされているサーバで起動してください。なお、情報を出力するためのジョブスケジューラコマンドについては、“第5章 ジョブスケジューラコマンド(印刷用)”を参照してください。

注意

WindowsのUAC機能を有効にしている環境でコマンドを実行する場合、OSにログインしたユーザの権限によって権限の昇格を求めるダイアログが表示され、対処が必要となります。

詳細は、“Systemwalker Operation Manager 運用ガイド”の、“Windowsでコマンドを実行する場合の注意事項【Windows版】”を参照してください。

注意

ジョブスケジューラコマンド実行時の注意事項

コマンド実行時には、以下の条件を確認してください。

- ジョブスケジューラサービス/デーモンが起動されていること(jobschconvertコマンドは除く)。
- 標準出力、標準エラー出力がクローズされていないこと。
- Operation Managerサーバインストール時に指定した、Operation Managerデーモンの動作言語種別とLANG環境変数が同じであること。(例えば、Solarisの場合でSystemwalker Operation Managerの動作言語種別をSJISとしてインストールしたときは、LANG環境変数にja_JP.PCKが設定されている環境から、ジョブスケジューラのコマンドを実行する必要があります。)【UNIX版】
- コマンドを実行するユーザが、ユーザごとのリソース制限(最大ファイルサイズの制限)を実施していないこと。(コマンドが正常に実行されない場合があります。また、最大ファイルサイズの制限により、スケジュール情報などのファイルが破損する可能性があります。)【UNIX版】

コマンドの実行ユーザは、Windowsサーバにログオン/UNIXサーバにログインしているユーザとなります。

4.1 コマンド一覧

ジョブスケジューラコマンドで使用するコマンドには以下のものがあります。

システム管理者権限で使用するコマンド

ジョブスケジューラが提供するシステム管理者権限で使用するコマンドおよび関連する制御文について説明します。システム管理者権限で使用するコマンドの一覧を以下に示します。

jobschsetholコマンド

カレンダの休日を登録/変更/削除します。

休日情報制御文

休日情報を設定する制御文です。

jobschsetcalコマンド

起動日雛形を登録/変更/削除します。

起動日雛形制御文

起動日雛形を設定する制御文です。

jobschcodコマンド

日変わり時刻を変更し、グループおよびジョブネットを再スケジュールします。

jobschsetprojectコマンド

プロジェクト情報を登録/変更/削除します。

プロジェクト情報制御文

プロジェクト情報を設定する制御文です。

jobschgetoutputコマンド

ジョブネットのジョブの出力を取得します。

jobschinitコマンド

ジョブスケジューラのスケジュール情報を初期化します。

stemSchDelete

プロジェクトをプロジェクト内の情報を含めて削除します。

jobschsetauthinfoコマンド【Windows版】【Linux版】

REST実行ジョブで接続するクラウドサービスの認証情報を登録/削除/参照します。

認証情報制御文

認証情報を設定する制御文です。

jobschrefstatusコマンド【Windows版】【Linux版】

災害対策運用で、ジョブ/ジョブネットの実行状況を反映します。

状態情報ファイル

出口(災対用)によりジョブ/ジョブネットの実行状態が記録されるファイルです。

jobschctlprojectコマンド【Windows版】【Linux版】

プロジェクトの無効化または有効化を行います。

一般ユーザの権限で使用するコマンド

ジョブスケジューラが提供する一般ユーザの権限で使用するコマンドおよび関連する制御文について説明します。一般ユーザの権限で使用するコマンドの一覧を以下に示します。なお、各コマンドが操作できるプロジェクトは、プロジェクトのアクセス権設定に従います。

ポイント

拡張ユーザ管理機能が有効な場合

拡張ユーザ管理機能が有効な場合は、以下のコマンドを実行したOSユーザがシステム管理者(Windows版の場合はAdministratorsグループに属するユーザ、UNIX版の場合はスーパーユーザ)かどうかでアクセス権の認証が変わります。

a. 実行したOSユーザがシステム管理者(Windows版の場合はAdministratorsグループに属するユーザ、UNIX版の場合はスーパーユーザ)の場合は、Operation Managerの管理者の権限(すべてのプロジェクトに対して更新権あり)で実行されます。

b. 実行したOSユーザがプロジェクトの所有者の場合は、所有者となっているプロジェクトに対して更新権を持っているものとして実行されます。

c. 上記以外のユーザの場合は、プロジェクトにアクセス権を持つすべてのOperation Managerユーザに対応するOSユーザの中に、コマンドを実行したOSユーザが含まれ、権限があれば、コマンドは実行されます。コマンドを実行したOSユーザが複数のOperation Managerユーザに対応づけられている場合、OSユーザは、対応づけられた複数のOperation Managerユーザのアクセス権の中で一番強い権限(更新権＞登録権・操作権＞参照権)を持ちます。なお、OSユーザに対応づけられたOperation Managerユーザに、登録権と操作権が設定されている場合、OSユーザは、両方の権限を持ちます。各プロジェクトに対するOSユーザのアクセス権は、mpprjcmdaclコマンドで確認できます。詳細は、“2.16 mpprjcmdaclコマンド【UNIX版】”を参照してください。

- jobschsetnetコマンド
- jobschsetgrpコマンド
- jobschctljobコマンド
- jobschcontrolコマンド
- jobschctlgrpコマンド
- jobschoperateコマンド
- jobschmoveコマンド
- jobschmsgclearコマンド
- jobschmsgeventコマンド
- jobschprintコマンド
- jobschprintcsvコマンド
- jobschdeftocsvコマンド
- jobschcsvtodefコマンド
- jobschchecktimeコマンド
- jobschchecklogコマンド
- chkfile/jobschchkfileコマンド
- jobschnetmemoコマンド
- jobschchkmsgevtコマンド
- jobschchknetvarコマンド
- jobschsleepコマンド

jobschsetnetコマンド

ジョブネットを登録/変更/削除/複写します。ジョブネットの起動日/メッセージ事象/一時変更情報を登録/削除することもできます。

ジョブネット制御文

ジョブネットの登録情報を設定する制御文です。

ジョブ制御文

ジョブの登録情報を設定する制御文です。

起動日制御文

ジョブネットの起動日を設定する制御文です。

メッセージ事象制御文

メッセージ事象を設定する制御文です。

一時変更制御文

ジョブネットの一時変更情報を設定する制御文です。

jobschsetgrpコマンド

グループを登録/変更/削除します。

グループ制御文

グループの登録情報を設定する制御文です。

グループ内ジョブネット制御文

グループ内のジョブネットの登録情報を設定する制御文です。

jobschctljobコマンド

ジョブを制御します。

jobschcontrolコマンド

ジョブネットを制御します。

jobschctlgrpコマンド

グループを制御します。

jobschoperateコマンド

プロジェクト内のジョブネットおよびグループを一括制御(すべて停止/すべて停止解除/すべて無効/すべて無効解除)します。

jobschmoveコマンド

グループ/ジョブネットの登録順序を変更します。

jobschmsgclearコマンド

発生済のメッセージ事象を再び待ち合わせます。

jobschmsgeventコマンド

メッセージ事象を発生させます。

jobschcancelnetコマンド【Windows版】

ジョブネット起動キューにつながっているジョブネットを削除します。

jobschdeftocsvコマンド

制御文形式のジョブネット、グループ、カレンダの休日および起動日雛形の登録情報を、CSV形式に変換します。
変換結果は標準出力(stdout)に出力します。

jobschcsvtodefコマンド

CSV形式のジョブネット、グループ、カレンダの休日および起動日雛形の登録情報を、制御文形式に変換します。
変換結果は標準出力(stdout)に出力します。

jobschchecktimeコマンド

指定した時刻がくるまで待つコマンドです。

jobschchecklogコマンド

テキスト形式のログファイルに特定の正規表現が含まれるかを調べます。

chkfile/jobschchkfileコマンド

ファイルの有無の確認、およびファイルの作成/更新/削除/サイズ変更を待つためのコマンドです。

jobschnetmemoコマンド

ジョブネットにメモを登録/削除します。メモ内容を出力/検索することもできます。

jobschchkmsgevtコマンド

メッセージ事象待ち合わせジョブで呼び出されるコマンドです。

jobschchknetvarコマンド

ジョブネット変数判定ジョブで呼び出されるコマンドです。

jobschsleepコマンド

指定した時間を待ち合わせます。

jobschrestexecコマンド【Windows版】【Linux版】

RESTインタフェースが提供されているクラウドサービス上のジョブ(API)の実行/完了待ち合わせ/強制終了を行います。

旧バージョンからの移行コマンド

jobschconvert

旧バージョンのジョブスケジューラデータベースを、本バージョンで使用可能なデータベースへ変換します。

制御文について

制御文は、メモ帳などのエディタを使用して記述してください。なお、制御文が記述されたファイルを定義ファイルと呼びます。

制御文の記述形式および注意事項を以下に示します。

記述形式

制御文は以下の形式で記述します。

```
KEYWORD                    ←制御文を示すキーワード
operand1 パラメタの並び    ←オプションとそのパラメタ
operand2 パラメタの並び
.............
operandn パラメタの並び
;                          ←制御文の終了
```

注意事項

- 制御文を示すキーワードからセミコロン“;”までを1つの制御文とします。
- キーワードやオプションは1行に1つ記述し、改行してください。
- オプションとパラメタの間は、空白1文字(半角)を挿入してください。
- 制御文の指定ミスが1つでもあれば、登録されません。
- コメントを記述することができます。コメント行には先頭に“#”を付加します。
- 各制御文の説明において、項目の長さを“バイト”で表す場合、2バイトは半角2文字/全角1文字に相当します。
- 日本語コードを含む文字列を指定する場合、Systemwalker Operaiton Managerが動作しているLANGに合わせた文字コードで指定してください。

4.2 jobschsetholコマンド

記述形式

```
[登録／変更の場合]
 jobschsethol -ent 定義ファイル
[削除の場合]
 jobschsethol -del カレンダ名
```

機能説明

「カレンダ」の休日を登録/変更/削除します。

オプション

-ent

定義ファイルの情報をもとに、「カレンダ」の休日を登録/変更します。

定義ファイル

休日情報制御文を格納したファイル名を指定します。

“休日情報制御文”については、“4.2.1 休日情報制御文”を参照してください。

-del

「システムカレンダ」以外のカレンダを削除します。

カレンダ名

削除する「システムカレンダ」以外のカレンダ名を指定します。

コマンド格納場所

Windows	Systemwalker Operation Managerインストール先ディレクトリ¥MpWalker.JM¥bin
Solaris HP-UX AIX Linux	/usr/bin

実行に必要な権限

システム管理者(Administratorsグループに属するユーザ/スーパーユーザ)権限が必要です。

注意事項

- 「カレンダ」の休日は、システム管理者(“Administrators”グループに属するユーザ/スーパーユーザ)のみ登録/変更/削除することができます。一般利用者は、jobschsetholコマンドを使用することはできません。
- ジョブネットが引用中の「カレンダ」を削除することはできません。
- 「システムカレンダ」を変更する場合、稼働日から休日に変更された場合は、その電源制御のパターン情報が失われます。また、休日から稼働日に変更された場合は、電源制御パターンとして標準パターンが設定されます(HP-UX、AIXを除く)。
- 本コマンドでは、「カレンダ」の休日を上書きして再作成します。そのため、個々に設定した休日の情報はクリアされます。必要に応じて再度、個々に休日を設定してください。
- 本コマンドは、カレンダ反映開始日の設定をすることはできません。

[ジョブスケジューラ起動パラメタの定義]ウィンドウの[利用機能 2]シートで[起動日再作成の抑止]を指定していない場合は、以下に注意してください。

- ジョブネットが引用中の「カレンダ」を変更した場合、ジョブネットの起動日は変更された「カレンダ」をもとに再設定されます。「カレンダ」の変更後にジョブネット起動日(特異日の設定)を確認・修正してください。なお、反映開始日を設定していた場合についても、指定した反映開始日に関係なく、変更した「カレンダ」をもとにジョブネットの起動日が再設定されます。
- 本コマンドは、休日カレンダの変更をサーバに依頼するだけのものです。コマンドの実行が完了しても、サーバ上の、各ジョブネットの起動日へ、変更内容の反映が完了していることを保証するものではありません。そのため、本コマンドの完了後、しばらく経過してから、各ジョブネットの起動日に反映される場合があります。
- 本コマンドでは登録済みの休日カレンダと同名の休日カレンダを登録する場合、上書きして再作成します。そのため登録済みの休日カレンダは新たに作成したカレンダ制御文の記載内容のとおりになります。
- 「年間の休日設定」([年間休日の設定]ウィンドウ)の以下の情報は設定できません。
  - [曜日選択域]の第○週の情報
  - [振替休日を有効にする]の情報
- jobschprint -rhコマンドで出力した休日情報制御文をそのまま使用して本コマンドでカレンダを登録/変更した場合、「年間の休日設定」([年間休日の設定]ウィンドウ)の情報は設定されません。

4.2.1 休日情報制御文

休日情報制御文は、休日情報を設定する制御文です。

記述形式

```
HOL                    ---------- (1)
calendarname カレンダ名 ---------- (2)   (*)   (注)
yearset 年設定          ---------- (3)   (*)
monthset 月設定         ---------- (4)   (*)
dayset 日設定           ---------- (5)   (**)
weekset 曜日設定        ---------- (6)   (**)
daycancel 日設定解除    ---------- (7)
;
```

注）calendarnameオペランドは、HOLキーワードのすぐ後に指定してください。
(*) ：必ず指定してください。
(**)：(**)の付いたオペランドのうち両方またはどちらかを指定してください。

制御文の説明

(1) HOL

休日情報制御文の開始を表すキーワードです。

(2) calendarname カレンダ名

カレンダ名(システムカレンダ/システムカレンダ以外のカレンダ名)を24バイト以内で指定します。/?*:,¥"<>|:空白、半角カナ、およびユーザ定義文字を指定することはできません。すでに設定済みのカレンダ名を指定した場合は、以前の設定情報が更新されますので注意してください。

(3) yearset 年設定

休日を設定する西暦を半角数字4文字で指定します。

今年、来年および再来年の3年分まで指定できます。毎年同じパターンの休日を指定する場合は、“EVERY”を指定します。

年設定指定例

今年が2005年の場合、今年および来年を指定する例を以下に示します。

```
yearset 2005,2006
```

(4) monthset 月設定

休日を設定する月を半角数字で指定します。複数指定する場合は、“3,5,10,...”の形式で指定します。毎月同じパターンの休日を指定する場合は、“EVERY”を指定します。

(5) dayset 日設定

休日を設定する日付を半角数字で指定します。複数指定する場合は、“1,2,15,...”の形式で指定します。

(6) weekset 曜日設定

休日を曜日単位で指定します。曜日を表す英単語の先頭3文字を半角文字で指定します。複数指定する場合は、“SUN,MON,TUE,WED,THU,FRI,SAT”の形式で指定します。

(7) daycancel 日設定解除

特定の日だけ、休日設定を解除する場合に指定します。半角文字で“YYYY,MM,DD(,DD)”の形式で指定します。日設定解除には、“EVERY”を指定することはできません。

日設定解除指定例

指定例を以下に示します。

2005年11月1日の休日設定を解除する場合

```
daycancel 2005,11,01
```

2005年11月1日、および15日の休日設定を解除する場合

```
daycancel 2005,11,01,15
```

2005年11月1日、および12月20日の休日設定を解除する場合

```
daycancel 2005,11,01
daycancel 2005,12,20
```

2005年11月3日、2006年11月3日の休日設定を解除する場合

```
daycancel 2005,11,03
daycancel 2006,11,03
```

休日情報制御文指定例

以下の年月日を休日として設定する例を以下に示します。

- 2005年11月1日
- 2005年12月20日

```
HOL
calendarname 本社カレンダ
yearset 2005
monthset 11
dayset 1
yearset 2005
monthset 12
```

```
dayset 20
;
```

注意事項

- “月設定”は、“年設定”の後に指定してください。
- “日設定”および“曜日設定”は、“月設定”の後に指定してください。
- 複数の休日パターンを指定する場合、“年設定”および“月設定”の組合せを1つの休日パターンとし、その組合せを必要なだけ指定してください。

4.3 jobschsetcalコマンド

記述形式

```
[登録／変更の場合]
  jobschsetcal -ent 定義ファイル [-sys サブシステム番号] (注)
[削除の場合]
  jobschsetcal -del 起動日雛形名 [-sys サブシステム番号] (注)
```

注)

“ [-sys サブシステム番号]”はSystemwalker Operation Manager EEのオプションです。

機能説明

起動日雛形を登録/変更/削除します。起動日雛形を利用すると、ジョブネットの起動日を簡単に設定することができます。特に、起動日に周期性を持つジョブネットを登録する場合や複数のジョブネットに類似した起動日を登録する場合に便利です。

オプション

-ent

定義ファイルの情報をもとに起動日雛形を登録/変更します。

定義ファイル

起動日雛形制御文を格納したファイル名を指定します。

“起動日雛形制御文”については、“4.3.1 起動日雛形制御文”を参照してください。

-del

起動日雛形の情報を削除します。

起動日雛形名

削除する起動日雛形名を指定します。

EE GEE **-sys サブシステム番号**

複数サブシステム運用をしているシステムで、操作対象のサブシステムを指定するためのオプションです。操作対象とするサブシステム番号を、0から9の範囲で指定します。このオプションを省略した場合、サブシステム番号0が操作対象となります。

コマンド格納場所

Windows	Systemwalker Operation Managerインストール先ディレクトリ¥MpWalker.JM¥bin

Solaris HP-UX AIX Linux	/usr/bin

実行に必要な権限

システム管理者(Administratorsグループに属するユーザ/スーパーユーザ)権限が必要です。

注意事項

- 起動日雛形は、システム管理者(“Administrators”グループに属するユーザ/スーパーユーザ)のみ登録/変更/削除することができます。一般利用者は、jobschsetcalコマンドを使用することはできません。

[ジョブスケジューラ起動パラメタの定義]ウィンドウの[利用機能 2]シートで[起動日再作成の抑止]を指定していない場合は、以下に注意してください。

- 登録済みの起動日雛形を変更した場合、その雛形を指定しているジョブネットの起動日が再設定されます。変更後に起動日を確認/修正してください。
- 本コマンドの実行の完了が、依頼した処理の完了を保証しているわけではありません。そのため、本コマンドの実行完了後、サーバ上では、各ジョブネットの起動日に反映されるまでにしばらく時間がかかる場合があります。
- 本コマンドでは登録済みの起動日雛形と同名の起動日雛形を登録する場合、上書きして再作成します。そのため登録済みの起動日雛形は新たに作成した起動日雛形制御文の記載内容のとおりになります。

4.3.1 起動日雛形制御文

起動日雛形を設定する制御文です。

記述形式

```
CAL                                    ---------- (1)
execpatname 起動日雛形名               ---------- (2) (*) (注)
monthset 月設定                        ---------- (3) (**)
dayset 日設定                          ---------- (4) (**)
weekset 曜日設定                       ---------- (5) (**)
daycancel 日設定解除                   ---------- (6) (**)
monthlyday 毎月の日付指定              ---------- (7) (**)
monthlyweek 毎月の曜日指定             ---------- (8) (**)
bnsmode 営業日の選択基準               ---------- (9) (**)
bnsbase 営業日の月初め日               ---------- (10)(**)
bnsday 営業日                          ---------- (11)(**)
bnslimit 月度内の営業日のみを起動日とする ---------- (12)(**)
;
```

注)execpatnameオペランドは、CALキーワードのすぐ後に指定してください。
(*) ：必ず指定してください。
(**)：以下のa.(毎年基準)/b.(毎月基準)/c.(営業日基準)からどれか1つの組み合わせで指定してください。
  a. monthset、dayset、weekset、daycancel (monthsetの指定は必須、daysetとweeksetは
     どちらかまたは両方、daycancelはdaysetを指定した場合のみ指定可能)
  b. monthlydayとmonthlyweekの組み合わせ(どちらかまたは両方)
  c. bnsmode、bnsbase、bnsday、bnslimit (bnsmodeおよびbnsbaseはこのオペランド群を指定するときは必須)

制御文の説明

(1) CAL

起動日雛形制御文の開始を表すキーワードです。

(2) execpatname 起動日雛形名

起動日雛形名を24バイト以内で指定します。空白、半角カナおよびユーザ定義文字を含むことはできません。なお、すでに設定済みの起動日雛形名を指定した場合、以前の設定情報が消去されるので注意してください。

(3) monthset 月設定

毎年基準を利用して起動日雛形情報を定義する場合に、起動日雛形を設定する月を半角数字で指定します。必ず指定してください。複数指定する場合は、“3,5,10,..”の形式で指定します。毎月同じパターンの起動日を指定する場合は、“EVERY”を指定します。

(4) dayset 日設定

毎年基準を利用して起動日雛形情報を定義する場合に、起動日雛形を設定する日付を半角数字で指定します。複数指定する場合は、“1,2,15,21,...”の形式で指定します。

(5) weekset 曜日設定

毎年基準を利用して起動日雛形情報を定義する場合に、起動日雛形を曜日単位で指定します。曜日を表す英単語の先頭3文字を半角文字で指定します。複数指定する場合は、“SUN,MON,TUE,WED,THU,FRI,SAT”の形式で指定します。

(6) daycancel 日設定解除

毎年基準を利用して起動日雛形情報を定義する場合に、特定の日だけ、起動日雛形設定を解除する場合に指定します。解除したい日の月および日付をそれぞれ半角数字2文字で“MMDD”の形式で指定します。

日設定解除指定例

指定例を以下に示します。

1月1日に設定した起動日雛形を解除する場合

```
daycancel 0101
```

1月1日、10日および2月10日に設定した起動日雛形を解除する場合

```
daycancel 0101,0110,0210
```

(7) monthlyday 毎月の日付指定

毎月基準を利用して起動日雛形情報を定義する場合に、毎月のジョブネットを起動する日付を指定します。毎月DD日を起動日として設定する場合は、“DD”と指定します。複数の起動日を設定することができます。なお、毎月月末に起動する場合は“LAST”と指定します。

(8) monthlyweek 毎月の曜日指定

毎月基準を利用して起動日雛形情報を定義する場合に、毎月のジョブネットを起動する曜日を週ごとに指定します。1つのオペランドの指定で1つの週しか設定できません。複数の週を設定する場合は、本オペランドを複数回指定します。指定する形式は、何週目かを表す数値を0～6(第1～第5まで。0は毎週の指定。6は月の最終週)の範囲で指定し、その後、曜日を表す英単語の先頭3文字を半角文字で指定します。“5,SUN,MON,TUE,WED,THU,FRI,SAT”のような形式で指定します。

(9) bnsmode 営業日の選択基準

営業日基準を利用して起動日雛形情報を定義する場合に、月初め、月締めのどちらを基準とするかを指定します。

START

営業日を月初めから数えます。

END

営業日を月締めから逆算します。

(10) bnsbase 営業日の月初め日

営業日基準を利用して起動日雛形情報を定義する場合に、月初め日を指定します。“0”は指定できません。

1〜28

1日から28日です。

LAST

月末です。

(11) bnsday 営業日

営業日基準を利用して起動日雛形情報を定義する場合に、営業日を指定します。複数指定した場合には、“,”(コンマ)で区切って指定します。“0”は指定できません。

1〜31

第1営業日(月初め)から数えた営業日です。bnsmodeオペランドで“END”を指定した場合は指定できません。

-1〜-31

第-1営業日(月締め)から逆算した営業日です。bnsmodeオペランドで“START”を指定した場合は指定できません。

(12) bnslimit 月度内の営業日のみを起動日とする

営業日基準を利用して起動日雛形情報を定義する場合に、“月度内の営業日のみを起動日とする”の有無(ON/OFF)を指定します。bnsmodeオペランドが指定されている場合の省略値は“ON”です。

ON

“月度内の営業日のみを起動日とする”を指定します。

OFF

“月度内の営業日のみを起動日とする”を指定しません。

起動日雛形制御文指定例

起動日雛形として以下の月日を起動日に設定する例を以下に示します。

- 1月1日
- 2月10日

```
CAL
execpatname サンプル
monthset 1
dayset 1
monthset 2
dayset 10
;
```

注意事項

- “日設定”および“曜日設定”は、“月設定”の後に指定してください。
- “日設定”および“曜日設定”に複数のパターンを指定する場合、その都度“月設定”を必ず指定してください。

4.4 jobschcodコマンド

記述形式

```
[設定／変更の場合]
```

```
jobschcod 日変わり時刻 [-sys サブシステム番号]
[設定解除の場合]
jobschcod -off -sys サブシステム番号
```

機能説明

日変わり時刻を変更します。

オプション

日変わり時刻

日変わり時刻は4桁の数値“hhmm”で設定します。“hh”は“00～23”(時)、“mm”は“00～59”(分)の形式で設定します。

指定例

日変わり時刻を午前9時5分に設定する場合の指定例を以下に示します。

```
jobschcod 0905
```

EE GEE **-sys サブシステム番号**

複数サブシステム運用をしているシステムで、操作対象のサブシステムを指定するためのオプションです。操作対象とするサブシステム番号を、0から9の範囲で指定します。設定/変更時にこのオプションを省略した場合、システムの日変わり時刻が操作対象となります。

-off

指定したサブシステム番号の日変わり時刻の設定を解除します。

コマンド格納場所

Windows	Systemwalker Operation Managerインストール先ディレクトリ¥MpWalker.JM¥bin
Solaris HP-UX AIX Linux	/usr/bin

実行に必要な権限

システム管理者(Administratorsグループに属するユーザ/スーパーユーザ)権限が必要です。

注意事項

- 日変わり時刻は、システム管理者(“Administrators”グループに属するユーザ/スーパーユーザ)のみ変更できます。一般利用者は、jobschcodコマンドを使用することはできません。
- システムの日変わり時刻は「システムカレンダ/SYSTEM_CALENDAR」に格納されています。jobschcodコマンドの利用により、ジョブスケジューラのスケジュール以外にも、電源スケジュールにも影響を与えるので、変更には注意が必要です(HP-UX、AIXを除く)。

4.5 jobschsetprojectコマンド

記述形式

[登録／変更の場合]
 jobschsetproject -ent 定義ファイル [-sys サブシステム番号]　(注)
[削除の場合]
 jobschsetproject -del プロジェクト名 [-sys サブシステム番号]　(注)

注)

“ [-sys サブシステム番号]”はSystemwalker Operation Manager EEのオプションです。

機能説明

プロジェクト情報を登録/変更/削除します。

プロジェクト内ジョブネットのアクセス(更新/操作/参照)可能ユーザも定義します。

オプション

-ent

定義ファイルの内容をもとに、プロジェクト情報を登録/変更します。

定義ファイル

プロジェクト情報制御文を格納したファイル名を指定します。

“プロジェクト情報制御文”については、“4.5.1 プロジェクト情報制御文”を参照してください。

-del

プロジェクト情報を削除します。

プロジェクト名

削除するプロジェクト名を指定します。プロジェクト内の定義(グループ/ジョブネット)がすべて削除されていないと、プロジェクトは削除できません。

EE GEE

-sys サブシステム番号

複数サブシステム運用をしているシステムで、操作対象のサブシステムを指定するためのオプションです。操作対象とするサブシステム番号を、0から9の範囲で指定します。このオプションを省略した場合、サブシステム番号0が操作対象となります。

コマンド格納場所

Windows	Systemwalker Operation Managerインストール先ディレクトリ¥MpWalker.JM¥bin
Solaris HP-UX AIX Linux	/usr/bin

実行に必要な権限

システム管理者(Administratorsグループに属するユーザ/スーパーユーザ)権限が必要です。

注意事項

- プロジェクトは、システム管理者(“Administrators”グループに属するユーザ/スーパーユーザ)のみ登録/変更/削除することができます。一般利用者は、jobschsetprojectコマンドを使用することはできません。
- ドメインユーザで本コマンドを実行する場合、そのドメインユーザはローカル環境のAdministratorsグループに直接所属する必要があります。

4.5.1　プロジェクト情報制御文

プロジェクト情報を設定する制御文です。

記述形式

```
PRJ                          ---------- (1)
projectname プロジェクト名    ---------- (2) (*)  (注1)
username 所有者(OSユーザ)                          ---------- (3) (**) (注2)
manage_user 更新権ユーザーID(OSユーザ)             ---------- (4)       (注2)
change_user 登録権ユーザーID(OSユーザ)             ---------- (5)       (注2)
operate_user 操作権ユーザーID(OSユーザ)            ---------- (6)       (注2)
monitor_user 参照権ユーザーID(OSユーザ)            ---------- (7)       (注2)
;
```

注1)projectnameオペランドは、PRJキーワードのすぐ後に指定してください。
注2)ユーザプリンシパル名を指定することはできません。
(*) ：必ず指定してください。
(**)：登録時には必ず指定してください。

制御文の説明

(1) PRJ

プロジェクト情報制御文の開始を表すキーワードです。

(2) projectname プロジェクト名

プロジェクトの名前を50バイト以内で指定します。/?*:,¥"<>|:'.空白、半角カナおよびユーザ定義文字を指定することはできません。

また、1文字目に # を指定することはできません。

注：

プロジェクト内のすべてのジョブネットの属性が以下の場合は、1文字目に # を指定することができます。

－ 旧バージョン互換（旧・標準）属性

大文字/小文字の区別はありません。【Windows版】

すでに設定済みのプロジェクト名が指定された場合は、以前の設定情報と置き換わりますので注意してください。

英大文字と英小文字は区別されますが、英大文字と英小文字が異なるだけで同じつづりのプロジェクト名を別々に登録した場合、ポリシー抽出でスケジュール情報を正しく抽出することができません。【UNIX版】

出口プログラムを利用する場合、以下の文字は指定できません。【Windows版】

& = ^ %

(3) username 所有者(OSユーザ)

プロジェクトの所有者となるユーザーIDを指定します。

ジョブ実行制御により偽装オプションが指定された場合、ジョブの実行権限を偽装するために使用するユーザーIDにもなります。ドメイン名¥ユーザーIDの形式で指定します。長さは36バイト以内(ドメイン名15バイト+¥記号1バイト+ユーザーID 20バイト)で指定します。ドメイン名部分を省略した場合は、ジョブスケジューラサービス/デーモンの存在するコンピュータの省略値に従います。なお、グループ名(ローカルグループおよびグローバルグループ)は指定できません。また、空白を含むユーザーIDも指定できません。【Windows版】

長さは20バイト以内で指定します。空白を含むユーザーIDは指定できません。拡張ユーザ管理機能【UNIX版】が有効な場合についても、所有者には必ずOSユーザを指定してください。

(4) manage_user 更新権ユーザーID(OSユーザ)

更新権を与えるユーザーIDを指定します。

ドメイン名¥ユーザーIDの形式で指定します。ドメイン名(¥まで)は省略できます。長さは36バイト(ユーザーIDのみを指定する場合には20バイト)以内で指定します。【Windows版】

拡張ユーザ管理機能が有効な場合は、Operation Managerユーザを指定します。長さは20バイト以内で指定します。【UNIX版】

グループ名も指定可能です。複数指定する場合は、ユーザーIDとユーザーIDの間をコンマ(,)で区切ってください。空白を含むユーザーIDは指定できません。

すでに登録されている更新権ユーザをすべて削除したい場合には、manage_userに指定するユーザーIDを省略してください。

(5) change_user 登録権ユーザーID(OSユーザ)

登録権を与えるユーザーIDを指定します。

ドメイン名¥ユーザーIDの形式で指定します。ドメイン名(¥まで)は省略できます。長さは36バイト(ユーザーIDのみを指定する場合には20バイト)以内で指定します。【Windows版】

拡張ユーザ管理機能が有効な場合は、Operation Managerユーザを指定します。長さは20バイト以内で指定します。【UNIX版】

グループ名も指定可能です。複数指定する場合は、ユーザーIDとユーザーIDの間をコンマ(,)で区切ってください。空白を含むユーザーIDは指定できません。

すでに登録されている登録権ユーザをすべて削除したい場合には、change_userに指定するユーザーIDを省略してください。

(6) operate_user 操作権ユーザーID(OSユーザ)

操作権を与えるユーザーIDを指定します。

ドメイン名¥ユーザーIDの形式で指定します。ドメイン名(¥まで)は省略できます。長さは36バイト(ユーザーIDのみを指定する場合には20バイト)以内で指定します。【Windows版】

拡張ユーザ管理機能が有効な場合は、Operation Managerユーザを指定します。長さは20バイト以内で指定します。【UNIX版】

グループ名も指定可能です。複数指定する場合は、ユーザーIDとユーザーIDの間をコンマ(,)で区切ってください。空白を含むユーザーIDは指定できません。

すでに登録されている操作権ユーザをすべて削除したい場合には、operate_userに指定するユーザーIDを省略してください。

(7) monitor_user 参照権ユーザーID(OSユーザ)

参照権を与えるユーザーIDを指定します。

ドメイン名¥ユーザーIDの形式で指定します。ドメイン名(¥まで)は省略できます。長さは36バイト(ユーザーIDのみを指定する場合には20バイト)以内で指定します。【Windows版】

拡張ユーザ管理機能が有効な場合は、Operation Managerユーザを指定します。長さは20バイト以内で指定します。【UNIX版】

グループ名も指定可能です。複数指定する場合は、ユーザーIDとユーザーIDの間をコンマ(,)で区切ってください。空白を含むユーザーIDは指定できません。

すでに登録されている参照権ユーザをすべて削除したい場合には、monitor_userに指定するユーザーIDを省略してください。

注意事項

- 制御文全体に対して、同一のユーザーIDやグループ名を複数回指定することはできません。
- すでにアクセス権を設定済みのユーザに対して、別のアクセス権を設定した場合、新しいアクセス権で上書きされます。

プロジェクト情報制御文指定例

プロジェクト名に総務部門、所有者にishikawa、更新権ユーザーIDにyamada、登録権ユーザーIDにnakamura、操作権ユーザーIDにhayashi、参照権ユーザ名にsuzukiおよびsatoを設定する例を以下に示します。

```
PRJ
projectname 総務部門
username ishikawa
manage_user yamada
change_user nakamura
operate_user hayashi
monitor_user suzuki,sato
;
```

4.6 jobschgetoutputコマンド

記述形式

```
jobschgetoutput -m メッセージ -f ジョブ出力取得ファイル名 [ -s 最大ファイルサイズ]
```

機能説明

引数で指定したメッセージが示すジョブネットのジョブの出力を取得します。

Systemwalker Centric Managerと連携する場合のみ使用できます。自動アクションのメール通知の実行コマンドとして使用します。

本コマンドが扱えるメッセージは、[ジョブスケジューラ起動パラメタの定義]ウィンドウの[イベント出力]シートで、[正常終了]、[異常終了]、[強制終了]、[実行中断]、[疑似正常]、 [自動通知/対処をする]チェックボックスを選択した結果出力されるメッセージです。

オプション

-m メッセージ

Systemwalker Centric Managerで監視するSystemwalker Operation Managerのメッセージを指定します。本コマンドは、Systemwalker Centric Managerの自動アクションのメール通知の実行コマンドとして使用するので、%MSGを指定します。

-f ジョブ出力取得ファイル名

メッセージに含まれるジョブネット内ジョブの出力情報を格納するファイルを絶対パスで指定します。

Systemwalker Centric Managerの自動アクションのメール通知の実行コマンドとして使用した際に、指定したファイルに情報を格納します。

-s 最大ファイルサイズ

ジョブ出力取得ファイルの最大バイト数を指定します。1000～10000000の間の数値を指定します。本オプションを指定しない場合、最大バイト数は20000になります。

復帰値

復帰値	意味
0	処理が正常に終了したことを示します。
11	メッセージがこのコマンドで扱えない形式であることを示します。
12	ジョブネットの情報の読み込みに失敗したことを示します。

復帰値	意味
13	ジョブの出力の読み込みに失敗したことを示します。
14	ジョブ出力取得ファイルの書き込みに失敗したことを示します。
21	オプションの指定に誤りがあることを示します。
22	システムまたはユーザの環境に関するエラーが発生したことを示します。
31	Systemwalker Operation Managerの環境に関する内部エラーが発生したことを示します。
35	他の内部エラーが発生したことを示します。

コマンド格納場所

Windows	Systemwalker Operation Managerインストール先ディレクトリ¥MpWalker.JM¥bin
Solaris HP-UX AIX Linux	/usr/bin

実行に必要な権限/実行環境

- システム管理者(Administrators グループに属するユーザ/スーパーユーザ)権限が必要です。
- Systemwalker Operation Managerサーバと以下の製品が同じマシンで動作していなければなりません。
  - Systemwalker Centric Manager V13.3.0以降(運用管理サーバ、部門管理サーバ、業務サーバのいずれか)

使用例

自動アクションのメール通知の実行コマンドとしてジョブの出力を取得し、最大ファイルサイズをデフォルト、格納先を/var/tmp/jobschoutput.logにする場合、UNIX版では、以下のように定義します。

```
/usr/bin/jobschgetoutput -m %MSG -f /var/tmp/jobschoutput.log
```

実行結果/出力形式

ジョブ出力取得ファイルの形式を示します。

```
Subsystem Number:サブシステム番号
Project Name:プロジェクト名
Job Net Name:ジョブネット名
Job Net Comment:ジョブネット名称
Job Name:
ジョブ番号:ジョブ名(ジョブの状態)
ジョブ番号:ジョブ名(ジョブの状態)
・
・

Job Output(ジョブの出力の割合):
ジョブ番号 ジョブの出力
ジョブ番号 ジョブの出力
ジョブ番号 ジョブの出力
・
・
```

サブシステム番号:

Standard Editionでは0、Enterprise Edition、Global Enterprise Editionではメッセージが示すジョブネットが属するサブシステム番号です。

プロジェクト名:

メッセージが示すジョブネットが属するプロジェクト名です。

ジョブネット名:

メッセージが示すジョブネットのジョブネット名です。

ジョブネット名称:

メッセージが示すジョブネットのジョブネット名称です。

ジョブ番号:

メッセージが示すジョブネット内のジョブの一意の番号です。

ジョブ名:

メッセージが示すジョブネット内のジョブの名前です。ジョブネットの実行属性やジョブの種類によって、以下のようになります。

- ジョブがグローバルサーバMSPジョブまたはグローバルサーバXSPジョブの場合は、ホストのジョブ名／業務IDです(SymfoNETに未定義の場合はホストのジョブ名、定義済の場合は業務ID)。
- ジョブがバッチフレームワークジョブの場合は、ジョブ名が設定されているときはジョブ名、設定されていないときはバッチジョブ定義名です。
- ジョブが子ジョブネットの場合は、子ジョブネットのジョブネット名です。
- ジョブがリンクジョブネットの場合は、リンクジョブネットのジョブネット名です。
- ジョブネットの実行属性が“Interstage”の場合は、ワークユニット名です。
- ジョブネットの実行属性が“旧バージョン互換(旧・標準)”の場合は、コマンドです。
- 上記以外は、ジョブ名が設定されていればジョブ名、設定されてなければコマンドです。パラメタが設定されているときは、コマンドを名前にする場合だけスペースを間に挟みつなげます。コマンドやパラメタに含まれるジョブ定義変数は置き換えません。

ジョブの状態:

以下のように表します。

- 正常終了：“Completed”
- 異常終了：“Abended”
- 強制終了：“Canceled”
- 警告：“Warning”
- 実行中：“Executing”
- 実行待ち：“Waiting”
- 停止中：“Paused”
- 無効状態：“Disabled”
- 疑似正常：“Pseudo-normal”
- 実行遅延：“Delayed”

ジョブの出力の割合:

ジョブの出力がジョブ出力取得ファイルの最大ファイルサイズに収まった場合は“all”、収まらなかった場合は格納できた割合を“0%”～“99%”で表します。割合については、小数点以下を切り捨てた値となるため正確ではありません。ジョブの出力が格納されている割合を知るための目安としてください。

ジョブの出力:

行単位のジョブの出力です。１つのジョブが複数行出力した場合、ジョブ番号とジョブの出力の対が、その分だけジョブ出力取得ファイルに格納されます。

注意事項

- ジョブ出力取得ファイルを残す必要がない場合は、Systemwalker Centric Managerの自動アクションのメール通知の実行コマンドの設定時に、送信ファイルの削除を有効にしてください。
- 最大ファイルサイズを超える場合、ジョブ出力取得ファイルは以下のようになります。

  プロジェクト名やジョブネット名などのジョブの出力以前の部分(以後“ヘッダ部”と呼びます)が最大ファイルサイズを超える場合:

  先頭から最大ファイルサイズを超えない範囲を格納します。

  行単位に格納し、途中で切れる行については格納しません。

  ヘッダ部とジョブの出力を合わせたサイズが最大ファイルサイズを超える場合:

  超えたサイズ分のデータを先頭から削除した後にジョブの出力を格納します。

  行単位に格納し、途中で切れる行については格納しません。

  なお、ジョブの出力は、ジョブネットごとに終了順で格納されています。
- 本コマンドは、ジョブの出力の先頭1000行を順次読み込むことによってヘッダ部の文字コードを決めます。UTF-8、SJIS、EUCのいずれか判断できた時点で文字コードを決定します。判断できなかった場合はUTF-8とします。UTF-8、SJIS、EUCは同じ文字に同じコードを使用している場合があるため、正確に判断できないことがあります。この場合、ジョブ出力取得ファイルを表示したときに文字化けすることがあります。
- ジョブの出力は文字コードを変換せずに格納します。ジョブの出力の文字コードが混在する場合、ジョブ出力取得ファイルを表示したときに文字化けすることがあります。
- ジョブネットが終了してから本コマンドが実行される間にジョブネットを操作／編集した場合、ジョブ出力取得ファイルのヘッダ部とジョブの出力の内容が一致しないことや本コマンドが復帰値13で終了することがあります。
- ジョブネットが終了してからこのコマンドが実行される間にジョブネットを削除した場合、本コマンドは復帰値12で終了します。

4.7 jobschinitコマンド

記述形式

```
jobschinit [-sys { all | サブシステム番号 } ] (注)
```

注)

“[-sys {all|サブシステム番号}]”はSystemwalker Operation Manager EEのオプションです。

機能説明

ジョブスケジューラのスケジュール情報（プロジェクト・グループ・ジョブネット）を初期化(削除)します。

指定したサブシステム内のすべてのスケジュール情報を削除します。

オプション

EE GEE **-sys { all | サブシステム番号 }**

複数サブシステムを運用しているシステムで、操作対象のサブシステムを指定するためのオプションです。このオプションを省略した場合、サブシステム番号0が操作対象となります。

all

すべてのサブシステムを操作対象とします。

サブシステム番号

操作対象とするサブシステム番号を、0から9の範囲で指定します。

復帰値

復帰値	意味
0	正常終了
1	異常終了。エラーの内容はエラーメッセージを確認してください。

コマンド格納場所

Windows	Systemwalker Operation Managerインストール先ディレクトリ¥MpWalker.JM¥bin
Solaris Linux	/usr/bin

実行に必要な権限/実行環境

システム管理者(Administratorsグループに属するユーザ/スーパーユーザ)権限が必要です。

使用例

```
jobschinit
```

出力形式

出力例【EE版】

```
Initialize schedule information of subsystem [0]? (Y/N)
```

出力例【SE版】

```
Initialize schedule information? (Y/N)
```

注意事項

- 実行中のジョブネットまたは実行中のグループが存在するプロジェクトは削除できません。
- ジョブスケジューラのサービス/デーモンが起動している必要があります。
- 本コマンドは内部的にstemSchDeleteコマンドを使用しているため、stemSchDeleteコマンドのエラーが発生する場合があります。異常発生時には、出力されたエラーメッセージより対処を行い、コマンドを再実行してください。

4.8 stemSchDeleteコマンド

記述形式

```
stemSchDelete プロジェクト名 [-mod mode] [-sys サブシステム番号] (注)
```

注)

“[-sys サブシステム番号]”はSystemwalker Operation Manager EEのオプションです。

機能説明

指定されたプロジェクトを削除します。プロジェクトは、プロジェクト内のジョブネット情報の有無にかかわらず削除されます。ただし、-mod で指定した状態のジョブネットが存在する場合は削除されません。

オプション

プロジェクト名

削除するプロジェクト名を50バイト以内で指定します。

ジョブスケジューラに登録されているプロジェクト名を指定します。

マスタスケジュール管理機能を利用している場合は、以下を指定します。

- マスタ情報の場合：“プロジェクト名”
- スケジュール情報の場合：“プロジェクト名+MMDD”
- 運用変更用プロジェクトの場合：“プロジェクト名=MMDD”

-mod *mode*

削除時のモードを0または1で指定します。モードによりチェックするジョブネットの状態が変わります。

0:

実行中のジョブネットまたは実行中のグループが存在するとき、プロジェクトを削除しません。

1:

未完了のジョブネットまたはグループが存在するとき、プロジェクトを削除しません。未完了とは、以下の状態を示します。

- ー 実行中、警告、または停止状態である。
- ー 開始予定日時が設定されていて、実績がない。(終了していない)
- ー 異常終了または強制終了状態である。(確認操作の有効が設定されており、かつ、確認操作が実施されていない場合)

このオプションの省略値は、0です。

EE GEE

-sys サブシステム番号

複数サブシステムを運用しているシステムで、操作対象のサブシステムを指定するためのオプションです。削除したいプロジェクトが存在するサブシステム番号を0から9の範囲で指定します。このオプションを省略した場合、サブシステム0が操作対象となります。

復帰値

code	意味	対処
0	正常終了	ー
1	Usage表示	オプションの設定の仕方を確認してください。
2	コマンドの実行権限がありません。	本コマンドの実行にはシステム管理者(Administratorsグループに属するユーザ/スーパーユーザ)の権限が必要です。
3	システムエラーが発生しました。	システムの環境を確認してください。
4	コード系が異なるため実行できません。	Systemwalker Operation Managerのコード系とコマンド実行環境のコード系が異なります。Systemwalker

code	意味	対処
		Operation Managerのコード系と同じ環境で実行してください。
5	DBへのアクセスに失敗しました。	DBのオープン、読み書きの処理に失敗しました。以下の確認をしてください。 ・イベントログ/SYSLOGでMpJobschのエラーをチェックしてください。 ・ウィルスチェックまたはバックアップソフトなどで他のプロセスがDBにアクセスしている可能性があるため確認してください。
6	ジョブスケジューラのサービス/デーモンが起動していないか、起動中です。	サービス/デーモンを起動してください。すでに起動している場合は起動中か正しく起動されていない可能性があるため、しばらく待つか、再起動してください。
7	指定されたプロジェクトが存在しません。	指定したプロジェクト名に誤りがないか確認してください。
8	グループまたはジョブネットが実行中のため削除できません。	実行中のグループまたはジョブネットが終了してから再度コマンドを実行してください。
11	コマンドはACLマネージャライブラリの読み込みに失敗しました。	以下のファイルが存在しているか確認してください。存在している場合、ファイルが壊れている可能性があります。インストールしなおしてください。 【Windows版】: Systemwalkerインストールディレクトリ¥mpwalker¥bin¥f3crsapi.dll 【Solaris版/AIX版/Linux版】: /opt/systemwalker/lib/libmpfwsecjm.so 【HP-UX版】: /opt/systemwalker/lib/libmpfwsecjm.sl
12	持ち越しジョブネットがあるため、プロジェクトの削除はできません。	持ち越しジョブネットを無効にするなどの処置を実施後、コマンドを実行してください。
13	グループまたはジョブネットが操作中/保存中のため削除することができません。	操作中/保存中のグループまたはジョブネットが終了してから再度コマンドを実行してください。
14	タイムアウト	Systemwalker Operation Managerの負荷を確認し、再度、コマンドを実行してください。
15	終了処理ジョブネット(JSHENDまたはJSHFORCE)が開始したため削除できません。	ジョブスケジューラの再起動後、コマンドを再実行してください。

コマンド格納場所

Windows	Systemwalker Operation Managerインストールディレクトリ¥mpwalker.jm¥bin

Solaris HP-UX AIX Linux	/opt/FJSVstem/bin

実行に必要な権限

システム管理者(Administratorsグループに属するユーザ/スーパーユーザ)の権限が必要です。

実行結果／出力形式

正常終了 : なし

異常終了 : エラー内容は標準出力に表示されます。

4.9 jobschsetauthinfoコマンド【Windows版】【Linux版】

記述形式

```
[登録の場合]
  jobschsetauthinfo -add 定義ファイル [-sys サブシステム番号](注)
[削除の場合]
  jobschsetauthinfo -del 認証情報名 [-sys サブシステム番号] (注)
[参照の場合]
  jobschsetauthinfo -show [認証情報名] [-sys サブシステム番号] (注)
```

注)

“[-sys サブシステム番号]”はSystemwalker Operation Manager EEのオプションです。

機能説明

REST実行ジョブがクラウドサービスに接続するための認証情報を登録/削除/参照します。

オプション

-add

定義ファイルの情報をもとに認証情報を登録します。なお、同一の認証情報名がすでに存在する場合は上書きされます。

定義ファイル

認証情報制御文(JSON形式)を格納したファイル名を指定します。

複数の認証情報を登録するときは、認証情報をJSONのリスト形式で複数記述します。

認証情報制御文の詳細については“4.9.1 認証情報制御文【Windows版】【Linux版】”を参照してください。

-del

指定された認証情報名の認証情報を削除します。

-show

指定された認証情報名の認証情報を参照します。

認証情報名を指定しない場合、登録されている認証情報の一覧を表示します。

以下の情報は、値が“***”で表示されます。

- Basic認証のパスワード
- ベアラートークン認証の認証トークン

- AWS_SigV4認証のシークレットアクセスキー
- OAuth2.0認証のクライアントシークレット
- Oracle Cloud InfrastructureのAPIキーベース認証の秘密鍵の値、秘密鍵のパスフレーズ
- プロキシサーバ認証のパスワード
- 任意に設定したキーの値

認証情報名

削除または参照する認証情報の認証情報名を指定します。

EE GEE **-sys サブシステム番号**

複数サブシステムを運用しているシステムで、操作対象のサブシステムを指定するためのオプションです。削除したいプロジェクトが存在するサブシステム番号を0から9の範囲で指定します。このオプションを省略した場合、サブシステム0が操作対象となります。

復帰値

復帰値	意味
0	正常終了
0以外	異常終了。エラーの内容はエラーメッセージを確認してください。

コマンド格納場所

Windows	Systemwalker Operation Managerインストール先ディレクトリ¥MpWalker.JM¥bin
Linux	/usr/bin

実行に必要な権限

システム管理者(Administratorsグループに属するユーザ/スーパーユーザ)の権限が必要です。

使用例

登録の場合

```
jobschsetauthinfo -add /tmp/authinfo.json
```

削除の場合

```
jobschsetauthinfo -del test1
```

参照の場合

```
jobschsetauthinfo -show test1
```

実行結果／出力形式

- -add、-delオプションを指定した場合の出力例

  標準出力なし

- -showオプションを指定した場合の出力例

```
jobschsetauthinfo -show test1
{
```

```
"name": "test1",
"authType": "basic",
"basicUser": "testuser1",
"basicPasswd": "***"
}
```

注意事項

- 定義ファイルの内容に誤りがあった場合、jobschsetauthinfoコマンド実行時にエラーになります。
- マスタスケジュール管理機能でスケジュール情報の分散を行う場合、jobschsetauthinfoコマンドで登録した認証情報は配付されません。そのため、あらかじめスケジュールサーバ側で必要な認証情報を設定してください。
- jobschsetauthinfoコマンドで登録した認証情報は、ジョブスケジューラのスケジュール情報の一部として、ポリシー情報の抽出/配付の対象となります。

4.9.1 認証情報制御文【Windows版】【Linux版】

REST実行ジョブがクラウドサービスに接続するための認証情報を設定する制御文です。

設定可能な認証種別は以下です。

- Basic認証
- ベアラートークン認証
- AWS_SigV4認証
- Microsoft Entra ID認証(OAuth2.0 Client Credentials認証)
- SAPクラウドサービス認証(OAuth2.0 Client Credentials認証)
- Oracle Cloud InfrastructureのAPIキーベース認証
- プロキシサーバ認証
- 任意のキー、値の認証

記述形式

JSON形式、文字コードはUTF-8で記載します。データの型はAWS_SigV4認証のsigv4Headerを除き、すべてstring型で指定します。sigv4Headerはobject型で指定します。string型の値にダブルクォーテーションやバックスラッシュ、および、制御文字などを含める場合、JSONのフォーマットに従いエスケープした文字列を指定してください。

リスト形式にすることで、複数の認証情報をまとめて登録することもできます。

以下に、認証種別ごとの記述形式について説明します。

Basic認証の場合の制御文

Basic認証でリクエストするときに本認証種別を選択します。

本認証種別で設定した認証情報名は、REST実行ジョブの接続情報ファイルの認証情報名パラメタで指定します。

```
{
  "name": "認証情報名",                          ------(1)  (*)
  "authType": "認証種別",                        ------(2)  (*)
  "basicUser": "Basic認証のユーザー名",          ------(3)  (*)
  "basicPasswd": "Basic認証のパスワード",        ------(4)  (*)
  "memo": "メモ",                                ------(5)
  "任意のキー": "任意の値"                       ------(6)
}
```

(*)：必ず指定してください。

(1) name 認証情報名

登録する認証情報名を指定します。

64バイト以内で指定します。

半角英数字、“_”(半角アンダーバー)、“-”(半角ハイフン)を利用できます。ただし、先頭に“-”(半角ハイフン)は利用できません。

(2) authType 認証種別

basicを指定します。

(3) basicUser Basic認証のユーザー名

Basic認証の認証ユーザ名を指定します。

1024バイト以内で指定してください。

(4) basicPasswd Basic認証のパスワード

Basic認証の認証パスワードを指定します。

1024バイト以内で指定してください。

(5) memo メモ

登録する認証情報のメモ情報を指定します。

1024バイト以内で指定してください。

(6) 任意のキー 任意の値

REST実行ジョブに指定するクエリパラメタやHTTPリクエストヘッダの値を接続情報ファイルに直接記載したくない場合、本オペランドで任意のキーおよび値を指定します。接続情報ファイルでの本オペランドの指定方法については、“20.9.1 接続情報ファイル”を参照してください。

本オペランドはキーの名前が一意であれば複数設定できます。

任意のキー、任意の値は1024バイト以内で指定してください。

なお、任意のキーの先頭文字に“-”(半角ハイフン)は利用できません。

ベアラートークン認証の場合の制御文

ベアラートークン認証でリクエストするときに本認証種別を選択します。

本認証種別で設定した認証情報名は、REST実行ジョブの接続情報ファイルの認証情報名パラメタで指定します。

```
{
  ”name”: ”認証情報名”,                    ------(1)  (*)
  ”authType”: ”認証種別”,                  ------(2)  (*)
  ”token”: ”認証トークン”,                 ------(3)  (*)
  ”memo”: ”メモ”,                          ------(4)
  ”任意のキー”: ”任意の値”                 ------(5)
}
```

(*)：必ず指定してください。

(1) name 認証情報名

登録する認証情報名を指定します。

64バイト以内で指定します。

半角英数字、“_”(半角アンダーバー)、“-”(半角ハイフン)を利用できます。ただし、先頭に“-”(半角ハイフン)は利用できません。

(2) authType 認証種別

bearerを指定します。

(3) token 認証トークン

認証トークンを指定します。

(4) memo メモ

登録する認証情報のメモ情報を指定します。

1024バイト以内で指定してください。

(5) 任意のキー 任意の値

REST実行ジョブに指定するクエリパラメタやHTTPリクエストヘッダの値を、接続情報ファイルに直接記載したくない場合、本オペランドで任意のキーおよび値を指定します。接続情報ファイルでの本オペランドの指定方法については、“20.9.1 接続情報ファイル”を参照してください。

本オペランドはキーの名前が一意であれば複数設定できます。

任意のキー、任意の値は1024バイト以内で指定してください。

なお、任意のキーの先頭文字に“-”(半角ハイフン)は利用できません。

AWS_SigV4認証の場合の制御文

Amazon Web ServicesのAPIにリクエストするときに本認証種別を選択します。

AWS_SigV4認証では、アクセスキーを用いて署名バージョン4による署名を作成し、リクエストの認証情報に追加することでAmazon Web ServicesのAPIにリクエストします。

Amazon Web Servicesの署名バージョン4の詳細については、Amazon Web Servicesの公式ドキュメントを参照してください。

本認証種別で設定した認証情報名は、REST実行ジョブの接続情報ファイルの認証情報名パラメタで指定します。

```
{
  ”name”: ”認証情報名”,                          ------(1)  (*)
  ”authType”: ”認証種別”,                        ------(2)  (*)
  ”accKey”: ”アクセスキーID”,                    ------(3)  (*)
  ”secKey”: ”シークレットアクセスキー”,          ------(4)  (*)
  ”service”: ”接続するサービス名”,               ------(5)  (*)
  ”region”: ”接続するサービスのリージョン”,      ------(6)  (*)
  ”endpoint”: ”エンドポイント”,                  ------(7)  (*)
  ”sigv4Header”: {署名用のリクエストヘッダ},     ------(8)
  ”memo”: ”メモ”,                                ------(9)
  ”任意のキー”: ”任意の値”                       ------(10)
}
```

(*)：必ず指定してください。

(1) name 認証情報名

登録する認証情報名を指定します。

64バイト以内で指定します。

半角英数字、“_”(半角アンダーバー)、“-”(半角ハイフン)を利用できます。ただし、先頭に“-”(半角ハイフン)は利用できません。

(2) authType 認証種別

awssigv4を指定します。

(3) accKey アクセスキーID

Amazon Web ServicesのAPIを利用するためのアクセスキーIDを指定します。

(4) secKey シークレットアクセスキー

Amazon Web ServicesのAPIを利用するためのシークレットアクセスキーを指定します。

(5) service 接続するサービス名

接続するサービス名を指定します。

(例1) AWS Batchの場合：batch

(例2) AWS Lambdaの場合：lambda

各サービス名の形式については、Amazon Web Servicesの公式ドキュメントを参照してください。

(6) region 接続先のリージョン

接続先のリージョンを指定します。

(例) 東京リージョンの場合：ap-northeast-1

各リージョンの形式については、Amazon Web Servicesの公式ドキュメントを参照してください。

(7) endpoint エンドポイント

接続先のエンドポイントを指定します。

(例1) AWS Batchサービス(東京リージョン)の場合：batch.ap-northeast-1.amazonaws.com

(例2) AWS Lambdaサービス(東京リージョン)の場合：lambda.ap-northeast-1.amazonaws.com

Amazon Web ServicesのAPIエンドポイントの形式の詳細については、Amazon Web Servicesの公式ドキュメントを参照してください。

(8) sigv4Header 署名用のリクエストヘッダ

REST実行ジョブのAWS_SigV4認証では署名用のリクエストヘッダとして、“host”と“x-amz-date”を自動的に設定します。これ以外に署名に必要なリクエストヘッダがあれば、ヘッダ名：値の形式で指定します。

ヘッダ名と値は、以下のようにsigv4Headerオブジェクト内で複数指定可能です。

```
sigv4Header: {
ヘッダ名1:値1,
ヘッダ名2:値2,
…
}
```

署名に必要なリクエストヘッダの形式の詳細は、Amazon Web Serviceの公式ドキュメントを参照してください。

(9) memo メモ

登録する認証情報のメモ情報を指定します。

1024バイト以内で指定してください。

(10) 任意のキー 任意の値

REST実行ジョブに指定するクエリパラメタやHTTPリクエストヘッダの値を、接続情報ファイルに直接記載したくない場合、本オペランドで任意のキーおよび値を指定します。接続情報ファイルでの本オペランドの指定方法については、“20.9.1 接続情報ファイル”を参照してください。

本オペランドはキーの名前が一意であれば複数設定できます。

任意のキー、任意の値は1024バイト以内で指定してください。

なお、任意のキーの先頭文字に“-”(半角ハイフン)は利用できません。

Microsoft Entra ID認証(OAuth2.0 Client Credentials認証)の場合の制御文

AzureのAPIにリクエストするときに本認証種別を選択します。

Microsoft Entra ID認証では、OAuth2.0のClient Credentialsフローでの認証を実施します。

本認証種別で設定した認証情報名は、REST実行ジョブの接続情報ファイルの認証情報名パラメタで指定します。

Microsoft Entra ID(v1.0)エンドポイントを使用した認証の場合

```
{
  ”name”: ”認証情報名”,                          ------(1)  (*)
  ”authType”: ”認証種別”,                        ------(2)  (*)
  ”clientId”: ”クライアントID”,                  ------(3)  (*)
  ”clientSecret”: ”クライアントシークレット”,    ------(4)  (*)
  ”endpoint”: ”エンドポイント”,                  ------(5)  (*)
  ”resource”: ”リソース”,                        ------(6)  (*)
  ”memo”: ”メモ”,                                ------(8)
  ”任意のキー”: ”任意の値”                       ------(9)
}
```

(*)：必ず指定してください。

Microsoft ID プラットフォーム(v2.0)エンドポイントを使用した認証の場合

```
{
  ”name”: ”認証情報名”,                          ------(1)  (*)
  ”authType”: ”認証種別”,                        ------(2)  (*)
  ”clientId”: ”クライアントID”,                  ------(3)  (*)
  ”clientSecret”: ”クライアントシークレット”,    ------(4)  (*)
  ”endpoint”: ”エンドポイント”,                  ------(5)  (*)
  ”scope”: ”スコープ”,                           ------(7)  (*)
  ”memo”: ”メモ”,                                ------(8)
  ”任意のキー”: ”任意の値”                       ------(9)
}
```

(*)：必ず指定してください。

(1) name 認証情報名

登録する認証情報名を指定します。

64バイト以内で指定します。

半角英数字、“_”(半角アンダーバー)、“-”(半角ハイフン)を利用できます。ただし、先頭に“-”(半角ハイフン)は利用できません。

(2) authType 認証種別

azureadを指定します。

(3) clinetId クライアントID

OAuth2.0認証のクライアントIDを指定します。

クライアントIDの形式の詳細については、Azureの公式ドキュメントを参照してください。

(4) clinetSecret クライアントシークレット

OAuth2.0認証のクライアントシークレットを指定します。

クライアントシークレットの形式の詳細については、Azureの公式ドキュメントを参照してください。

(5) endpoint エンドポイント

OAuth2.0認証のエンドポイントを指定します。

アクセストークン取得先のURLを以下のように指定します。

例) https://login.microsoftonline.com/<テナントID>/oauth2/token

エンドポイントの形式の詳細については、Azureの公式ドキュメントを参照してください。

(6) resource リソース

Microsoft Entra ID(v1.0)エンドポイントを使用した認証を利用する場合に指定します。

OAuth2.0認証のリソースを指定します。

scopeと同時に指定することはできません。

リソースの形式の詳細については、Azureの公式ドキュメントを参照してください。

(7) scope スコープ

Microsoft ID プラットフォーム(v2.0)エンドポイントを使用した認証を利用する場合に指定します。

OAuth2.0認証のスコープを指定します。

resourceと同時に指定することはできません。

スコープの形式の詳細については、Azureの公式ドキュメントを参照してください。

(8) memo メモ

登録する認証情報のメモ情報を指定します。

1024バイト以内で指定してください。

(9) 任意のキー 任意の値

REST実行ジョブに指定するクエリパラメタやHTTPリクエストヘッダの値を接続情報ファイルに直接記載したくない場合、本オペランドで任意のキーおよび値を指定します。接続情報ファイルでの本オペランドの指定方法については、“20.9.1 接続情報ファイル”を参照してください。

本オペランドはキーの名前が一意であれば複数設定できます。

任意のキー、任意の値は1024バイト以内で指定してください。

なお、任意のキーの先頭文字に“-”(半角ハイフン)は利用できません。

SAPクラウドサービス認証(OAuth2.0 Client Credentials認証)の場合の制御文

SAP Job Scheduling サービスのAPIに、OAuth2.0認証でリクエストするときに本認証種別を選択します。

SAPクラウドサービス認証では、OAuth2.0 の Client Credentialsフローでの認証を実施します。

本認証種別で設定した認証情報名は、REST実行ジョブの接続情報ファイルの認証情報名パラメタで指定します。

```
{
  ”name”: ”認証情報名”,                          ------(1)  (*)
  ”authType”: ”認証種別”,                        ------(2)  (*)
  ”clientId”: ”クライアントID”,                  ------(3)  (*)
  ”clientSecret”: ”クライアントシークレット”,    ------(4)  (*)
  ”endpoint”: ”エンドポイント”,                  ------(5)  (*)
  ”memo”: ”メモ”                                 ------(6)
}
```

(*)：必ず指定してください。

(1) name 認証情報名

登録する認証情報名を指定します。

64バイト以内で指定します。

半角英数字、“_”(半角アンダーバー)、“-”(半角ハイフン)を利用できます。ただし、先頭に“-”(半角ハイフン)は利用できません。

(2) authType 認証種別

scsを指定します。

(3) clinetId クライアントID

OAuth2.0認証のクライアントIDを指定します。

クライアントIDの形式の詳細については、SAPクラウドサービスの公式ドキュメントを参照してください。

(4) clinetSecret クライアントシークレット

OAuth2.0認証のクライアントシークレットを指定します。

クライアントシークレットの形式の詳細については、SAPクラウドサービスの公式ドキュメントを参照してください。

(5) endpoint エンドポイント

OAuth2.0認証のエンドポイントを指定します。

アクセストークン取得先のURLを以下のように指定します。

例) https://cfazure.authentication.<リージョン>.hana.ondemand.com/oauth/token

エンドポイントの形式の詳細については、SAPクラウドサービスの公式ドキュメントを参照してください。

(6) memo メモ

登録する認証情報のメモ情報を指定します。

1024バイト以内で指定してください。

Oracle Cloud InfrastructureのAPIキーベース認証の場合の制御文

Oracle Cloud InfrastructureのAPIにリクエストするときに本認証種別を選択します。

Oracle Cloud InfrastructureのAPIキーベース認証では、秘密鍵を用いて署名バージョン1による署名を作成し、リクエストの認証情報に追加することで Oracle Cloud InfrastructureのAPIにリクエストします。

Oracle Cloud Infrastructureの署名バージョン1の詳細については、Oracle Cloud Infrastructureの公式ドキュメントを参照してください。

本認証種別で設定した認証情報名は、REST実行ジョブの接続情報ファイルの認証情報名パラメタで指定します。

```
{
  "name": "認証情報名",                          ------(1)  (*)
  "authType": "認証種別",                        ------(2)  (*)
  "ociUser": "ユーザーのOCID",                   ------(3)  (*)
  "tenancy": "テナンシのOCID",                   ------(4)  (*)
  "fingerprint": "公開鍵のフィンガープリント",   ------(5)  (*)
  "privateKey": "秘密鍵の値",                    ------(6)  (*)
  "passPhrase": "秘密鍵のパスフレーズ",          ------(7)
  "memo": "メモ",                                ------(8)
  "任意のキー": "任意の値"                       ------(9)
}
```

(*)：必ず指定してください。

(1) name 認証情報名

登録する認証情報名を指定します。

64バイト以内で指定します。

半角英数字、“_”(半角アンダーバー)、“-”(半角ハイフン)を利用できます。ただし、先頭に“-”(半角ハイフン)は利用できません。

(2) authType 認証種別

ociを指定します。

(3) ociUser ユーザーのOCID

Oracle Cloud InfrastructureのAPIを利用するためのユーザーのOCIDを指定します。

OCIDの形式の詳細については、Oracle Cloud Infrastructureの公式ドキュメントを参照してください。

(4) tenancy テナンシのOCID

接続先のテナンシのOCIDを指定します。

OCIDの形式の詳細については、Oracle Cloud Infrastructureの公式ドキュメントを参照してください。

(5) fingerprint 公開鍵のフィンガープリント

Oracle Cloud InfrastructureのAPIを利用するための公開鍵のフィンガープリントを指定します。

フィンガープリントの形式の詳細については、Oracle Cloud Infrastructureの公式ドキュメントを参照してください。

(6) privateKey 秘密鍵の値

Oracle Cloud InfrastructureのAPIを利用するための秘密鍵の値を指定します。

秘密鍵の詳細については、Oracle Cloud Infrastructureの公式ドキュメントを参照してください。

指定例)

```
-----BEGIN PRIVATE KEY----¥nBase64エンコードされたデータ¥n----END PRIVATE KEY----¥n
```

- PEM形式の文字列を、JSONのパラメタ値として1行で指定します。
- 改行文字は、「¥n」で指定してください。なお、末尾の「¥n」は省略可能です。
- PEM形式のファイルから1行の文字列への変換処理は、以下のコマンドを参考にしてください。
  なお、PEM形式のファイルで、フッターの下に文字列がある場合は、あらかじめ削除してください。

  Linuxの場合:

  ```
  # awk '{gsub("¥r",""); printf "%s¥¥n", $0}' <秘密鍵のファイルパス>
  ```

  Windowsの場合(Powershell):

  ```
  # (Get-Content <秘密鍵のファイルパス> -Raw) -replace "¥r?¥n", "¥n"
  ```

(7) passPhrase 秘密鍵のパスフレーズ

Oracle Cloud InfrastructureのAPIを利用するための秘密鍵に、パスフレーズを設定した場合に指定します。

(8) memo メモ

登録する認証情報のメモ情報を指定します。

1024バイト以内で指定してください。

(9) 任意のキー 任意の値

REST実行ジョブに指定するクエリパラメタやHTTPリクエストヘッダの値を接続情報ファイルに直接記載したくない場合、本オペランドで任意のキーおよび値を指定します。接続情報ファイルでの本オペランドの指定方法については、“20.9.1 接続情報ファイル”を参照してください。

本オペランドはキーの名前が一意であれば複数設定できます。

任意のキー、任意の値は1024バイト以内で指定してください。なお、任意のキーの先頭文字に“-”(半角ハイフン)は利用できません。

プロキシサーバ認証を設定する場合の制御文

プロキシサーバ経由でHTTPリクエストする場合、本認証種別でプロキシサーバ認証用の情報を設定します。

本認証種別で設定した認証情報名は、REST実行ジョブの接続情報ファイルのプロキシサーバの認証情報名パラメタで指定します。

```
{
  ”name”: ”認証情報名”,                          ------(1)  (*)
  ”authType”: ”認証種別”,                        ------(2)  (*)
  ”proxyAddress”: ”プロキシサーバのアドレス”,      ------(3)  (*)
  ”proxyPort”: ”プロキシサーバのポート番号”,       ------(4)  (*)
  ”proxyUser”: ”プロキシサーバの認証ユーザ名”,     ------(5)
  ”proxyPasswd”: ”プロキシサーバの認証パスワード”, ------(6)
  ”memo”: ”メモ”                                 ------(7)
}
```

(*)：必ず指定してください。

(1) name 認証情報名

登録する認証情報名を指定します。

64バイト以内で指定します。

半角英数字、“_”(半角アンダーバー)、“-”(半角ハイフン)を利用できます。ただし、先頭に“-”(半角ハイフン)は利用できません。

(2) authType 認証種別

proxyを指定します。

(3) proxyAddress プロキシサーバのアドレス

プロキシサーバのアドレスを指定します。

1024バイト以内で指定してください。

(4) proxyPort プロキシサーバのポート番号

プロキシサーバのポート番号を指定します。

0～65535の間で指定してください。

(5) proxyUser プロキシサーバの認証ユーザ名

プロキシサーバの認証ユーザ名を指定します。

1024バイト以内で指定してください。

プロキシサーバの認証パスワードが指定されている場合は、必ず指定してください。

(6) proxyPasswd プロキシサーバの認証パスワード

プロキシサーバの認証パスワードを指定します。

1024バイト以内で指定してください。

プロキシサーバの認証ユーザ名が指定されている場合は、必ず指定してください。

(7) memo メモ

登録する認証情報のメモ情報を指定します。

1024バイト以内で指定してください。

任意のキー、値の認証を設定する場合の制御文

他の認証種別による認証を実施しない、かつ、REST実行ジョブに指定するクエリパラメタやHTTPリクエストヘッダ、HTTPリクエストメッセージボディの値を接続情報ファイルに直接記載したくない場合は、本認証種別で任意のキーおよび値を設定できます。

本認証種別で設定した認証情報名は、REST実行ジョブの接続情報ファイルの認証情報名パラメタで指定します。

```
{
  "name": "認証情報名",                          ------(1)  (*)
  "authType": "認証種別",                        ------(2)  (*)
  "memo": "メモ",                                ------(3)
  "任意のキー": "任意の値"                       ------(4)
}
```

(*)：必ず指定してください。

(1) name 認証情報名

登録する認証情報名を指定します。

64バイト以内で指定します。

半角英数字、“_”(半角アンダーバー)、“-”(半角ハイフン)を利用できます。ただし、先頭に“-”(半角ハイフン)は利用できません。

(2) authType 認証種別

othersを指定します。

(3) memo メモ

登録する認証情報のメモ情報を指定します。

1024バイト以内で指定してください。

(4) 任意のキー 任意の値

任意のキーおよび値を指定します。接続情報ファイルでの本オペランドの指定方法については、“20.9.1 接続情報ファイル”を参照してください。本オペランドはキーの名前が一意であれば複数設定できます。

任意のキー、任意の値は1024バイト以内で指定してください。なお、任意のキーの先頭文字に“-”(半角ハイフン)は利用できません。

4.10 jobschrefstatusコマンド【Windows版】【Linux版】

記述形式

jobschrefstatus プロジェクト名 状態情報ファイル [-title] [-sys サブシステム番号](注)

注)

“[-sys サブシステム番号]”はSystemwalker Operation Manager EEのオプションです。

機能説明

災害対策運用を行う場合に使用します。

ジョブスケジューラの起動パラメタファイル(jobschev.txt)で、DR_OPTをONに設定した場合に使用できます。起動パラメタファイルの設定については、“Systemwalker Operation Manager 運用ガイド”の“災対運用の準備作業(現用環境・災対環境)”を参照してください。

状態情報ファイルに格納されているジョブ/ジョブネットの状態情報を、プロジェクト名で指定したプロジェクトのジョブネットの定義情報ファイルに反映します。

反映結果は、CSV形式で標準出力に出力されます。

指定した状態情報ファイルの行数を超える出力があるので、必要に応じてファイルなどへリダイレクトするようにしてください。

オプション

プロジェクト名

状態を反映するプロジェクト名を指定します。

状態情報ファイル

状態情報が格納されている状態情報ファイル名をフルパスで指定します。

-title

指定した場合、1行目にタイトル行が出力されます。

EE GEE

-sys サブシステム番号

複数サブシステム運用をしているシステムで、操作対象のサブシステムを指定するためのオプションです。操作対象とするサブシステム番号を、0から9の範囲で指定します。このオプションを省略した場合、サブシステム番号0が操作対象となります。

復帰値

復帰値	意味
0	正常終了
1	状態情報の指定に誤りがある(誤りのある行についてメッセージを表示し、処理は継続する)
2	指定のプロジェクトが存在しない(異常終了)
3	状態情報ファイル名の指定に誤りがある(異常終了)
4	その他(ファイルアクセス失敗、メモリ獲得失敗など)のエラー(異常終了)

コマンド格納場所

Windows	Systemwalker Operation Managerインストール先ディレクトリ¥MpWalker.JM¥bin
Linux	/usr/bin

実行に必要な権限

システム管理者(Administratorsグループに属するユーザ/スーパーユーザ)の権限が必要です。

出力項目の説明

出力項目について説明します。

Log Type

出力されるログのタイプは以下に示した4種類から選択表示されます。

ERROR:

状態情報ファイルの構文誤り、反映不可な対象が指定された場合などに出力されます。

原因コード(Code)が出力されます。

WARNING:

状態情報ファイルの状態に、実行中(execute/rerun)や、完了状態(normal/notice/abend/cancel)が指定された場合で、対象の先行ジョブに完了状態ではないジョブが存在する場合に、対象と変更した状態(Status)が出力されます。

原因コード(Code)が出力されます。

UPDATE:

状態反映後に、状態の整合性をあわせるための補正を行った場合に出力されます。

補正対象と補正後の状態(Status)が出力されます。

原因コード(Code)が出力されます。

以下の場合に補正されます。

- 状態反映後、子ジョブネットと親ジョブネット側のジョブ(子ジョブネット)の状態に不整合が発生した場合
  以下のルールで状態を補正します。(原因コード：0301)
  - 子ジョブネットのみ状態が変更されている場合
    子ジョブネットの状態を基準とし、ジョブ(子ジョブネット)の状態を更新します。
  - ジョブ(子ジョブネット)のみ状態が変更されている場合
    ジョブ(子ジョブネット)の状態を基準とし、子ジョブネットの状態を更新します。
  - 両方の状態が変更されている場合
    先に変更された状態を基準として、状態を合わせます。
- ジョブネットが完了状態で、配下のジョブや子ジョブネットが実行中の場合
  異常終了にします。(原因コード:0301)
- 停止・無効に変更した場合で、変更前の状態が実行中や警告の場合
  変更前の状態を異常終了にします。(原因コード:0304)

INFO:

状態反映後に問題がなかった場合に出力されます。状態は出力されません。

Job Type

ジョブネットかジョブかを示す情報が出力されます。

NET:

変更対象がジョブネットの場合に出力されます。

JOB:

変更対象がジョブの場合に出力されます。

Job Net

変更対象のジョブネット名が出力されます。

Job Number

JOBの場合に変更対象のジョブ番号が出力されます。

Status

WARNINGの場合は、問題のある状態が出力されます。

UPDATEの場合は、変更後の状態が出力されます。

出力される状態は以下に示した8種類の状態から選択表示されます。

WAIT:

実行待ち

EXECUTE:

実行中

ABEND:

異常終了

NORMAL:

正常終了

NOTICE:

疑似正常

CANCEL:

強制終了

PAUSE:

停止中

DISABLE:

無効

Code

ERROR、WARNING、UPDATEの場合に原因コードが出力されます。

原因コード	意味	システムの処理	利用者の処置
0101	グループ内ジョブネットが指定されました。	スキップします。	指定を確認してください。
0105	構文誤りがあります.	スキップします。	指定を確認してください。
0108	指定されたジョブネットまたはジョブが存在しません。	スキップします。	指定を確認してください。
0201	先行ジョブに完了状態ではないジョブが存在します。	反映します。	先行ジョブの状態を確認してください。
0301	状態の不整合を調整しました。	反映します。	なし
0304	停止または無効にする前の状態を異常終了にしました。(注)	反映します。	なし

注) 停止または無効にする前の状態は内部的に保持されていますが、その状態を異常終了として設定します。

Line

状態情報ファイルの行数が出力されます。

注意事項

- 本コマンドは、ジョブスケジューラが起動中は実行できません。
- 本コマンドの出力結果のみを出力しなおすことはできませんので、標準出力をファイルなどにリダイレクトするようにしてください。
- 反映した状態情報ファイルの内容に間違いがあった場合は、間違った箇所を変更する状態情報ファイルを作成し(差分のみを記載します)、本コマンドにより再度反映してください。

使用例

状態情報ファイル /tmp/stateinfo に格納されているジョブ/ジョブネットの状態情報を、サブシステム1にあるproject01プロジェクトのジョブネットの定義情報ファイルに反映する場合

```
jobschrefstatus project01 /tmp/stateinfo -title -sys 1
```

出力例

コマンドの出力結果の例を以下に示します。

```
Log Type,Job Type,Job Net,Job Number,Status,Code,Line
INFO,NET,jobnet02,,,,1
ERROR,NET,jobnet10,,,0101,2
WARNING,JOB,jobnet01,3,NORMAL,0201,3
WARNING,JOB,jobnet03,2,NORMAL,0201,4
UPDATE,JOB,jobnet04,3,ABEND,0301,5
INFO,JOB,jobnet02,1,,,6
INFO,JOB,jobnet02,2,,,7
```

4.10.1 状態情報ファイル

記述形式

[ジョブ]

```
JOB,ジョブネット名,ジョブ番号,状態,日時,終了コード
```

[ジョブネット]

```
NET,ジョブネット名,状態,日時,終了コード
```

パラメタの説明

種別(JOB/NET)

状態を反映する種別(ジョブ/ジョブネット)を指定します。

ジョブネット名

状態を反映するジョブネット名を指定します。

グループやグループ内ジョブネットは指定できません。

ジョブ番号

状態を反映するジョブの番号を指定します。

状態

反映する状態を指定します。以下の文字列のうちどれか1つを指定します。

状態	説明
execute	状態を実行中にします。この文字列を指定した場合、終了日時と終了コードがクリアされます。 種別が“NET”の場合、ジョブネット内のジョブの状態はすべて実行待ちになります。
rerun	状態を実行中にします。この文字列を指定した場合、終了日時と終了コードがクリアされます。 種別が“NET”の場合、再起動で、ジョブネット内のジョブの状態を前回の起動時から変更したくない場合に指定します。
normal	状態を正常終了にします。
notice	状態を擬似正常にします。
abend	状態を異常終了にします。
cancel	状態を強制終了にします。この文字列を指定した場合、終了コードに256が設定されます。
pause	状態を停止中にします。
disable	状態を無効にします。

日時

YYYY/MM/DD hh:mm:ssの形式で指定します。

状態にexecute/rerunを指定した場合には、開始日時を指定します。

状態にnormal/notice/abend/cancelを指定した場合には、終了日時を指定します。

状態にpause/disableを指定した場合には、本パラメタを指定する必要はありません。指定した場合は無視されます。

終了コード

状態にnormal/notice/abendを指定した場合に、ジョブ、ジョブネットの終了コードを0～255の範囲で指定します。(省略不可)

状態にexecute/rerun/cancel/pause/disableを指定した場合には、本パラメタを指定する必要はありません。指定した場合は無視されます。

状態情報ファイル指定例

[ジョブ]

```
JOB,jobnet01,25,normal,2025/12/12 12:12:12,0
```

[ジョブネット]

```
NET,jobnet02,abend,2025/12/10 15:20:00,255
```

注意事項

- 状態情報ファイルは、1行(改行まで)を1定義と判断します。
- 状態情報ファイル中に空行(改行のみの行)があった場合、その行は無視します。

- 状態情報ファイルは、時系列で記述してください。(時系列になっていない場合、状態が正しく反映されない場合があります。)
- 実行中のジョブおよびジョブネットを停止/無効に変更した場合、停止解除/無効解除の操作をすると、状態は実行中ではなく異常終了になります。
- 実行中のジョブネットを完了状態(注)に変更する場合、配下のジョブおよび子ジョブネットで実行中のものがあれば、異常終了に変更されます。
- 状態反映後に、状態の整合性をあわせるための補正を行う場合があります。
- ジョブネットの状態を変更した場合(停止に変更する場合は除く)、メッセージ事象の発生状況をクリアします。

注) 完了状態とは、正常終了、異常終了、疑似正常、強制終了、確認済のいずれかの状態です。

4.11 jobschctlprojectコマンド【Windows版】【Linux版】

記述形式

```
jobschctlproject プロジェクト名 control [-sys サブシステム番号](注)
```

注)

"[-sys サブシステム番号]"はSystemwalker Operation Manager EEのオプションです。

機能説明

災害対策運用を行う場合に使用します。

プロジェクトの無効化または有効化を行います。また、現在指定されているプロジェクト状態を表示します。

プロジェクトを無効化すると、配下のジョブネットは起動条件が成立しても起動しません。(ジョブネットやジョブ自体の状態は変更されません。)

マスタプロジェクト、および運用変更用プロジェクトは指定できません。

本コマンドで指定したプロジェクトの状態の適用は、以下のタイミングで行われます。コマンドを実行しただけでは状態は変化しません。

- ジョブスケジューラサービス/デーモンの起動時
- 日変わり時刻の到来時

オプション

プロジェクト名

操作対象のプロジェクト名を指定します。

control

以下に示す3つのうちどれか1つを指定します。

disable

指定されたプロジェクトを無効化します。

enable

指定されたプロジェクトを有効化します。

check

指定されたプロジェクトの状態を表示します。

SETに指定値を、STATUSに現在の適用状態を表示します。

EE GEE **-sys サブシステム番号**

複数サブシステム運用をしているシステムで、操作対象のサブシステムを指定するためのオプションです。操作対象とするサブシステム番号を、0から9の範囲で指定します。このオプションを省略した場合、サブシステム番号0が操作対象となります。

復帰値

復帰値	意味
0	正常終了
0以外	異常終了。エラーの内容はエラーメッセージを確認してください。

checkを指定した場合の出力形式

指定値を指定してコマンドを実行した直後に、checkを指定してコマンドを実行した場合の出力内容は以下のとおりです。

指定値	プロジェクトの状態	出力内容
disable	無効	SET:disable,STATUS:disable
disable	有効	SET:disable,STATUS:enable
disable	サーバ停止中	SET:disable,STATUS:-
enable	無効	SET:enable,STATUS:disable
enable	有効	SET:enable,STATUS:enable
enable	サーバ停止中	SET:enable,STATUS:-

コマンド格納場所

Windows	Systemwalker Operation Managerインストール先ディレクトリ¥MpWalker.JM¥bin
Linux	/usr/bin

実行に必要な権限

システム管理者(Administratorsグループに属するユーザ/スーパーユーザ)の権限が必要です。

注意事項

本コマンドは、ジョブスケジューラサービス/デーモンが停止している場合でも実行可能です。

4.12 jobschsetnetコマンド

本コマンドは、ジョブネットの設定(登録/変更/削除/複写)および、ジョブネット情報の起動日、メッセージ事象および一時変更情報を設定(登録/削除)する場合に使用します。

<ジョブネットの設定>

ジョブネットを設定する場合の本コマンドについて、記述形式、機能説明およびオプションの説明を以下に示します。

記述形式

```
[登録の場合]
  jobschsetnet -nent [プロジェクト名] 定義ファイル [-sys サブシステム番号] (注)
[変更の場合]
```

```
 jobschsetnet -nche [プロジェクト名] 定義ファイル [-sys サブシステム番号] (注)
 jobschsetnet -ncheall [プロジェクト名] 定義ファイル [-sys サブシステム番号] (注)
[削除の場合]
 jobschsetnet -ndel [プロジェクト名/]ジョブネット名 [-mod mode][-sys サブシステム番号]
(注)
[複写の場合]
 jobschsetnet -ncpy [複写元プロジェクト名/]複写元ジョブネット名
                    [複写先プロジェクト名/]複写先ジョブネット名
                    [-sys サブシステム番号]     (注)
```

注)

“ [-sys サブシステム番号]”はSystemwalker Operation Manager EEのオプションです。

機能説明

ジョブネットを登録/変更/削除/複写します。

ジョブネットの登録/変更時には、jobschprint -rコマンドにより展開された起動日情報(特異日の情報を含む)を設定できます。

操作対象とするプロジェクト名が指定できます。

注意

プロジェクトに登録できるジョブネット数について

1つのプロジェクトに登録できるジョブネット数は、Editionの違い(SE/EE)によって、以下のように異なります。

- Systemwalker Operation Manager SEは、最大255個まで登録できます。
- Systemwalker Operation Manager EEは、登録できるジョブネット数に制限はありません。

オプション

-nent

定義ファイルの情報をもとにジョブネットを登録します。なお、同一ジョブネット名がすでに存在する場合は登録できません。

注意

定義ファイルにジョブネット制御文を複数記述し、階層化したジョブネットが含まれる場合、子ジョブネットとリンクジョブネットは親ジョブネットより先に記述する必要があります。

定義ファイルにジョブネット制御文を複数記述し、基準ジョブネットが含まれる場合、その基準ジョブネットの起動日を利用しているジョブネットと別の定義ファイルにする必要があります。

プロジェクト名

ジョブネットの情報を格納する格納先のプロジェクト名を指定します。登録時の場合、jobschsetnetコマンドの実行ユーザが更新アクセスできるプロジェクトが1つしかない場合は、省略できます。変更/削除/複写の場合、jobschsetnetコマンド実行のユーザが更新アクセスできるプロジェクトのうちジョブネットが登録されているプロジェクトが1つしかない場合は、省略できます。

定義ファイル

ジョブネット制御文およびジョブ制御文を格納したファイル名を指定します。

複数のジョブネットを登録するときは、ジョブネット制御文を複数記述します。

“ジョブネット制御文”については、“4.12.1 ジョブネット制御文”を参照してください。

“ジョブ制御文”については、“4.12.2 ジョブ制御文”を参照してください。

-nche
-ncheall

定義ファイルの情報をもとに既存のジョブネットを変更します。

-ncheオプションは、ジョブネット登録時に使用した定義ファイルに、変更内容を記述して利用すると便利です。定義ファイルに記述されていないオペランドについては、変更対象がジョブネットの場合は変更されませんが、ジョブの場合は定義が削除されるので、注意が必要です。ジョブを変更する際は、定義ファイルにすべてのオペランドを記述してください。

-ncheallオプションは、ジョブネット/ジョブに関わらず、定義ファイルに記述されていないオペランドについては、パラメタに省略値が設定されます。定義ファイルにオペランドだけを記載した場合も、パラメタに省略値が設定されます。起動日/メッセージ事象/一時変更情報など省略値がないものは削除されます。

注意

ジョブネット制御文では、オペランドとパラメタは対で指定する必要があります。-ncheオプションは、定義ファイルにオペランドだけを記載してもパラメタは削除されません。登録してある起動日/メッセージ事象/一時変更情報を削除する場合は、-ncheallオプションを指定するか、jobschsetnetコマンドの-sdelオプションを実行してください。

注意

ジョブネット変更時の発生済メッセージ事象のクリアについて

ジョブネットを変更した場合、発生済みのメッセージ事象はすべてクリアされます。そのため、メッセージ事象発生後にジョブネットを変更した場合は、再度メッセージ事象を発生させる必要があります。

注意

定義ファイルにジョブネット制御文を複数記述し、階層化したジョブネットが含まれる場合、子ジョブネットとリンクジョブネットは親ジョブネットより先に記述する必要があります。

親ジョブネットを先に記述した場合、以下のエラーメッセージを出力して、異常終了する場合があります。

[UNIX]

MpJobsch: ERROR: 0225: ジョブネットは保存中または制御中です

[Windows]

MpJobsch:エラー :6225:ジョブネットは保存中または制御中です

定義ファイルにジョブネット制御文を複数記述し、基準ジョブネットが含まれる場合、その基準ジョブネットの起動日を利用しているジョブネットと別の定義ファイルにする必要があります。

同じ定義ファイルに記述した場合、以下のエラーメッセージを出力して、異常終了する場合があります。

[UNIX]

MpJobsch: ERROR: 0225: ジョブネットは保存中または制御中です

[Windows]

MpJobsch:エラー :6225:ジョブネットは保存中または制御中です

-ndel

ジョブネットの削除を行います。ジョブネットの削除を行った場合、ジョブネットおよびジョブネットを構成するジョブの登録情報がすべて削除されます。

なお、階層化されたジョブネットで、-modオプションを省略して削除した場合は、第2階層のジョブネットが第1階層のジョブネットとなります。

ジョブネット名

削除するジョブネット名を指定します。第2階層以下のジョブネットは指定できません。

-mod *mode*

削除時のモードを0または1で指定します。モードにより削除対象のジョブネットが変わります。

ただし、削除対象が複写先ジョブネットの場合、本指定にかかわらず、配下の子ジョブネットもあわせて削除します。省略値は“0”です。

0

指定したジョブネットのみ削除します。

削除対象が階層化されたジョブネットの場合、第2階層のジョブネットが第1階層のジョブネットとなります。

1

配下の子ジョブネットもあわせて削除します。

-ncpy

ジョブネットを複写します。複写先のジョブネットは“無効状態”となります。なお、複写先のプロジェクトに同一ジョブネット名がすでに存在する場合は複写できません。

また、複写元と複写先が異なるプロジェクトに“基準ジョブネット名”を指定しているジョブネットを複写した場合、基準ジョブネット名が消去されます。ただし、基準ジョブネット名以外の起動日情報は複写元のジョブネットと同一の値となります。

複写元プロジェクト名

複写元となるジョブネットの情報が格納されているプロジェクト名を指定します。jobschsetnetコマンドの実行ユーザが参照できるプロジェクトのうち、ジョブネットが登録されているプロジェクトが1つしかない場合は、省略できます。

複写元ジョブネット名

複写元のジョブネット名を指定します。

複写先プロジェクト名

複写先となるジョブネットの情報が格納されているプロジェクト名を指定します。省略した場合は、複写元のプロジェクト名と同一のプロジェクト名が指定されたものとみなします。jobschsetnetコマンドの実行ユーザが更新アクセス可能なプロジェクト名を指定します。

複写先ジョブネット名

複写先のジョブネット名を指定します。



参考

V13.7.0以前のジョブネットの複写用オプション“-ncop”について

ジョブネットを複写するオプションとして、V13.7.0以前では“-ncop”オプションが提供されていました。本バージョンでも“-ncop”オプションは、“-ncpy”オプションと同じ指定方法で使用できますが、“-ncop”オプション指定で複写した場合、複写先のジョブネットの状態は“実行待ち”となります。

EE GEE

-sys サブシステム番号

複数サブシステム運用をしているシステムで、操作対象のサブシステムを指定するためのオプションです。操作対象とするサブシステム番号を、0から9の範囲で指定します。このオプションを省略した場合、サブシステム番号0が操作対象となります。

＜起動日/メッセージ事象/一時変更情報の設定＞

起動日/メッセージ事象/一時変更情報を設定する場合の本コマンドについて、記述形式、機能説明、オプションの説明、および注意事項を以下に示します。

記述形式

```
[登録の場合]
 jobschsetnet -sent [プロジェクト名] 定義ファイル [-sys サブシステム番号]
(注)
[削除の場合]
 jobschsetnet -sdel [プロジェクト名/]ジョブネット名 [SCH|MSG|TMP]
            [-sys サブシステム番号]  (注)
```

注)

“ [-sys サブシステム番号]”はSystemwalker Operation Manager EEのオプションです。

機能説明

ジョブネット情報の起動日、メッセージ事象および一時変更情報を登録/削除します。

オプション

-sent

定義ファイルの情報をもとに起動日、メッセージ事象および一時変更情報を登録します。

プロジェクト名

ジョブネットの情報が格納されているプロジェクト名を指定します。jobschsetnetコマンドの実行ユーザが更新アクセスできるプロジェクトのうちジョブネットが登録されているプロジェクトが1つしかない場合は、省略できます。

定義ファイル

起動日制御文、メッセージ事象制御文、および一時変更制御文を格納したファイル名を指定します。

“起動日制御文”については、“4.12.3 起動日制御文”を参照してください。

“メッセージ事象制御文”については、“4.12.4 メッセージ事象制御文”を参照してください。

“一時変更制御文”については、“4.12.5 一時変更制御文”を参照してください。

-sdel

起動日/メッセージ事象/一時変更情報を削除します。

ジョブネット名

起動日/メッセージ事象/一時変更情報を削除するジョブネット名を指定します。

SCH

起動日のみを削除します。

MSG

メッセージ事象のみを削除します。

TMP

一時変更情報のみを削除します。

EE GEE **-sys サブシステム番号**

複数サブシステム運用をしているシステムで、操作対象のサブシステムを指定するためのオプションです。操作対象とするサブシステム番号を、0から9の範囲で指定します。このオプションを省略した場合、サブシステム番号0が操作対象となります。

復帰値

復帰値	意味
0	正常終了
0以外	異常終了。エラーの内容はエラーメッセージを確認してください。

コマンド格納場所

Windows	Systemwalker Operation Managerインストール先ディレクトリ¥MpWalker.JM¥bin
Solaris HP-UX AIX Linux	/usr/bin

注意事項

- ジョブネットの削除で削除に失敗した場合は、配下のジョブネットも含め、すべて削除されません。
- 起動日/メッセージ事象/一時変更情報を登録する場合、すでに起動日/メッセージ事象/一時変更情報の情報が登録されている場合は、定義ファイルの情報に置き換えられます。
- 起動日/メッセージ事象/一時変更情報のすべてを削除する場合、SCHパラメタ、MSGパラメタおよびTMPパラメタは指定しないでください。
- 本コマンドは、ジョブネットの変更をサーバに依頼するだけであり、本コマンドの実行完了が、ジョブネットの変更処理の完了を保証するものではありません(本コマンドの実行完了後、ジョブネットの変更処理が完了するまでに、しばらく時間がかかる場合があります)。
- 本コマンドで登録するジョブネットが利用する起動日雛形および休日カレンダを変更／削除する場合には、本コマンドの実行完了後に実施してください。
- jobschprint -r コマンド、jobschprint -R コマンドで出力されたファイルを本コマンドを使用してジョブネットを登録する場合は、ジョブのアイコン位置情報が初期化されるため、適宜修正してください。
- jobschprintコマンドによりジョブネットの終了状態を確認したにも関わらず、本コマンドで以下のメッセージが出力された場合は、ジョブネットの状態が完全に終了状態に遷移するまでのわずかなタイミングにより発生したものなので、しばらく待ってから再度実行してください。

  「ジョブネットは既に実行中または警告状態です」

  「ジョブネットは既に実行中状態または警告状態です」

4.12.1 ジョブネット制御文

ジョブネットの登録情報を設定する制御文です。

記述形式

```
NET                                  ---------- (1)
jobnetname ジョブネット名            ---------- (2)  (*)   (注1)
```

jobnetcomment ジョブネット名称 ---------- (3) (*) (注2)(注3)
noexecution 起動条件なし ---------- (4) (**) (注3)(注4)
starttime 単一起動 ---------- (5) (**) (注3)
multitime 複数起動 ---------- (6) (**) (注3)
intervalstart 一定間隔 ---------- (7) (**) (注3)
iplstart サーバ起動時 ---------- (8) (**) (注3)
msgonly メッセージ起動 ---------- (9) (**) (注3)
msgonlymode メッセージ起動時の処置 ---------- (10)
msgstarttime 開始予定時刻 ---------- (11) (注5)
stoptime 停止時刻 ---------- (12)
alarmtime 終了予定時刻 ---------- (13) (注6)
estimatetime 実行予測時間 ---------- (14) (注7)
powermode 電源未投入時の処置 ---------- (15)
iplmode サーバ起動時の処置 ---------- (16)
execattr 実行属性 ---------- (17)
carryovermode 運用指示 ---------- (18)
copystart 複写起動する ---------- (19) (注18)
jobnetsavenumber 複写先ジョブネットの保存数 -------- (20) (注18)(注19)
defaulthost デフォルトホスト名 ---------- (21)
standnet 基準ジョブネット宣言 ---------- (22)
execpattern 起動日雛形名 ---------- (23) (***)
execpatmode 起動日雛形組み合わせ条件 ---------- (24) (注8)
basenet 基準ジョブネット名 ---------- (25) (***)
execday 起動日設定 ---------- (26)
cancelday 起動日設定解除 ---------- (27)
jobschprintr 起動日ビットパターン ---------- (28)
annuallydate 毎年の日付指定 ---------- (29) (***)
annuallyweek 毎年の曜日指定 ---------- (30) (***)
monthlydate 毎月の日付指定 ---------- (31) (***)
monthlyweek 毎月の曜日指定 ---------- (32) (***)
bnsmode 営業日の選択基準 ---------- (33) (***) (注9)
bnsbase 営業日の月初め日 ---------- (34) (***) (注10)
bnsday 営業日 ---------- (35) (***)
bnslimit 月度内の営業日のみを起動日とする ---------- (36) (***)
offsetmode オフセット ---------- (37)
offsetday オフセット日数 ---------- (38)
offsetholiday 基準日が休日の場合は稼働日とする ----- (39)
holidayshift 休日シフト ---------- (40) (*) (注3)
shiftlimit シフト上限日数 ---------- (41) (注12)
holidaycalendar カレンダ名 ---------- (42)
holidaymode カレンダ組み合わせ条件 ---------- (43) (注11)
enablestart 有効期限開始日 ---------- (44)
enableend 有効期限終了日 ---------- (44)
disablestart 無効期限開始日 ---------- (45)
disableend 無効期限終了日 ---------- (45)
tstarttime 単一起動(一時変更) ---------- (46) (****) (注13)
tmultitime 複数起動(一時変更) ---------- (47) (****) (注13)
tintervalstart 一定間隔(一時変更) ---------- (48) (****) (注13)
tiplstart サーバ起動時(一時変更) ---------- (49) (****) (注13)
tmsgonly メッセージ起動(一時変更) ---------- (50) (****) (注13)
tmsgonlymode メッセージ起動時の処置(一時変更) ------ (51)
tmsgstarttime 開始予定時刻(一時変更) ---------- (52) (注14)
tstoptime 停止時刻(一時変更) ---------- (53)
talarmtime 終了予定時刻(一時変更) ---------- (54) (注15)
tpowermode 電源未投入時の処置(一時変更) ---------- (55)
tiplmode サーバ起動時の処置(一時変更) ---------- (56)
temporarystart 一時変更開始日 ---------- (57) (注13)
temporaryend 一時変更終了日 ---------- (57)
messagemode メッセージ条件 ---------- (58) (注16)
carryover 持ち越し処理 ---------- (59) (注17)
messagename メッセージ事象名 ---------- (60)
job ジョブネット内ジョブ数 ---------- (61) (*) (注3)

```
operate 操作                              ---------- (62)
;
```

注1)

jobnetnameオペランドは、NETキーワードのすぐ後に指定してください。

注2)

jobnetcommentオペランドは、jobnetnameオペランドのすぐ後に指定してください。

注3)

jobschsetnet -nche指定の場合は、省略できます。

注4)

noexecutionオペランドを指定する場合は、tstarttime、tmultitime、tintervalstart、tiplstart、tmsgonly、tstoptime、talarmtime、tpowermode、tiplmode、temporarystart、temporaryend、messagemode、messagename、msgonlymode、msgstarttime、tmsgonlymode、tmsgstarttime、carryoverの各オペランドは指定できません。

注5)

msgstarttimeオペランドは、msgonlymodeオペランドに“ONCE”を指定したときに指定できます。msgonlymodeオペランドのすぐ後ろに指定してください。

注6)

alarmtimeオペランドを指定する場合は、starttimeオペランドも指定してください。stoptimeオペランド、estimatetimeオペランドとは同時に指定できません。

注7)

estimatetimeオペランドは、alarmtimeオペランドとは同時に指定できません。

注8)

execpatmodeオペランドは、execpatternオペランドのすぐ後ろに指定してください。

注9)

bnsmodeオペランドはoffsetmode,offsetday,offsetholidayの各オペランドとは同時には指定できません。

注10)

bnsmodeオペランドを指定する場合には必ず指定してください。

注11)

holidaymodeオペランドは、holidaycalendarオペランドのすぐ後ろに指定してください。

注12)

shiftlimitオペランドは、holidayshiftオペランドのすぐ後ろに指定してください。

注13)

一時変更のオペランドを指定する場合は、必ず指定してください。jobschsetnet -nche指定の場合は、省略できます。

注14)

tmsgstarttimeオペランドは、tmsgonlymodeオペランドに“ONCE”を指定したときに指定できます。tmsgonlymodeのすぐ後ろに指定してください。

注15)

talarmtimeオペランドを指定する場合は、tstarttimeオペランドも指定してください。tstoptimeオペランド、estimatetimeオペランドとは同時に指定できません。

注16)

messagemodeオペランドを指定する場合は、messagenameオペランドも指定してください。

注17)

carryoverオペランドは、messagemodeオペランドのすぐ後ろに指定してください。

注18)

実行属性が“ジョブ実行制御”(JES)の場合に指定できます。

注19)

jobnetsavenumberオペランドは、copystartオペランドのすぐ後ろに指定してください。

備考.

ジョブネットの登録情報を変更する場合は、jobnetname以外のオペランドはすべて省略することができます。省略したオペランドの情報は変更されません。

(*)：

必ず指定してください。

(**)：

どれか1つを必ず選択してください。

(***)：

以下のどれか1つを指定してください。なお、省略することもできます。

- execpattern
- basenet
- annuallydateとannuallyweekの組合せ(どちらかまたは両方)
- monthlydateとmonthlyweekの組合せ(どちらかまたは両方)
- bnsmode、bnsbase、bnsday、bnslimit(bnsmodeおよびbnsbaseはこのオペランド群を指定するときは必須)

(****)：

指定する場合は、どれか1つを選択してください。

制御文の説明

(1) NET

ジョブネット制御文の開始を表すキーワードです。

(2) jobnetname ジョブネット名

ジョブネット名を50バイト以内で指定します。他のジョブネット名と重複しないでください。/?*;,¥"<>|:'.空白、半角カナおよびユーザ定義文字を含むことはできません。

また、1文字目に # を指定することはできません。

注：

ジョブネットの実行属性が以下の場合は、1文字目に # を指定することができます。

- 旧バージョン互換（旧・標準）属性

出口プログラムを利用する場合、以下の文字は指定できません。【Windows版】

& = ^ %

(3) jobnetcomment ジョブネット名称

ジョブネットへのコメントを100バイト以内で指定します。"、空白、半角カナおよびユーザ定義文字を含むことはできません。

出口プログラムを利用する場合、以下の文字は指定できません。【Windows版】

& < > ; = ^ | % ,

(4) noexecution 起動条件なし

ジョブネットの起動条件を“なし”とする場合に、“ON”を指定します。

(5) starttime 単一起動

ジョブネットを1日に1回起動する場合に、起動時刻をHHMM(時分)の形式で4バイトで指定します。

単一起動指定例

午前9時5分に起動する場合

```
starttime 0905
```

午後9時に起動する場合

```
starttime 2100
```

(6) multitime 複数起動

ジョブネットを一日に複数回起動する場合に、起動時刻をHHMM(時分)の形式で4バイトで指定します。起動時刻は6個まで指定することができます。起動時刻を1個しか指定しなかった場合、単一起動として登録されます。複数起動は、グループに含まれるジョブネットでは指定できません。

複数起動指定例

午前8時40分、午後1時、および午後5時30分に起動する場合

```
multitime 0840,1300,1730
```

(7) intervalstart 一定間隔

ジョブネットを一定の間隔で起動する場合に、起動時刻、終了時刻、および起動間隔を指定します。起動間隔の指定範囲は1～240(分)です。なお、起動時刻および終了時刻を指定する場合、登録済の“日変わり時刻”に注意してください。日変わり時刻をはさんで、起動時刻と終了時刻を設定することはできません。一定間隔は、グループに含まれるジョブネットでは指定できません。

一定間隔指定例

午前10時から午後5時30分まで、60分間隔で起動する場合

```
intervalstart 1000,1730,60
```

この例の場合、午後5時がジョブネットの最終起動時刻となります。

(8) iplstart サーバ起動時

ジョブスケジューラサービス/デーモンの起動時にジョブネットを起動する場合に、“ON”を指定します。サーバ起動時は、グループに含まれるジョブネットでは指定できません。

(9) msgonly メッセージ起動

特定のメッセージ事象が発生したときだけジョブネットを起動する場合に、“ON”を指定します。メッセージ起動は、グループに含まれるジョブネットでは指定できません。

(10) msgonlymode メッセージ起動時の処置

待ち合わせているメッセージ事象が1日複数回発生する場合に、メッセージ事象が発生するたびにジョブネットを起動するかどうかを指定します。省略した場合は“EVERYTIME”となります。なお、ジョブネットの起動条件が「メッセージ事象発生時のみ起動」以外を指定した場合、本オペランドは無視されます。

本オペランドで“ONCE”を指定した場合、messagemodeオペランドで“ORD”および“BOT”を指定できません。

EVERYTIME

待ち合わせているメッセージ事象が発生するたびにジョブネットが起動されます。

ONCE

待ち合わせているメッセージ事象が1日複数回発生しても、ジョブネットは最初の1回のみ起動されます。

(11) msgstarttime 開始予定時刻

メッセージ起動する場合の、ジョブネットの開始予定時刻を指定します。

HHMM(時分)の形式で4バイトで指定します。省略した場合、開始予定時刻は設定されません。なお、ジョブネットの起動条件が「メッセージ事象発生時のみ起動」以外を指定した場合、本オペランドは無視されます。

「メッセージ起動時の処置」が“ONCE”の場合のみ指定可能です。

(12) stoptime 停止時刻

ジョブネットの「実行属性」が“INTS”の場合に、停止する時刻を指定します。停止時刻は、HHMM(時分)の形式で4バイトで指定します。省略した場合、停止時刻は指定されません。

停止時刻指定例

午後5時に停止する場合

```
stoptime 1700
```

(13) alarmtime 終了予定時刻

ジョブネットの終了予定時刻を指定します。

HHMM(時分)の形式で4バイトで指定します。省略した場合、終了予定時刻は設定されません。日変わり時刻をはさんで、起動時刻と終了予定時刻を指定することはできません。なお、ジョブネットの起動条件が「単一起動」以外の時は指定できません。

(14) estimatetime 実行予測時間

ジョブネットの実行予測時間を指定します。

数字文字列で指定します。指定範囲は1～9999(分)です。省略した場合、および“0”を指定した場合は実行予測時間は指定されません。

ジョブネットの実行属性に“INTS”を指定した場合には、本オペランドを指定することはできません。

(15) powermode 電源未投入時の処置

ジョブネット起動予定時刻にジョブスケジューラサービス/デーモンが起動されていなかった時の処置を指定します。省略した場合は、“RESET”となります。ジョブネットの起動条件として「単一起動」以外が指定されている場合、および、ジョブネットがグループに含まれる場合は、「電源未投入時の処置」の指定値は無視されます。

RESET

ジョブスケジューラサービス/デーモンが起動されてもジョブネットは起動されません。

SET

ジョブスケジューラサービス/デーモン起動時にジョブネットが起動されます。

(16) iplmode サーバ起動時の処置

サーバ起動時のジョブネットをジョブスケジューラサービス/デーモン起動時ごとに起動するか指定します。省略した場合は“EVERYTIME”となります。

EVERYTIME

ジョブスケジューラサービス/デーモンが起動されるごとにジョブネットが起動されます。

ONCE

1日に複数回ジョブスケジューラサービス/デーモンが起動されてもジョブネットは最初の1回のみ起動されます。

(17) execattr 実行属性

ジョブネットの実行の依頼先を、実行属性として指定します。

【Windows版】

実行属性は以下です。省略した場合は“JES”となり、ジョブ実行制御に実行を依頼します。

JES

ジョブネットの実行をジョブ実行制御に依頼する場合に指定します。

INTS

ジョブネットの実行をInterstageに依頼する場合に指定します。

【UNIX版】

実行属性は以下です。省略した場合は“NORMAL”となり、ジョブスケジューラに実行を依頼します。

NORMAL

旧バージョン互換（旧・標準）属性でジョブスケジューラに依頼する場合に指定します。また、省略した場合の実行属性になります。

JES

ジョブネットの実行をジョブ実行制御に依頼する場合に指定します。

INTS

ジョブネットの実行をInterstageに依頼する場合に指定します。

PAIM

ジョブネットの実行をPowerAIMに依頼する場合に指定します。

(18) carryovermode 運用指示

ジョブネットを持ち越した場合の運用指示を指定します。本オペランドは、マスタスケジュール管理環境が構築され、日ごと管理のスケジュールが行われているときに有効となります。省略した場合は“CANCEL”になります。

CANCEL

打ち切りを指示します。ジョブネットを持ち越した場合に自動的に打ち切ります。

OVER

完了待ちを指示します。ジョブネットを持ち越した場合に、翌日の同一ジョブネットを持ち越したジョブネットが完了した時点でスケジュールします。

PARALLEL

並列走行を指示します。ジョブネットを持ち越した場合に翌日の同一ジョブネットは持ち越したジョブネットの完了の有無にかかわらず、日変わり時刻到来時にスケジュールします。

ACTION

オペレータ対処を指示します。持ち越したジョブネットは停止状態になります。

(19) copystart 複写起動する

複写起動するかどうかを指定します。省略値は“OFF”です。

ON

複写起動します。

OFF

複写起動しません。

(20) jobnetsavenumber 複写先ジョブネットの保存数

複写起動した複写先ジョブネットの保存数を指定します。“1”から“999”までの間で指定します。「複写起動する」が“ON”の場合は省略できません。

(21) defaulthost デフォルトホスト名

ジョブ実行制御に依頼するジョブのデフォルトホストを64バイト以内で指定します。ジョブの制御文で依頼先ホストを省略した場合、デフォルトホストにジョブが投入されます。

ジョブ実行制御属性のジョブネットの場合、実行サブシステム名を使用することができます。

実行サーバがシステムダウンなどで使用できない時に第2候補のサーバへ実行依頼する場合は、“第1候補のホスト名,第2候補のホスト名”の形式で指定します。第2候補のホスト名として、自ホスト(接続先ホスト)を指定することはできません。また、第1候補のホスト名、第2候補のホスト名に実行サブシステム名を使用することはできません。記述形式を以下に示します。

```
defaulthost 第1候補のホスト名,第2候補のホスト名
```



注意

第1候補および第2候補の実行サーバ指定時の注意事項

第1候補および第2候補の実行サーバを指定する場合の注意事項を以下に示します。

- コンマ“,”は、第1候補と第2候補のホスト名の区切り記号として認識されます。コンマを含んだホスト名は指定できません。
- 指定できるホスト名の長さは、第1候補、第2候補のホスト名、およびコンマを含めて64バイト以内です。

(22) standnet 基準ジョブネット宣言

起動日生成の時に基準となるジョブネットであることを宣言する場合に指定します。省略した場合は、“RESET”となります。

RESET

起動日生成の時の基準ジョブネットになりません。

SET

起動日生成の時の基準ジョブネットになります。

(23) execpattern 起動日雛形名

起動日雛形を利用する場合に、登録済みの起動日雛形名を指定します。1つのジョブネットについて４個まで起動日雛形名を指定することができます。複数の起動日雛形名を指定する場合には、半角空白で区切って指定します。

(24) execpatmode 起動日雛形組み合わせ条件

複数の起動日雛形名を指定する時の組み合わせ条件を以下に示すパラメタから選択して指定します。省略した場合は、“AND”が指定されたことになります。

AND

すべての起動日雛形で起動日設定されている日のみを起動日とします。

OR

少なくとも1つの起動日雛形で起動日設定されている日を起動日とします。

(25) basenet 基準ジョブネット名

基準ジョブネットを利用する場合に、基準ジョブネットとして登録されているジョブネット名を指定します。基準ジョブネットであることを宣言している場合は指定することはできません。なお、基準ジョブネットは同一プロジェクトのものしか引用できません。

(26) execday 起動日設定

ジョブネットの起動予定日を設定します。YYYY年MM月DD日を起動日として設定する場合は、“YYYY,MM,DD”と指定します。複数の起動日を設定することができます。今年、来年および再来年の3年分まで指定できます。

起動日設定指定例

2010年10月1日および10日を起動日として設定する場合

```
execday 2010,10,01,10
```

2010年10月1日および12月10日を起動日として設定する場合

```
execday 2010,10,01
execday 2010,12,10
```

(27) cancelday 起動日設定解除

起動を解除したい日付を指定します。“起動日設定”と同じ形式で指定します。

(28) jobschprintr 起動日ビットパターン

jobschprint -rコマンドにより展開される起動日設定のビットパターンデータを指定します(jobschprintコマンドにより生成された値以外が指定された場合の動作は保証されません。また、同一バージョンのjobschprintコマンドによる展開結果以外の動作は保証されません。jobschprintコマンドにより展開した年号と違う年に本パラメタを指定することはできません。)。ビットパターンデータを指定した場合、3年間はjobschprint -rコマンドにより展開したジョブネットの起動日と同一の起動日を設定することができます。なお、本オペランドを指定した直後は、他の起動日設定用オペランド(毎月カレンダ、毎年カレンダ、基準ジョブネット名、起動日雛形、カレンダ名、休日シフト、オフセット、有効期間/無効期間など)による起動日の再計算は実施されません。また、他ジョブネットの起動日には反映されません。

(29) annuallydate 毎年の日付指定

毎年カレンダを利用する場合に、毎年のジョブネットを起動する日付を指定します。MM月DD日を起動日として設定する場合は、“MMDD”と指定します。複数の起動日を設定することができます。

(30) annuallyweek 毎年の曜日指定

毎年カレンダを利用する場合に、ジョブネットを起動する曜日を月ごとに指定します。1つのオペランドで1つの月しか設定できません。複数の月を設定する場合は、本オペランドを複数回指定します。指定する形式は、何月かを表す数値を1～12の範囲で指定し、その後、曜日を表す英単語の先頭3文字を半角文字で指定します。“12,SUN,MON,TUE,WED,THU,FRI,SAT”のような形式で指定します。

(31) monthlydate 毎月の日付指定

毎月カレンダを利用する場合に、毎月のジョブネットを起動する日付を指定します。毎月DD日を起動日として設定する場合は、“DD”と指定します。複数の起動日を設定することができます。なお、毎月月末に起動する場合は“LAST”と指定します。

(32) monthlyweek 毎月の曜日指定

毎月カレンダを利用する場合に、毎月のジョブネットを起動する曜日を週ごとに指定します。1つのオペランドの指定で1つの週しか設定できません。複数の週を設定する場合は、本オペランドを複数回指定します。指定する形式は、何週目かを表す数値を0～6(第1～第5まで。0は毎週の指定。6は月の最終週)の範囲で指定し、その後、曜日を表す英単語の先頭3文字を半角文字で指定します。“5,SUN,MON,TUE,WED,THU,FRI,SAT”のような形式で指定します。

(33) bnsmode 営業日の選択基準

基準日情報“営業日”の指定の場合に、月初め、月締めのどちらを基準とするかを指定します。

START

営業日を月初めから数えます。

END

営業日を月締めから逆算します。

(34) bnsbase 営業日の月初め日

基準日情報“営業日”指定の場合に、月初め日を指定します。“0”は指定できません。

1〜28

1日から28日です。

LAST

月末です。

(35) bnsday 営業日

基準日情報“営業日”指定の場合に、営業日を指定します。複数指定した場合には、“,”(コンマ)で区切って指定します。“0”は指定できません。

1〜31

第1営業日（月初め）から数えた営業日です。bnsmodeオペランドで“END”を指定した場合は指定できません。

-1〜-31

第-1営業日（月締め）から逆算した営業日です。bnsmodeオペランドで“START”を指定した場合は指定できません。

(36) bnslimit 月度内の営業日のみを起動日とする

基準日情報“営業日”指定の場合に、“月度内の営業日のみを起動日とする”の有無（ON/OFF）を指定します。bnsmodeオペランドが指定されている場合の省略値は“ON”です。

ON

“月度内の営業日のみを起動日とする”を指定します。

OFF

“月度内の営業日のみを起動日とする”を指定しません。

(37) offsetmode オフセット

基準日から移動する日数を数える方法を以下に示すパラメタから選択して指定します。省略した場合は、“EXECDAY”が指定されたことになります。

EXECDAY

カレンダの稼働日で移動する日数を数えます。

DAY

休日には関係なく、絶対日数で移動する日数を数えます。

(38) offsetday オフセット日数

基準日から移動する日数を“-99”から“99”までの間で指定します。省略した場合は“0”となります。

(39) offsetholiday 基準日が休日の場合は稼働日とする

基準日から移動する日数を数える時、基準日が休日であった場合の扱いを指定します。省略した場合は、“OFF”が指定されたことになります。

オフセットに“DAY”を指定した場合には、本オペランドを指定することはできません。

ON

基準日が休日の場合も稼働日とし、移動する日数の基準日とします。

OFF

基準日が休日の場合は稼働日とせず、移動する日数の基準日としません。

(40) holidayshift 休日シフト

起動日の登録および起動日雛形で設定される起動日が、休日と重なった場合の対処方法を以下に示すパラメタから選択して指定します。

BEF

ジョブネットの起動日を前日に変更します。

AFT

ジョブネットの起動日を翌日に変更します。

NOT

当日を休日として認識し、ジョブネットを起動しません。

IGN

当日を起動日として認識し、ジョブネットを起動します。

(41) shiftlimit シフト上限日数

休日シフトを行う時に休日が連続した時に起動を抑止する日数“0”から“99”までの間で指定します。省略した場合は、“0”となります。

(42) holidaycalendar カレンダ名

起動日を生成するときに元となる、カレンダ名を指定します。1つのジョブネットについて5個までカレンダを指定することができます。

複数のカレンダを指定する場合には、“,”で区切って指定します。

(43) holidaymode カレンダ組み合わせ条件

複数のカレンダを指定する時の組み合わせ方を以下に示すパラメタから選択して指定します。省略した場合は、“AND”が指定されたことになります。

AND

すべてのカレンダで休日設定されている日のみを休日とします。

OR

少なくとも1つのカレンダで休日設定されている日を休日とします。

(44) enablestart 有効期限開始日/enableend 有効期限終了日

ジョブネットを運用する期間を“YYYY,MM,DD”の形式で指定します。有効期限開始日を省略した場合、即日から有効期限終了日まで有効となります。有効期限終了日を省略した場合、有効期限開始日以降、有効となります。有効期限開始日、有効期限終了日共に省略した場合は、すべての日が有効となります。

(45) disablestart 無効期限開始日/disableend 無効期限終了日

ジョブネットを運用しない期間を“YYYY,MM,DD”の形式で指定します。無効期限開始日を省略した場合、即日から無効期限終了日まで無効となります。無効期限終了日を省略した場合、無効期限開始日以降、無効となります。無効期限開始日、無効期限終了日共に省略した場合は、すべての日が有効となります。

(46) tstarttime 単一起動(一時変更)

ジョブネットを1日に1回起動する場合に、起動時刻をHHMM(時分)の形式で4バイトで指定します。なお、本指定は、一時変更開始日から一時変更終了日までの期間が有効になります。

(47) tmultitime 複数起動(一時変更)

ジョブネットを1日に複数回起動する場合に、起動時刻をHHMM(時分)の形式で4バイト以内で指定します。起動時刻は6個まで指定することができます。なお、本指定は、一時変更開始日から一時変更終了日までの期間が有効になります。起動時刻を1個しか指定しなかった場合、単一起動(一時変更)として登録されます。複数起動(一時変更)は、グループに含まれるジョブネットでは指定できません。

(48) tintervalstart 一定間隔(一時変更)

ジョブネットを一定の間隔で起動する場合に、起動時刻、終了時刻、および起動間隔を指定します。起動間隔の指定範囲は1～240(分)です。日変わり時刻をはさんで、起動時刻と終了時刻を指定することはできません。なお、本指定は、一時変更開始日から一時変更終了日までの期間が有効になります。一定間隔(一時変更)はグループに含まれるジョブネットでは指定できません。

(49) tiplstart サーバ起動時(一時変更)

ジョブスケジューラサービス/デーモンの起動時にジョブネットを起動する場合に,“ON”を指定します。なお、本指定は、一時変更開始日から一時変更終了日までの期間が有効になります。サーバ起動時(一時変更)はグループに含まれるジョブネットでは指定できません。

(50) tmsgonly メッセージ起動(一時変更)

特定のイベント事象が発生したときだけジョブネットを起動する場合に、“ON”を指定します。なお、本指定は、一時変更開始日から一時変更終了日までの期間が有効になります。メッセージ起動(一時変更)はグループに含まれるジョブネットでは指定できません。

(51) tmsgonlymode メッセージ起動日の処置(一時変更)

待ち合わせているメッセージ事象が1日複数回発生する場合に、メッセージ事象が発生するたびにジョブネットを起動するかどうかを指定します。省略した場合は“EVERYTIME”となります。
本指定は、一時変更開始日から一時変更終了日までの期間が有効になります。なお、ジョブネットの起動条件が「メッセージ事象発生時のみ起動」以外を指定した場合、本オペランドは無視されます。

EVERYTIME

待ち合わせているメッセージ事象が発生するたびにジョブネットが起動されます。

ONCE

待ち合わせているメッセージ事象が1日複数回発生しても、ジョブネットは最初の1回のみ起動されます。

(52) tmsgstarttime 開始予定時刻(一時変更)

メッセージ起動する場合の、ジョブネットの開始予定時刻を指定します。

HHMM(時分)の形式で4バイトで指定します。省略した場合、開始予定時刻は設定されません。
本指定は、一時変更開始日から一時変更終了日までの期間が有効になります。なお、ジョブネットの起動条件が「メッセージ事象発生時のみ起動」以外を指定した場合、本オペランドは無視されます。

「メッセージ起動時の処置」が“ONCE”の場合のみ指定可能です。

(53) tstoptime 停止時刻(一時変更)

ジョブネットの「実行属性」が“INTS”の場合に、停止する時刻を指定します。停止時刻は、HHMM(時分)の形式で4バイトで指定します。省略した場合、停止時刻は指定されません。なお、本指定は、一時変更開始日から一時変更終了日までの期間が有効になります。

(54) talarmtime 終了予定時刻(一時変更)

ジョブネットの終了予定時刻(一時変更)を指定します。

HHMM(時分)の形式で4バイトで指定します。省略した場合、終了予定時刻は設定されません。日変わり時刻をはさんで、起動時刻と終了予定時刻を指定することはできません。なお、本指定は、一時変更開始日から一時変更終了日までの期間が有効になります。

(55) tpowermode 電源未投入時の処置(一時変更)

ジョブネット起動予定時刻にジョブスケジューラサービス/デーモンが起動されていなかった時の処置を指定します。省略した場合は、“RESET”となります。なお、本指定は、一時変更開始日から一時変更終了日までの期間が有効になります。ジョブネットの一時変更の起動条件として「単一起動(一時変更)」以外が指定されている場合、および、ジョブネットがグループに含まれている場合、「電源未投入時の処置(一時変更)」の指定値は無視されます。

RESET

ジョブスケジューラサービス/デーモンが起動されてもジョブネットは起動しません。

SET

ジョブスケジューラサービス/デーモン起動時にジョブネットを起動します。

(56) tiplmode サーバ起動時の処置(一時変更)

サーバ起動時(一時変更)のジョブネットをジョブスケジューラサービス/デーモン起動時ごとに起動するか否かを指定します。省略した場合は、“EVERYTIME”となります。なお、本指定は、一時変更開始日から一時変更終了日までの期間が有効になります。

EVERYTIME

ジョブスケジューラサービス/デーモンが起動されるごとにジョブネットが起動します。

ONCE

1日に複数回ジョブスケジューラサービス/デーモンが起動されてもジョブネットは最初の1回のみ起動します。

(57) temporarystart 一時変更開始日/temporaryend 一時変更終了日

ジョブネットを一時変更する期間を“YYYY,MM,DD”の形式で指定します。一時変更開始日のみを指定した場合は、指定した当日のみが一時変更の対象になります。

(58) messagemode メッセージ条件

メッセージ事象の待ち合わせ条件(AND/OR)、起動条件(ORD/COG/BOT/UNC)、および起動時刻との条件(ANDTIME/ORTIME)を組み合わせて指定します。messagemodeオペランドを省略した場合、パラメタとして“AND,UNC,ORTIME”が指定されたことになります。「起動時刻との条件」として“ANDTIME”を指定した場合、または、ジョブネットがグループに含まれる場合は、「起動条件」として“ジョブネット実行中も有効”(“ORD”および“BOT”)は指定できません。ジョブネットがグループに含まれる場合、「起動時刻との条件」で指定した値は無視されます。

AND

登録してあるメッセージ事象のすべてが発生した場合に、ジョブネットを起動します。

OR

登録してあるメッセージ事象のうちどれか1つでも発生した場合に、ジョブネットを起動します。

ORD

“ジョブネット実行中も有効”です。ジョブネットが実行中に発生したメッセージ事象が有効となります。その場合、ジョブネットが終了次第、実行中にメッセージ事象が発生した回数分、再びジョブネットが起動されます。ORDパラメタを指定しない場合、ジョブネットが実行中に発生したメッセージ事象は無効となります。

COG

“起動日のみ有効”です。ジョブネットの起動予定日に登録したメッセージ事象が発生した場合、ジョブネットを起動します。

BOT

“ジョブネット実行中も有効&起動日のみ有効”です。ジョブネットの起動予定日に登録したメッセージ事象が発生した場合、メッセージが発生した回数分ジョブネットを起動します。

UNC

起動条件を設定しない場合に指定します。

ANDTIME

起動予定時刻が到来し、待ち合わせていたメッセージ事象が発生した時にジョブネットが起動されます。ジョブネットの起動条件が“単一起動”(一時変更期間中は“単一起動(一時変更)”)の場合のみ有効になります。

ORTIME

メッセージ事象が発生する度にジョブネットが起動されます。また、メッセージ事象発生の有無に関わりなく、起動時刻になると、ジョブネットが起動されます。

(59) carryover 持ち越し処理

日変わり時刻が到来したときの持ち越し処理の指定をします。「起動時刻の到来を待つ」(messagemodeオペランドに“ANDTIME”が指定されている)場合、または起動条件が「メッセージ事象発生時のみ起動」(msgonlyオペランドに“ON”が指定されている)の場合に指定できます。省略値は“RESET”です。

RESET

日変わり時刻の到来時に、ジョブネットの持ち越し処理を行いません。

SET

日変わり時刻の到来時に、ジョブネットの持ち越し処理を行います。

(60) messagename メッセージ事象名

ジョブネットの起動の契機となるメッセージ事象を12バイト以内で指定します。“,”、“:”、空白、半角カナ、およびユーザ定義文字を含むことはできません。CSVファイルでメッセージ事象名を扱う機能(jobschdeftocsvコマンド, jobschcsvtodefコマンド, Job Designer、インポート・エクスポート機能)を使用する場合は、“;”を含むこともできません。70個まで指定できます。

メッセージ事象指定例

MSG1,MSG2,MSG3の3個のメッセージ事象を登録する場合

```
messagename MSG1,MSG2,MSG3
```

(61) job ジョブネット内ジョブ数

ジョブネットを構成するジョブの数を指定します。指定範囲は1～255です。ただし、Interstage属性のジョブネットでは1つしか指定できません。省略した場合、ジョブ制御文は記述できません。

(62) operate 操作

操作による停止または無効の状態を表します。省略した場合、登録時は“NONE”になり、変更時は状態の変更はされません。

PAUSE

停止

DISABLE

無効

NONE

登録時は実行待ちになります。変更時は停止状態を停止解除、または無効状態を無効解除します。停止状態、無効状態以外の状態のときは、状態は変化しません。

注意事項

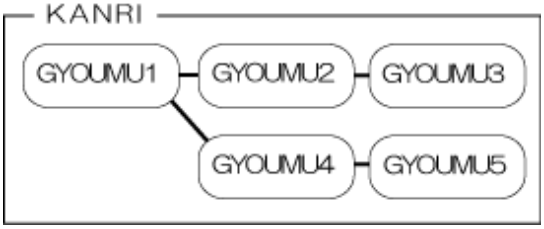
以下のオペランドを含む制御文を利用して、ジョブネットを変更した場合、起動日が再作成されます。

この変更により、[起動日]ウィンドウ、[起動日(プレビュー)]ウィンドウおよび[運用予定]ウィンドウで指定した起動日が、制御文の内容で上書きされる場合があります。また、変更したジョブネットが基準ジョブネットの場合、その基準ジョブネットの起動日を利用しているジョブネットについても、起動日が再作成されます。

execpattern、basenet、execday、cancelday、jobschprintr、annualydate、annualyweek、monthlydate、monthlyweek、bnsmode、bnsbase、bnsday、bnslimit、offsetmode、offsetday、holidaycalendar、holidaymode、holidayshift、shiftlimit、enablestart、enableend、disablestart、disableend

ジョブネット制御文指定例

下図のようなジョブ構成を持つジョブネットを定義する場合の制御文を図の下に示します。




```
NET
jobnetname KANRI                ← KANRIのジョブネット名
jobnetcomment 業務管理          ← KANRIのジョブネット名称
starttime 0850                  ← 8時50分から起動します。
execpattern 毎月20日     ← 起動日雛形名「毎月20日」を
                                   ジョブネットの起動日にします。
holidaycalendar SYSTEM_CALENDAR  ← カレンダとして「SYSTEM_CALENDAR」を利用します。
holidayshift BEF      ← カレンダの休日とジョブネットの起動日が重なった時ジョブネットの起動日を前日に変更します。
job 5                   ← ジョブ制御文を5つ定義します。
;
JOB
jobname GYOUMU1.bat
jobnumber 1
;
JOB
jobname GYOUMU2.bat
jobnumber 2
pre_job 1
;
JOB
jobname GYOUMU3.bat
jobnumber 3
pre_job 2
;
JOB
jobname GYOUMU4.bat
jobnumber 4
```

```
pre_job 1
;
JOB
jobname GYOUMU5.bat
jobnumber 5
pre_job 4
;
```

4.12.2　ジョブ制御文

ジョブの登録情報を設定する制御文です。なお、ジョブ制御文を指定する時は、ジョブネット制御文を必ず直前に指定してください。

記述形式

ジョブ実行制御属性のジョブ制御文

```
JOB                                                  ---------- (1)
jobname ジョブ                                       ---------- (2)  (*)   (注1)
jobnumber ジョブ番号                                 ---------- (3)  (*)   (注2)
jobattr ジョブ属性                                   ---------- (4)        (注4)
jobname_jes ジョブ実行制御でのジョブ名               ---------- (5)        (注8)
masterlinkjobnet マスタリンクジョブネット名          ---------- (6)        (注17)
jobparam パラメタ                                    ---------- (7)        (注8)
directory ディレクトリ                               ---------- (8)        (注8)
jobcomment コメント                                  ---------- (9)        (注8)
req_host 依頼ホスト名                                ---------- (10)       (注8)
queuename キュー名                                   ---------- (11)       (注8)
normallimit 正常終了コード上限値                     ---------- (12)       (注8)
endcodeignore 終了コードに関わらず正常               ---------- (13)       (注8)
nooutput 出力を保存しない                            ---------- (14)       (注8)
restartpoint 再起動ポイント                          ---------- (15)       (注5)
pre_job 先行ジョブ番号                               ---------- (16)
pre_job_endcode 先行ジョブの終了コードによる起動条件 --------- (17)       (注3)(注4)
execdaymode 起動日のみ有効                           ---------- (18) (注12)(注13)(注14)(注15)
execuser 実行ユーザ名                                ---------- (19)       (注8)
resource リソース名                                  ---------- (20)       (注8)
resource_attr リソース割り当て                       ---------- (21)       (注7)(注8)
stdoutfile 標準出力ファイル名                        ---------- (22)       (注8)
stderrfile 標準エラー出力ファイル名                  ---------- (23)       (注8)
errtostdfile 標準エラー出力を標準出力ファイルに出力 ---------- (24)       (注8)(注10)
jobfile_trans 実行コマンドの転送                     ---------- (25)       (注8)
priority 実行優先度                                  ---------- (26)       (注8)
predictiontime 実行予測時間                          ---------- (27)       (注8)
limittime 打ち切り時間                               ---------- (28)       (注8)
netvar_prefix ジョブネット変数のプレフィックス       ---------- (29)       (注8)
envname 環境変数名                                   ---------- (30)       (注8)
envvalue 環境変数値                                  ---------- (31)       (注8)(注9)
joblogfile ジョブログファイル名                      ---------- (32)       (注11)
stepname 開始ステップ名                              ---------- (33)       (注11)
codelaststep 最後に実行されたジョブステップの終了コードを --- (34)       (注11)
             ジョブの終了コードとする
operate 操作                                         ---------- (35)
jobicon アイコン種別                                 ---------- (36)       (注6)
iconposition アイコン位置                            ---------- (37)       (注16)
;
```

注1)

jobnameオペランドは、JOBキーワードのすぐ後に指定してください。

注2)

jobnumberオペランドは、jobnameオペランドのすぐ後に指定してください。

注3)

pre_job_endcodeオペランドを指定する場合、pre_jobオペランドに先行ジョブ番号が1つだけ指定されている必要があります。

注4)

pre_job_endcodeオペランドは、jobattrオペランドと同時に指定することはできません。

注5)

再起動ポイントはジョブネットを構成するジョブの中で1つのジョブだけに設定できます。

注6)

“GLJO”,“GLBO”,“INTS”以外の値を指定してください。【Windows版】。“INTS”以外の値を指定してください。【UNIX版】

注7)

resource_attrオペランドは、resourceオペランドのすぐ後に指定してください。

注8)

アイコン種別でジョブネット“JNET”、または、リンクジョブネット“LNET”を指定する場合には、指定できません。

注9)

envvalueオペランドは、envnameオペランドのすぐ後ろに指定してください。

注10)

stderrfileオペランドとは同時に指定できません。

注11)

アイコン種別がバッチフレームワークジョブ“BJOB”の場合のみ指定できます。

注12)

アイコン種別でジョブネット“JNET”、または、リンクジョブネット“LNET”を指定する場合のみ指定できます。

注13)

jobattrオペランドが“RECOVERYONLY”、“RECOVERYRERUN”指定の場合は指定できません。

注14)

restartpointオペランドが“ON”指定の場合は指定できません。

注15)

pre_job_endcodeオペランドに指定がある場合は指定できません。また、後続のジョブに、pre_job_endcodeオペランドに指定がある場合は指定できません。

注16)

ジョブネットを構成するすべてのジョブに指定する必要があります。

注17)

アイコン種別でリンクジョブネット“LNET”を指定する場合は、必ず指定してください。

(*)：

必ず指定してください。

Interstage属性のジョブ制御文

```
JOB                                  ---------- (1)
workunitname ワークユニット名          ---------- (38) (*) (注1)
jobnumber ジョブ番号                   ---------- (3)  (*) (注2)
systemname システム名                  ---------- (39)
jobcomment コメント                    ---------- (9)
req_host 依頼ホスト名                  ---------- (10)
queuename キュー名                     ---------- (11)
resource リソース名                    ---------- (20)
resource_attr リソース割り当て         ---------- (21)      (注4)
normallimit 正常終了コード上限値       ---------- (12)
endcodeignore 終了コードに関わらず正常 ---------- (13)
operate 操作                         ---------- (35)
jobicon アイコン種別                   ---------- (36) (*) (注3)
iconposition アイコン位置              ---------- (37)      (注5)
;
```

注1)

workunitnameオペランドは、JOBキーワードのすぐ後に指定してください。

注2)

jobnumberオペランドは、workunitnameオペランドのすぐ後に指定してください。

注3)

必ず“INTS”を指定してください。Interstage属性のジョブネットに “INTS”以外を指定した場合または Interstage属性以外のジョブネットに“INTS”を指定した場合、jobschsetnetコマンドが異常終了します。

注4)

resource_attrオペランドは、resourceオペランドのすぐ後に指定してください。

注5)

ジョブネットを構成するすべてのジョブに指定する必要があります。

(*)：

必ず指定してください。

PowerAIM属性のジョブ制御文【UNIX版】

```
JOB                                          ---------- (1)
systemname 業務システム名                    ---------- (40) (*) (注1)
workunitname ワークユニット名                ---------- (38) (*) (注1)
jobnumber ジョブ番号                         ---------- (3)  (*) (注2)
jobattr ジョブ属性                           ---------- (4)      (注4)
jobcomment コメント                          ---------- (9)
normallimit 正常終了コード上限値             ---------- (12)
endcodeignore 終了コードに関わらず正常       ---------- (13)
nooutput 出力を保存しない                    ---------- (14)
restartpoint 再起動ポイント                  ---------- (15)      (注5)
pre_job 先行ジョブ番号                       ---------- (16)
pre_job_endcode 先行ジョブの終了コードによる起動条件 ---------- (17)      (注3)(注4)
operate 操作                                 ---------- (35)
jobicon アイコン種別                         ---------- (36)
iconposition アイコン位置                    ---------- (37)      (注6)
;
```

注1)

systemnameオペランドとworkunitnameオペランドは、JOBキーワードのすぐ後ろに続けて指定してください。

注2)

jobnumberオペランドは、workunitnameオペランドのすぐ後に指定してください。

注3)

pre_job_endcodeオペランドを指定する場合、pre_jobオペランドに先行ジョブ番号が1つだけ指定されている必要があります。

注4)

pre_job_endcodeオペランドは、jobattrオペランドと同時に指定することはできません。

注5)

再起動ポイントはジョブネットを構成するジョブの中で1つのジョブだけに設定できます。

注6)

ジョブネットを構成するすべてのジョブに指定する必要があります。

(*)：

必ず指定してください。

旧バージョン互換（旧・標準）属性のジョブ制御文【UNIX版】

```
JOB                                                ---------- (1)
jobname ジョブ                                     ---------- (2)  (*)  (注1)
jobnumber ジョブ番号                               ---------- (3)  (*)  (注2)
jobattr ジョブ属性                                 ---------- (4)       (注4)
directory ディレクトリ                             ---------- (8)
jobcomment コメント                                ---------- (9)
normallimit 正常終了コード上限値                   ---------- (12)
endcodeignore 終了コードに関わらず正常             ---------- (13)
nooutput 出力を保存しない                          ---------- (14)
restartpoint 再起動ポイント                        ---------- (15)      (注5)
pre_job 先行ジョブ番号                             ---------- (16)
pre_job_endcode 先行ジョブの終了コードによる起動条件 -------- (17)      (注3)(注4)
operate 操作                                       ---------- (35)
jobicon アイコン種別                               ---------- (36)
iconposition アイコン位置                          ---------- (37)      (注6)
;
```

注1)

jobnameオペランドは、JOBキーワードのすぐ後に指定してください。

注2)

jobnumberオペランドは、jobnameオペランドのすぐ後に指定してください。

注3)

pre_job_endcodeオペランドを指定する場合、pre_jobオペランドに先行ジョブ番号が1つだけ指定されている必要があります。

注4)

pre_job_endcodeオペランドは、jobattrオペランドと同時に指定することはできません。

注5)

再起動ポイントはジョブネットを構成するジョブの中で1つのジョブだけに設定できます。

注6)

ジョブネットを構成するすべてのジョブに指定する必要があります。

(*):

必ず指定してください。

GEE

ジョブ制御文

```
JOB                                          ---------- (1)
jobid 業務ＩＤ／ホストのジョブ名             ---------- (43) (*)  (注1)
user_definition ユーザ定義名                 ---------- (44) (*)  (注2)
exec_host ジョブ実行ホスト名                 ---------- (47) (*)  (注3)
job_definition 業務定義                      ---------- (41) (*)  (注4)
filename 実行ファイル名／JCM名／データセット名 ---------- (45)      (注5)
jobnumber ジョブ番号                         ---------- (3)  (*)  (注6)
jobattr ジョブ属性                           ---------- (4)
job_kind 業務種別定義                        ---------- (42)      (注7)
option ジョブへのパラメタ                    ---------- (46)
alternate_host 代替ホスト名                  ---------- (48)
retry リトライ間隔                           ---------- (49)
jobcomment コメント                          ---------- (9)
queuename キュー名                           ---------- (11)
normallimit 正常終了コード上限値             ---------- (12)
endcodeignore 終了コードに関わらず正常       ---------- (13)
nooutput 出力を保存しない                    ---------- (14)
restartpoint 再起動ポイント                  ---------- (15)      (注11)
pre_job 先行ジョブ番号                       ---------- (16)
pre_job_endcode 先行ジョブの終了コードによる起動条件 ------- (17)      (注9)(注10)
execuser 実行ユーザ名                        ---------- (19)
resource リソース名                          ---------- (20)
resource_attr リソース割り当て               ---------- (21)      (注8)
stdoutfile 標準出力ファイル名                ---------- (22)
stderrfile 標準エラー出力ファイル名          ---------- (23)
errtostdfile 標準エラー出力を標準出力ファイルに出力 -------- (24)
priority 実行優先度                          ---------- (26)
predictiontime 実行予測時間                  ---------- (27)
limittime 打ち切り時間                       ---------- (28)
envname 環境変数名                           ---------- (30)
envvalue 環境変数値                          ---------- (31)
operate 操作                                 ---------- (35)
jobicon アイコン種別                         ---------- (36) (*)
iconposition アイコン位置                    ---------- (37)      (注12)
;
```

注1)

jobidオペランドは、JOBキーワードのすぐ後に指定してください。

注2)

user_definitionオペランドは、jobidオペランドのすぐ後に指定してください。

注3)

exec_hostオペランドは、user_definitionオペランドのすぐ後に指定してください。

注4)

job_definitionオペランドは、exec_hostオペランドのすぐ後に指定してください。

注5)

job_definitionオペランドの指定が“OFF”の場合は必ず指定してください。“ON”の場合は指定できません。

注6)

jobnumberオペランドは、filenameオペランドの指定がある場合はfilenameオペランドのすぐ後に指定します。filenameオペランドの指定がない場合はjob_definitionオペランドのすぐ後に指定します。

注7)

job_kindオペランドは、jobiconオペランドで“MXSP”を指定し、かつSymfoNETに業務定義されていない場合のみ指定できます。

注8)

resource_attrオペランドは、resourceオペランドのすぐ後に指定してください。

注9)

pre_job_endcodeオペランドを指定する場合、pre_jobオペランドに先行ジョブ番号が1つだけ指定されている必要があります。

注10)

pre_job_endcodeオペランドは、jobattrオペランドと同時に指定することはできません。

注11)

再起動ポイントは、ジョブネットを構成するジョブの中で1つのジョブだけに設定できます。

注12)

ジョブネットを構成するすべてのジョブに指定する必要があります。

(*)

必ず指定してください。

制御文の説明

(1) JOB

ジョブ制御文の開始を表すキーワードです。

(2) jobname ジョブ

ジョブとして起動するバッチプログラム名、コマンド名、またはプログラム名をパスを含めて260バイト以内で指定します。また、ファイル名とパラメタを合わせて300バイト以内で指定します。半角カナおよびユーザ定義文字、以下の半角文字を含むことはできません。

$ `

UTF-8コードの環境(ネットワークジョブの場合、投入サーバまたは実行サーバのいずれかがUTF-8コードの環境)で、指定するファイル名に日本語を含む場合は、必ず日本語コードを含まないジョブ名を指定してください。

アイコン種別でジョブネット“JNET”を指定する場合、子ジョブネットのジョブネット名を指定します。

アイコン種別でバッチフレームワークジョブ“BJOB”を指定する場合、バッチジョブ定義名を64バイト以内で指定します。英小文字(a-z)、英大文字(A-Z)、数字(0-9)、半角ハイフン(-)および半角アンダーバー(_)のみ使用できます。また、1文字目は英小文字(a-z)、英大文字(A-Z)にしてください。

アイコン種別でリンクジョブネット“LNET”を指定する場合、リンクジョブネットのジョブネット名を50バイト以内で指定します。他のジョブネットやリンクジョブネットのジョブネット名と重複しないでください。/ ? * ; , ¥ " < > | : ' . 空白、半角カナおよびユーザ定義文字を含むことはできません。

ジョブ実行制御のqsubコマンドやジョブ実行制御のqsubコマンドを呼び出しているバッチファイル/シェルスクリプトを指定しないでください。

(3) jobnumber ジョブ番号

ジョブに対応する番号を1～255の数字から割り当てます。ジョブの実行順序に従い、重複しないように番号を割り当てます。先行ジョブから後続ジョブへ昇順になるように番号を割り当てます。

(4) jobattr ジョブ属性

リカバリジョブの指定です。先行ジョブ番号を必ず指定する必要があります。また、リカバリジョブの先行ジョブを再実行するかどうかを指定します。

RECOVERYONLY

このジョブをリカバリジョブにします。このリカバリジョブが正常終了しても先行ジョブは再実行しません。

RECOVERYRERUN

このジョブをリカバリジョブにします。このリカバリジョブが正常終了したら先行ジョブを再実行します。

(5) jobname_jes ジョブ実行制御でのジョブ名

個々のジョブを区別するためのジョブ名を64バイト以内で指定します。空白、半角カナ、ユーザ定義文字および以下の半角文字はジョブ名として使用できません。

/ ? * ; , ¥ " < > | : ( ) ' . $ `

1文字目に # を指定することはできません。

ネットワークジョブおよび分散実行ジョブの場合は、& を指定することはできません。

ジョブ実行制御で使用するジョブ名としても利用されます。

UTF-8コードの環境(ネットワークジョブの場合、投入サーバまたは実行サーバのいずれかがUTF-8コードの環境)では日本語コードを使用しないでください。

省略時には、jobnameで指定した文字列(パス名および拡張子を取り除いたもの)が、jobname_jesとして扱われます。このため、UTF-8コードの環境(ネットワークジョブの場合、投入サーバまたは実行サーバのいずれかがUTF-8コードの環境)で、jobnameに日本語を含む場合は、本指定で日本語コードを含まない文字列を必ず指定してください。

(6) masterlinkjobnet マスタリンクジョブネット名

アイコン種別でリンクジョブネット“LNET”を指定する場合、リンクジョブネットの参照先に指定するマスタリンクジョブネットのジョブネット名を指定します。

(7) jobparam パラメタ

コマンドの引数として渡す文字列(パラメタ)を1024バイト以内で指定します。

半角空白を含む文字列をパラメタとして指定する場合は、1つのパラメタとする文字列全体をダブルクォーテーション(")で囲む必要があります。

アイコン種別でバッチフレームワークジョブ“BJOB”を指定する場合、指定できません。

(8) directory ディレクトリ

ジョブ実行制御属性の場合、ジョブ実行制御のqsubコマンドを実行するディレクトリを64バイト以内で指定します。空白、半角カナ、ユーザ定義文字を含むことはできません。

アイコン種別でバッチフレームワークジョブ“BJOB”を指定する場合、指定できません。

旧バージョン互換（旧・標準）属性の場合、ジョブを実行するディレクトリを指定します。【UNIX版】

(9) jobcomment コメント

ジョブへのコメントを64バイト以内で指定します。半角カナおよびユーザ定義文字を含むことはできません。

(10) req_host 依頼ホスト名

ジョブの投入先ホスト名を64バイト以内で指定します。省略した場合は、ジョブネットで指定されているデフォルトホストに対しジョブが投入されます。

ジョブ実行制御属性のジョブネットの場合、実行サブシステム名を使用することができます。

アイコン種別でバッチフレームワークジョブ“BJOB”を指定する場合、指定できません。

実行サーバがシステムダウンなどで使用できない時に第2候補のサーバへ実行依頼する場合は、“第1候補のホスト名,第2候補のホスト名”の形式で指定します。第2候補のホスト名として、自ホスト(接続先ホスト)を指定することはできません。また、第1候補のホスト名、第2候補のホスト名に実行サブシステム名を使用することはできません。記述形式を以下に示します。

```
req_host 第1候補のホスト名,第2候補のホスト名
```

注意

第1候補および第2候補の実行サーバ指定時の注意事項

第1候補および第2候補の実行サーバを指定する場合の注意事項を以下に示します。

- コンマ“,”は、第1候補と第2候補のホスト名の区切り記号として認識されます。コンマを含んだホスト名は指定できません。
- 指定できるホスト名の長さは、第1候補、第2候補のホスト名、およびコンマを含めて64バイト以内です。

(11) queuename キュー名

ジョブの投入先キュー名を英数字15バイト以内で指定します。省略した場合は、デフォルトキューにジョブが投入されます。

アイコン種別でバッチフレームワークジョブ“BJOB”を指定する場合、ジョブキュー名を15バイト以内で指定します。英小文字(a-z)、英大文字(A-Z)、数字(0-9)、半角ハイフン(-)および半角アンダーバー(_)のみ使用できます。また、1文字目は英小文字(a-z)、英大文字(A-Z)にしてください。省略可能です。

(12) normallimit 正常終了コード上限値

ジョブが正常に終了したか、異常終了したかを判断する値を指定します。

ジョブの終了コードが、0以上、正常終了コード上限値以下の場合、ジョブは正常に終了したとみなされます。ジョブの終了コードが、正常終了コード上限値で指定した値よりも大きい場合、ジョブは異常終了したとみなされます。

指定できる範囲は0から255までです。省略した場合は、0になります。

[ジョブネット起動パラメタの定義]ウィンドウで疑似正常状態を有効にしている場合、ジョブの状態は、終了コード0のジョブが“正常終了”状態、終了コードが1以上正常終了コード上限値以下のジョブは“疑似正常”として、区別して表示されることに注意してください。

(13) endcodeignore 終了コードに関わらず正常

ジョブの終了コードを無視する場合、“ON”を指定します。“ON”を指定すると、終了コードに関係なくジョブは正常終了となり、後続のジョブを実行します。

(14) nooutput 出力を保存しない

ジョブの標準出力/エラー出力を参照しない場合に“ON”を指定します。“ON”を指定すると、ジョブの標準出力/エラー出力は保存されません。

アイコン種別でバッチフレームワークジョブ“BJOB”を指定する場合、標準出力/標準エラー出力をジョブログファイルに保存しないときに“ON”を指定します。

(15) restartpoint 再起動ポイント

ジョブネットの再起動時、このジョブから起動する場合に“ON”を指定します。再起動ポイントは、ジョブネットを構成するジョブの中で1つのジョブにだけ設定できます。なお、リカバリジョブには再起動ポイントを指定することはできません。

(16) pre_job 先行ジョブ番号

先行ジョブのジョブ番号を指定します。先行ジョブ番号を複数指定する場合は、コンマ“,”で区切ります。指定が多くなる場合は、昇順に並べておくと効率よく確認できます。

先行ジョブの終了をOR条件で待ち合わせる場合は、先頭をORとコンマ“OR,”とし、それに続いて先行ジョブ番号を指定してください。“OR,”を指定する場合、以下の指定はできません。

- リカバリジョブには指定することはできません。
- 先行ジョブの終了コードによる起動条件を指定したジョブには指定することはできません。

・ 待ち合わせジョブ以外を先行ジョブにして指定することはできません。

・ 指定する先行ジョブは他ジョブの先行ジョブとして設定することはできません。

なお、先行ジョブに指定したジョブは、事前に定義されている必要があります。本指定を省略した場合は、先頭ジョブとなります。

(17) pre_job_endcode 先行ジョブの終了コードによる起動条件

先行ジョブの終了コードにより、後続ジョブの起動条件を切り分けたい場合に指定します。比較条件の場合と範囲条件の場合とで記述形式が異なります。

なお、先行ジョブが正常終了する条件と矛盾した実行条件を指定した場合はエラーとなります。指定する条件以上の値が先行ジョブの正常終了上限値に設定されているか、または先行ジョブに“終了コードに関わらず正常”が指定されていることを確認してください。

比較条件の場合

記述形式を以下に示します。

```
pre_job_endcode 判定用演算子,比較値
```

判定用演算子は、以下の6種類です。なお、先行ジョブの終了コードと比較する比較値には0から255までの整数値を設定します。

EQ(EQual):

“比較値”と等しい

NE(Not Equal):

“比較値”と等しくない

GT(Greater Than):

“比較値”より大きい

GE(Greater than Equal to):

“比較値”以上

LT(Less Than):

“比較値”より小さい

LE(Less than Equal to):

“比較値”以下

範囲条件の場合

```
pre_job_endcode 判定用演算子1,比較値1,判定用演算子2,比較値2
```

“比較値1”および“比較値2”には、0から255までの整数値を指定し、“比較値1”には“比較値2”より小さい数値を指定します。

“判定用演算子1”には、以下のいずれかを指定します。

GT(Greater Than):

“比較値1”より大きい

GE(Greater than Equal to):

“比較値1”以上

“判定用演算子2”には、以下のいずれかを指定します。

LT(Less Than):

“比較値2”より小さい

LE(Less than Equal to):

“比較値2”以下

(18) execdaymode 起動日のみ有効

ジョブ(ジョブネット)の“起動日のみ有効”を指定します。省略した場合は、“起動日のみ有効”は設定されません。

ON

“起動日のみ有効”を指定します。

(19) execuser 実行ユーザ名

ジョブを実行するユーザ名を64バイト以内で指定します。

アイコン種別でバッチフレームワークジョブ“BJOB”を指定する場合、指定できません。

(20) resource リソース名

利用する資源(リソース)名を64バイト以内で指定します。半角英数字、“_”(半角アンダーバー)、“-”(半角ハイフン)を利用することが可能です。省略した場合は、資源の割り当ては行いません。

アイコン種別でバッチフレームワークジョブ“BJOB”を指定する場合、指定できません。

(21) resource_attr リソース割り当て

指定した資源(リソース)の割り当て方法(SH:共用,EX:排他)を指定します。省略した場合は、SHが指定されたことになります。

アイコン種別でバッチフレームワークジョブ“BJOB”を指定する場合、指定できません。

(22) stdoutfile 標準出力ファイル名

ジョブの標準出力の出力先ファイル名を256バイト以内のフルパスで指定します。

アイコン種別でバッチフレームワークジョブ“BJOB”を指定する場合、指定できません。

(23) stderrfile 標準エラー出力ファイル名

ジョブの標準エラー出力の出力先ファイル名を256バイト以内のフルパスで指定します。

アイコン種別でバッチフレームワークジョブ“BJOB”を指定する場合、指定できません。

(24) errtostdfile 標準エラー出力を標準出力ファイルに出力

ジョブの標準エラー出力を標準出力ファイルに出力する場合に“ON”を指定します。

アイコン種別でバッチフレームワークジョブ“BJOB”を指定する場合、指定できません。

(25) jobfile_trans 実行コマンドの転送

ジョブとして起動する実行ファイルを依頼ホストに転送する場合、“ON”を指定します。省略した場合は、依頼ホスト上に存在する実行ファイルを起動します。

アイコン種別でバッチフレームワークジョブ“BJOB”を指定する場合、指定できません。

(26) priority 実行優先度

接続先のサーバがWindowsの場合、ジョブの実行優先度を4(最高)から0(最低)の範囲で指定します。

接続先のサーバがUNIXの場合、ジョブの実行優先度を39(最高)から0(最低)の範囲で指定します。

省略した場合、優先度は0となります。

アイコン種別でバッチフレームワークジョブ“BJOB”を指定する場合、ジョブキュー内実行優先順位を63(最高)から0(最低)の範囲で指定します。省略可能です。省略した場合、実行優先順位を32としてジョブを実行します。

(27) predictiontime 実行予測時間

ジョブの実行予測時間を指定します。指定できる範囲は、1～35791394(分)です。本オペランドを省略した場合、実行予測時間は設定されません。

なお、本オペランドは、実行属性が“ジョブ実行制御”であるジョブネット内のジョブに対してのみ有効となります。

(28) limittime 打ち切り時間

ジョブ実行経過時間の制限値(分)を1から1666666の範囲で指定します。

実行予測時間より小さい値を指定することはできません。

(29) netvar_prefix ジョブネット変数のプレフィックス

前回履歴の“NAME=VALUE”形式の行をジョブネット変数に取り込む場合に、ジョブネット変数のプレフィックスを13バイト以内で指定します。

半角英数字のみ使用できます。省略した場合は、ジョブネット変数に取り込まれません。

(30) envname 環境変数名

ジョブの実行時の環境変数を指定する場合の環境変数名を64バイト以内で指定します。英数字とアンダーバーのみ指定可能です。

アイコン種別でバッチフレームワークジョブ“BJOB”を指定する場合、バッチジョブ定義の置換えパラメタ名を64バイト以内で指定します。英小文字(a-z)、英大文字(A-Z)、数字(0-9)、半角ハイフン(-)および半角アンダーバー(_)のみ使用できます。また、1文字目は英小文字(a-z)、英大文字(A-Z)にしてください。

環境変数は32個まで指定可能です。

(31) envvalue 環境変数値

直前のenvnameオペランドで指定した環境変数名に対する環境変数値を256バイト以内で指定します。半角カナおよびユーザ定義文字を含むことはできません。

アイコン種別でバッチフレームワークジョブ“BJOB”を指定する場合、バッチジョブ定義の置換えパラメタ値を256バイト以内で指定します。半角カナおよびユーザ定義文字を含むことはできません。

環境変数の指定例

【Windows版】

```
envname TEMP_PATH
envvalue d:¥temp
envname DATA_PATH
envvalue d:¥gyoumu1¥data
```

【UNIX版】

```
envname LD_LIBRARY_PATH
envvalue /usr/openwin/lib
envname OPENWINHOME
envvalue /usr/openwin
```

(32) joblogfile ジョブログファイル名

ジョブログファイル名を256バイト以内のフルパスで指定します。空白、半角カナおよびユーザ定義文字を含むことはできません。

(33) stepname 開始ステップ名

開始ステップ名を64バイト以内で指定します。英小文字(a-z)、英大文字(A-Z)、数字(0-9)、半角ハイフン(-)および半角アンダーバー(_)のみ使用できます。また、1文字目は英小文字(a-z)、英大文字(A-Z)にしてください。省略可能です。

(34) codelaststep 最後に実行されたジョブステップの終了コードをジョブの終了コードとする

“最後に実行されたジョブステップの終了コードをジョブの終了コードとする”を指定します。最後に実行されたジョブステップの終了コードをジョブの終了コードとする場合に“ON”を指定します。

省略した場合には、バッチフレームワークのジョブで定義されている終了コード（終了コードの最大値または最後に実行されたジョブの終了コード）が、ジョブの終了コードとなります。

(35) operate 操作

操作による停止または無効の状態を表します。省略値は“NONE”です。

PAUSE

停止

DISABLE

無効

NONE

登録時、変更時ともに実行待ちになります。

(36) jobicon アイコン種別

登録するアイコン種別をパラメタで指定します。アイコン種別とパラメタの対応表、および記述例を以下に示します。本パラメタを省略した場合、コマンド実行“CMND”が登録されます。Interstage属性の場合、省略はできません。

アイコン種別	パラメタ
コマンド実行	CMND
メッセージ事象発生	MESG
サービス開始	SSTR
サービス停止	SEND
ファイル待ち合わせ	FWAT
バックアップ連携	BKUP
クライアントアプリケーション起動	EXAC
サーバアプリケーション起動	EXAS
クライアントファイル圧縮／伸長	FPRC
サーバファイル圧縮／伸長	FPRS
クライアントファイル操作	FCLC
サーバファイル操作	FCLS
クライアントファイル転送	FCPC
サーバファイル転送	FCPS
クライアント電源投入	PONC
クライアント電源切断	POFC
リモートアクセス制御【Windows版】	RASC
SAP ERP	SPR3
SAPクラウドサービス連携【Windows版】【Linux版】	SCSJ
Interstage	INTS
バッチフレームワークジョブ【EE】	BJOB
Oracle EBS コンカレント	OEBS

アイコン種別	パラメタ
リモートアクセス制御(接続)【Windows版】	RASS
リモートアクセス制御(切断)【Windows版】	RASE
クライアントファイル操作(ファイル名変更)	RNFC
サーバファイル操作(ファイル名変更)	RNFS
クライアントファイル操作(ファイル削除)	DLFC
サーバファイル操作(ファイル削除)	DLFS
クライアントファイル操作(ディレクトリ作成)	MKDC
サーバファイル操作(ディレクトリ作成)	MKDS
クライアントファイル操作(ディレクトリ削除)	RMDC
サーバファイル操作(ディレクトリ削除)	RMDS
ログファイル解析	CHKL
時刻待ち合わせ	WATT
サービス状態確認	CHKS
ジョブネット	JNET
リンクジョブネット	LNET
メッセージ事象待ち合わせ	CMSG
指定時間待ち合わせ	SLEP
REST実行【Windows版】【Linux版】	REST
ファイル存在判定	JGFL
ジョブネット変数判定	JGVR
MSP【GEE】	MMSP
XSP【GEE】	MXSP
ユーザ定義1(～ユーザ定義100)	U001(～U100)

記述例

```
jobicon CMND
```

(37) iconposition アイコン位置

ジョブアイコンの位置情報を以下の形式で表します。数字はジョブフローの左上を基準（1,1)とした、グリッド位置を表しています。

横位置(1～255), 縦位置(1～255)

省略した場合、アイコン位置は自動的に配置されます。省略して登録した後、[グループの変更]ウィンドウで変更を行った場合、その時点でのアイコン位置情報が保持され、jobschprint -rコマンドで本制御文が出力されます。

(38) workunitname ワークユニット名

Interstageに依頼するジョブ(ワークユニット名)を指定します。36バイト以内で指定します。

(39) systemname システム名

システム名を8バイト以内で指定します。

(40) systemname 業務システム名【UNIX版】

依頼先業務システム名を7バイト以内で指定します。

GEE **(41) job_definition 業務定義**

SymfoNETに業務定義してあるかどうかを指定します。業務定義がされている場合は“ON”を、まだ定義されていない場合は“OFF”を指定します。

GEE **(42) job_kind 業務種別定義**

起動する業務種別を以下のいずれかから指定します。XSPの場合で、かつSymfoNETに業務定義されていない場合に指定できます。省略した場合は、“BATCH”が指定されたものとし、バッチ業務を実行します。

BATCH

バッチ業務を起動します。

JCM

JCM(ジョブ制御マクロ)業務を起動します。

GEE **(43) jobid 業務ID/ホストのジョブ名**

グローバルサーバ上のジョブ名です。SymfoNETに定義済みの場合は業務ID、未定義の場合はホストのジョブ名を8バイト以内で指定します。先頭に“@”を使用したジョブ名は指定できません。

GEE **(44) user_definition ユーザ定義名**

グローバルサーバと連携するため、グローバルサーバ上のユーザ名と対応して登録した、UNIXシステム上のグループ名です。8バイト以内で指定します。

GEE **(45) filename 実行ファイル名/JCM名/データセット名**

業務定義が未定義の場合に指定します。

OS種別がXSPの場合は実行するファイル名またはJCM名を、MSPの場合はデータセット名を64バイト以内で指定します。業務定義済みの場合は指定できません。

GEE **(46) option ジョブへのパラメタ**

ジョブに渡すパラメタを指定します。

- 64バイト以内で指定します。
- 半角英数字で指定します。
- 以下の文字を指定することはできません。

  ¥, $, &, %, (,)

MSPの場合

指定した値は、グローバルサーバ上のEXPAND文のパラメタに置き換えられて実行されます。このとき、グローバルサーバ上のJCL内には、対応するEXPAND文が記述されている必要があります。

XSPの場合

指定した値は、グローバルサーバ上のJCM業務のパラメタに置き換えられて実行されます。

GEE **(47) exec_host ジョブ実行ホスト名**

ジョブを実行するグローバルサーバのホスト名です。16バイト以内で指定します。

GEE **(48) alternate_host 代替ホスト名**

ジョブ実行ホスト名に指定したホストに接続できない場合に、代わりに接続するホスト名を指定します。16バイト以内で指定します。

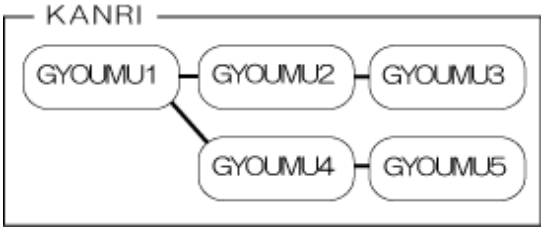
GEE

(49) retry リトライ間隔

グローバルサーバに接続できなかった時にリトライをする間隔を指定します。秒単位で5から30までの数値を指定します。リトライ回数は10回で固定されており、本オペランドで指定した間隔でリトライを10回行います。

ジョブ制御文指定例

下図のようなジョブ構成を持つジョブネットを定義する場合の制御文を図の下に示します。




```
NET
jobnetname KANRI
jobnetcomment 業務管理
starttime 0850
execpattern 毎月20日
holidaycalendar SYSTEM_CALENDAR     ←カレンダとして「SYSTEM_CALENDAR」を
holidayshift BEF                            利用します
job 5
;
JOB
jobname GYOUMU1.bat
jobnumber 1
;
JOB
jobname GYOUMU2.bat                 ← GYOUM2のジョブの定義
jobnumber 2                         ← GYOUM2のジョブ番号
pre_job 1                           ← GYOUM2の先行ジョブのジョブ番号
;                                     （先行ジョブ番号1のジョブはすでに、
JOB                                                 GYOUM1として定義してある。）
jobname GYOUMU3.bat
jobnumber 3
pre_job 2
;
JOB
jobname GYOUMU4.bat
jobnumber 4
pre_job 1
;
JOB
jobname GYOUMU5.bat
jobnumber 5
pre_job 4
;
```

4.12.3　起動日制御文

ジョブネットの起動日を設定する制御文です。

記述形式

```
SCH                                 ---------- (1)
jobnetname ジョブネット名           ---------- (2)  (*)  (注1)
execpattern 起動日雛形名            ---------- (3)  (**)
execpatmode 起動日雛形組み合わせ条件 ---------- (4)       (注2)
basenet 基準ジョブネット名          ---------- (5)  (**)
```

```
execday 起動日設定                          ---------- (6)
cancelday 起動日設定解除                    ---------- (7)
jobschprintr 起動日ビットパターン           ---------- (8)
annuallydate 毎年の日付指定                 ---------- (9)  (**)
annuallyweek 毎年の曜日指定                 ---------- (10) (**)
monthlydate 毎月の日付指定                  ---------- (11) (**)
monthlyweek 毎月の曜日指定                  ---------- (12) (**)
bnsmode 営業日の選択基準                    ---------- (13) (**)  (注5)
bnsbase 営業日の月初め日                    ---------- (14) (**)  (注6)
bnsday 営業日                               ---------- (15) (**)
bnslimit 月度内の営業日のみを起動日とする   ---------- (16) (**)
offsetmode オフセット                       ---------- (17)
offsetday オフセット日数                    ---------- (18)
offsetholiday 基準日が休日の場合は稼働日とする ----- (19)
holidayshift 休日シフト                     ---------- (20) (*)
shiftlimit シフト上限日数                   ---------- (21)       (注4)
holidaycalendar カレンダ名                  ---------- (22)
holidaymode カレンダ組み合わせ条件          ---------- (23)       (注3)
enablestart 有効期限開始日                  ---------- (24)
enableend 有効期限終了日                    ---------- (24)
disablestart 無効期限開始日                 ---------- (25)
disableend 無効期限終了日                   ---------- (25)
;
```

注1)

jobnetnameオペランドは、SCHキーワードのすぐ後に指定してください。

注2)

execpatmodeオペランドは、execpatternオペランドのすぐ後ろに指定してください。

注3)

holidaymodeオペランドは、holidaycalendarオペランドのすぐ後ろに指定してください。

注4)

shiftlimitオペランドは、holidayshiftオペランドのすぐ後ろに指定してください。

注5)

bnsmodeオペランドはoffsetmode,offsetday,offsetholidayの各オペランドとは同時には指定できません。

注6)

bnsmodeオペランドを指定する場合には必ず指定してください。

(*)：

必ず指定してください。

(**)：

以下のどれか1つを指定してください。なお、省略することもできます。

- execpattern
- basenet
- annuallydateとannuallyweekの組合せ(どちらかまたは両方)
- monthlydateとmonthlyweekの組合せ(どちらかまたは両方)
- bnsmode、bnsbase、bnsday、bnslimit(bnsmodeおよびbnsbaseはこのオペランド群を指定するときは必須)

制御文の説明

(1) SCH

起動日制御文の開始を表すキーワードです。

(2) jobnetname ジョブネット名

起動日を設定するジョブネット名を指定します。

“メッセージ事象制御文”と“一時変更制御文”を同時に指定する場合、同じジョブネット名を指定してください。

(3) execpattern 起動日雛形名

起動日雛形を利用する場合に、登録済みの起動日雛形名を指定します。

(4) execpatmode 起動日雛形組み合わせ条件

複数の起動日雛形名を指定する時の組み合わせ条件を以下に示すパラメタから選択して指定します。省略した場合は、“AND”が指定されたことになります。

AND

すべての起動日雛形で起動日設定されている日のみを起動日とします。

OR

少なくとも1つの起動日雛形で起動日設定されている日を起動日とします。

(5) basenet 基準ジョブネット名

基準ジョブネットを利用する場合に、基準ジョブネットとして登録されているジョブネット名を指定します。すでに自分自身で基準ジョブネットであると宣言している場合には、指定することはできません。

(6) execday 起動日設定

ジョブネットの起動予定日を設定します。YYYY年MM月DD日を起動日として設定する場合は、“YYYY,MM,DD”と指定します。複数の起動日を設定することができます。また、起動日雛形と組み合わせて設定すると、簡単に起動日を設定することができます。

起動日設定指定例

2010年10月1日および10日を起動日として設定する場合

```
execday 2010,10,01,10
```

2010年10月1日および12月10日を起動日として設定する場合

```
execday 2010,10,01
execday 2010,12,10
```

(7) cancelday 起動日設定解除

起動を解除したい日付を指定します。“起動日設定”と同じ形式で指定します。

(8) jobschprintr 起動日ビットパターン

jobschprint -rコマンドにより展開される起動日設定のビットパターンデータを指定します。(jobschprintコマンドにより生成された値以外が指定された場合の動作は保証されません。また同一バージョンのjobschprintコマンドによる展開結果以外の動作は保証されません。jobschprintコマンドにより展開した年号と違う年に本パラメタを指定することはできません。)ビットパターンデータを指定した場合、3年間はjobschprint -rコマンドにより展開したジョブネットの起動日と同一の起動日を設定することができます。なお、本オペランドを指定した直後は、他の起動日設定用オペランド(毎月カレンダ、毎年カレンダ、基準ジョブネット名、起動日雛形、カレンダ名、休日シフト、オフセット、有効期間/無効期間など)による起動日の再計算は実施されません。また、他ジョブネットの起動日には反映されません。

(9) annuallydate 毎年の日付指定

毎年カレンダを利用する場合に、毎年のジョブネットを起動する日付を指定します。MM月DD日を起動日として設定する場合は、“MMDD”と指定します。複数の起動日を設定することができます。

(10) annuallyweek 毎年の曜日指定

毎年カレンダを利用する場合に、ジョブネットを起動する曜日を月ごとに指定します。1つのオペランドで1つの月しか設定できません。複数の月を設定する場合は、本オペランドを複数回指定します。指定する形式は、何月かを表す数値を1～12の範囲で指定し、その後、曜日を表す英単語の先頭3文字を半角文字で指定します。“12,SUN,MON,TUE,WED,THU,FRI,SAT”のような形式で指定します。

(11) monthlydate 毎月の日付指定

毎月カレンダを利用する場合に、毎月のジョブネットを起動する日付を指定します。毎月DD日を起動日として設定する場合は、“DD”と指定します。複数の起動日を設定することができます。なお、毎月月末に起動する場合は“LAST”と指定します。

(12) monthlyweek 毎月の曜日指定

毎月カレンダを利用する場合に、毎月のジョブネットを起動する曜日を週ごとに指定します。1つのオペランドの指定で1つの週しか設定できません。複数の週を設定する場合は、本オペランドを複数回指定します。指定する形式は、何週目かを表す数値を0～6(第1～第5まで。0は毎週の指定。6は月の最終週)の範囲で指定し、その後、曜日を表す英単語の先頭3文字を半角文字で指定します。“5,SUN,MON,TUE,WED,THU,FRI,SAT”のような形式で指定します。

(13) bnsmode 営業日の選択基準

基準日情報“営業日”の指定の場合に、月初め、月締めのどちらを基準とするかを指定します。

START

営業日を月初めから数えます。

END

営業日を月締めから逆算します。

(14) bnsbase 営業日の月初め日

基準日情報“営業日”指定の場合に、月初め日を指定します。“0”は指定できません。

1～28

1日から28日です。

LAST

月末です。

(15) bnsday 営業日

基準日情報“営業日”指定の場合に、営業日を指定します。複数指定した場合には、“,”(コンマ)で区切って指定します。“0”は指定できません。

1～31

第1営業日（月初め）から数えた営業日です。bnsmodeオペランドで“END”を指定した場合は指定できません。

-1～-31

第-1営業日（月締め）から逆算した営業日です。bnsmodeオペランドで“START”を指定した場合は指定できません。

(16) bnslimit 月度内の営業日のみを起動日とする

基準日情報“営業日”指定の場合に、“月度内の営業日のみを起動日とする”の有無（ON/OFF）を指定します。bnsmodeオペランドが指定されている場合の省略値は“ON”です。

ON

“月度内の営業日のみを起動日とする”を指定します。

OFF

“月度内の営業日のみを起動日とする”を指定しません。

(17) offsetmode オフセット

基準日から移動する日数を数える方法を以下に示すパラメタから選択して指定します。省略した場合は、“EXECDAY”が指定されたことになります。

EXECDAY

カレンダの稼働日で移動する日数を数えます。

DAY

休日には関係なく、絶対日数で移動する日数を数えます。

(18) offsetday オフセット日数

基準日から移動する日数を“-99”から“99”までの間で指定します。省略した場合は“0”となります。

(19) offsetholiday 基準日が休日の場合は稼働日とする

基準日から移動する日数を数える時、基準日が休日であった場合の扱いを指定します。省略した場合は、“OFF”が指定されたことになります。

オフセットに“DAY”を指定した場合には、本オペランドを指定することはできません。

ON

基準日が休日の場合も稼働日とし、移動する日数の基準日とします。

OFF

基準日が休日の場合は稼働日とせず、移動する日数の基準日としません。

(20) holidayshift 休日シフト

起動日の登録および起動日雛形で設定される起動日が、休日と重なった場合の対処方法を以下に示すパラメタから選択して指定します。

BEF

ジョブネットの起動日を前日に変更します。

AFT

ジョブネットの起動日を翌日に変更します。

NOT

当日を休日として認識し、ジョブネットを起動しません。

IGN

当日を起動日として認識し、ジョブネットを起動します。

(21) shiftlimit シフト上限日数

休日シフトを行う時に休日が連続した時に起動を抑止する日数“0”から“99”までの間で指定します。省略した場合は、“0”となります。

(22) holidaycalendar カレンダ名

起動日を生成するときに元となる、カレンダ名(「システムカレンダ」および「システムカレンダ」以外のカレンダ)を指定します。1つのジョブネットについて5個までカレンダ名を指定することができます。複数のカレンダを指定する場合には、“,”で区切って指定します。

(23) holidaymode カレンダ組み合わせ条件

複数のカレンダ名を指定する時の組み合わせ方を以下に示すパラメタから選択して指定します。省略した場合は、“AND”が指定されたことになります。

AND

すべてのカレンダで休日設定されている日のみを休日とします。

OR

少なくとも1つのカレンダで休日設定されている日を休日とします。

(24) enablestart 有効期限開始日/ enableend 有効期限終了日

ジョブネットを運用する期間を“YYYY,MM,DD”の形式で指定します。有効期限開始日を省略した場合、即日から有効期限終了日まで有効となります。有効期限終了日を省略した場合、有効期限開始日以降、有効となります。有効期限開始日、有効期限終了日共に省略した場合は、すべての日が有効となります。

(25) disablestart 無効期限開始日/ disableend 無効期限終了日

ジョブネットを運用しない期間を“YYYY,MM,DD”の形式で指定します。無効期限開始日を省略した場合、即日から無効期限終了日まで無効となります。無効期限終了日を省略した場合、無効期限開始日以降、無効となります。無効期限開始日、無効期限終了日共に省略した場合は、すべての日が有効となります。

4.12.4 メッセージ事象制御文

メッセージ事象を設定する制御文です。

記述形式

```
MSG                          ---------- (1)
jobnetname  ジョブネット名   ---------- (2) (*) (注1)
messagemode メッセージ条件   ---------- (3)
carryover   持ち越し処理     ---------- (4)     (注2)
messagename メッセージ事象名 ---------- (5)
;
```

注1)

jobnetnameオペランドは、MSGキーワードのすぐ後に指定してください。

注2)

carryoverオペランドは、messagemodeオペランドのすぐ後ろに指定してください。

(*)：

必ず指定してください。

制御文の説明

(1) MSG

メッセージ事象制御文の開始を表すキーワードです。

(2) jobnetname ジョブネット名

メッセージ事象を設定するジョブネット名を指定します。

“起動日制御文”と“一時変更制御文”を同時に指定する場合、同じジョブネット名を指定してください。

(3) messagemode メッセージ条件

メッセージ事象の待ち合わせ条件(AND/OR)、起動条件(ORD/COG/BOT/UNC)、および起動時刻との条件(ANDTIME/ORTIME)を組み合わせて指定します。messagemodeオペランドを省略した場合、パラメタとして“AND,UNC,ORTIME”が指定されたことになります。「起動時刻との条件」として“ANDTIME”を指定した場合、または、ジョブネットがグループに含まれる場合は、「起動条件」として“ジョブネット実行中も有効”(“ORD”および“BOT”)は指定できません。ジョブネットがグループに含まれる場合、「起動時刻との条件」で指定した値は無視されます。

AND

登録してあるメッセージ事象がすべて発生した場合にジョブネットを起動します。

OR

登録してあるメッセージ事象のうちどれか1つでも発生した場合に、ジョブネットを起動します。

ORD

“ジョブネット実行中も有効”です。ジョブネットが実行中に発生したメッセージ事象が有効となります。その場合、ジョブネットが終了次第、実行中にメッセージ事象が発生した回数分、再びジョブネットが起動されます。ORDパラメタを指定しない場合、ジョブネットが実行中に発生したメッセージ事象は無効となります。

COG

“起動日のみ有効”です。ジョブネットの起動予定日に登録したメッセージ事象が発生した場合、ジョブネットを起動します。

BOT

“ジョブネット実行中も有効&起動日のみ有効”です。ジョブネットの起動予定日に登録したメッセージ事象が発生した場合、メッセージ事象が発生した回数分ジョブネットを起動します。

UNC

起動条件を設定しない場合に指定します。

ANDTIME

起動予定時刻が到来し、待ち合わせていたメッセージ事象が発生した時にジョブネットが起動されます。ジョブネットの起動条件が“単一起動”(一時変更期間中は“単一起動(一時変更)”)の場合にのみ有効になります。

ORTIME

起動予定時刻の到来または、待ち合わせていたメッセージ事象の発生のどちらか一方の起動条件が満たされればジョブネットが起動されます。

(4) carryover 持ち越し処理

日変わり時刻が到来したときの持ち越し処理の指定をします。「起動時刻の到来を待つ」(messagemodeオペランドに“ANDTIME”が指定されている)場合、または起動条件が「メッセージ事象発生時のみ起動」(ジョブネット制御文のmsgonlyオペランドに“ON”が指定されている)の場合に指定できます。省略値は“RESET”です。

RESET

日変わり時刻の到来時に、ジョブネットの持ち越し処理を行いません。

SET

日変わり時刻の到来時に、ジョブネットの持ち越し処理を行います。

(5) messagename メッセージ事象名

ジョブネットを起動する契機となるメッセージ事象を12バイト以内で指定します。“,”、“:”、空白、半角カナ、およびユーザ定義文字を含むことはできません。CSVファイルでメッセージ事象名を扱う機能(jobschdeftocsvコマンド, jobschcsvtodefコマンド, Job Designer、インポート・エクスポート機能)を使用する場合は、“;”を含むこともできません。70個まで指定できます。

メッセージ事象指定例

MSG1,MSG2,MSG3の3個のメッセージ事象を登録する場合

```
messagename MSG1,MSG2,MSG3
```

4.12.5　一時変更制御文

ジョブネットの一時変更情報を設定する制御文です。

記述形式

```
TMP                                    ---------- (1)
jobnetname ジョブネット名                ---------- (2) (*) (注)
tstarttime 単一起動（一時変更）          ---------- (3) (**)
tmultitime 複数起動（一時変更）          ---------- (4) (**)
tintervalstart 一定間隔（一時変更）      ---------- (5) (**)
tiplstart サーバ起動時（一時変更）          ---------- (6) (**)
tmsgonly メッセージ起動（一時変更）      ---------- (7) (**)
tpowermode 電源未投入時の処置（一時変更） --------- (8)
tiplmode サーバ起動時の処置（一時変更）       ---------- (9)
tstoptime 停止時刻（一時変更）           ---------- (10)
talarmtime 終了予定時刻（一時変更）      ---------- (11)
temporarystart 一時変更開始日            ---------- (12) (*)
temporaryend 一時変更終了日              ---------- (12)
tmsgonlymode メッセージ起動時の処置(一時変更) ---- (13)
tmsgstarttime 開始予定時刻(一時変更)     ---------- (14)
;
```

注1)

jobnetnameオペランドは、TMPキーワードのすぐ後に指定してください。

注2)

tmsgstarttimeオペランドは、tmsgonlymodeオペランドに“ONCE”を指定したときに指定できます。tmsgonlymodeのすぐ後ろに指定してください。

(*)：

必ず指定してください。

(**)：

どれか1つを必ず選択してください。

制御文の説明

(1) TMP

一時変更制御文の開始を表すキーワードです。

(2) jobnetname ジョブネット名

一時変更情報を設定するジョブネット名を指定します。

“起動日制御文”と“メッセージ事象制御文”を同時に指定する場合、同じジョブネット名を指定してください。

(3) tstarttime 単一起動(一時変更)

ジョブネットを1日に1回起動する場合に、起動時刻を4バイトで指定します。

一時変更開始日から一時変更終了日までの期間が有効になります。

(4) tmultitime 複数起動(一時変更)

ジョブネットを1日に複数回起動する場合に、起動時刻を4バイト以内で指定します。起動時刻は6個まで指定することができます。一時変更開始日から一時変更終了日までの期間が有効になります。起動時刻を1個しか指定しなかった場合、単一起動(一時変更)として登録されます。複数起動(一時変更)は、グループに含まれるジョブネットでは指定できません。

(5) tintervalstart 一定間隔(一時変更)

ジョブネットを一定の間隔で起動する場合に、起動時刻、終了時刻、および起動間隔を指定します。起動間隔の指定範囲は1～240(分)です。日変わり時刻をはさんで、起動時刻と終了時刻を指定することはできません。一時変更開始日から一時変更終了日までの期間が有効になります。一定間隔(一時変更)はグループに含まれるジョブネットでは指定できません。

(6) tiplstart サーバ起動時(一時変更)

ジョブスケジューラサービス/デーモンの起動時にジョブネットを起動する場合に、“ON”を指定します。一時変更開始日から一時変更終了日までの期間が有効になります。サーバ起動時(一時変更)はグループに含まれるジョブネットでは指定できません。

(7) tmsgonly メッセージ起動(一時変更)

特定のイベント事象が発生したときだけジョブネットを起動する場合に、“ON”を指定します。一時変更開始日から一時変更終了日までの期間が有効になります。メッセージ起動(一時変更)はグループに含まれるジョブネットでは指定できません。

(8) tpowermode 電源未投入時の処置(一時変更)

ジョブネット起動予定時刻にジョブスケジューラサービス/デーモンが起動されていなかった時の処置を指定します。省略した場合は、“RESET”となります。一時変更開始日から一時変更終了日までの期間が有効になります。ジョブネットの一時変更の起動条件として「単一起動(一時変更)」以外が指定されている場合、および、ジョブネットがグループに含まれている場合、「電源未投入時の処置(一時変更)」の指定値は無視されます。

RESET

ジョブスケジューラサービス/デーモンが起動されてもジョブネットは起動しません。

SET

ジョブスケジューラサービス/デーモン起動時にジョブネットを起動します。

(9) tiplmode サーバ起動時の処置(一時変更)

サーバ起動時のジョブネットをジョブスケジューラサービス/デーモン起動時ごとに起動するか否かを指定します。省略した場合は、“EVERYTIME”となります。一時変更開始日から一時変更終了日までの期間が有効になります。

EVERYTIME

ジョブスケジューラサービス/デーモンが起動されるごとにジョブネット起動します。

ONCE

1日に複数回ジョブスケジューラサービス/デーモンが起動されてもジョブネットは最初の1回のみ起動します。

(10) tstoptime 停止時刻(一時変更)

ジョブネットの「実行属性」が“INTS”の場合に、停止する時刻を指定します。停止時刻は、HHMM(時分)の形式で4バイトで指定します。省略した場合、停止時刻は設定されません。一時変更後の停止時刻に現在時刻より前の時刻が指定された場合、停止予定日時は翌日に設定されます。なお、本指定は、一時変更開始日から一時変更終了日までの期間が有効になります。

(11) talarmtime 終了予定時刻(一時変更)

ジョブネットの終了予定時刻(一時変更)を指定します。

HHMM(時分)の形式で4バイトで指定します。省略した場合、終了予定時刻は設定されません。日変わり時刻をはさんで、起動時刻と終了予定時刻を指定することはできません。なお、本指定は、一時変更開始日から一時変更終了日までの期間が有効になります。

(12) temporarystart 一時変更開始日/ temporaryend 一時変更終了日

ジョブネットを一時変更する期間を“YYYY,MM,DD”の形式で指定します。一時変更開始日のみを指定した場合は、指定した当日のみが一時変更の対象となります。

(13) tmsgonlymode メッセージ起動時の処置(一時変更)

待ち合わせているメッセージ事象が1日複数回発生する場合に、メッセージ事象が発生するたびにジョブネットを起動するかどうかを指定します。省略した場合は“EVERYTIME”となります。本指定は、一時変更開始日から一時変更終了日までの期間が有効になります。なお、ジョブネットの起動条件(一時変更)が「メッセージ事象発生時のみ起動」以外の時は本オペランドは無視されます。

EVERYTIME

待ち合わせているメッセージ事象が発生するたびにジョブネットが起動されます。

ONCE

待ち合わせているメッセージ事象が1日複数回発生しても、ジョブネットは最初の1回のみ起動されます。

(14) tmsgstarttime 開始予定時刻（一時変更）

メッセージ起動する場合の、ジョブネットの開始予定時刻を指定します。「メッセージ起動時の処置」(一時変更)が“ONCE”の場合のみ指定可能です。

HHMM(時分)の形式で4バイトで指定します。省略した場合、開始予定時刻は設定されません。本指定は、一時変更開始日から一時変更終了日までの期間が有効になります。なお、ジョブネットの起動条件(一時変更)が「メッセージ事象発生時のみ起動」以外の時は本オペランドは無視されます。

4.13 jobschsetgrpコマンド

記述形式

```
[登録の場合]
  jobschsetgrp -ent [プロジェクト名] 定義ファイル [-sys サブシステム番号] (注)
[変更の場合]
  jobschsetgrp -che [プロジェクト名] 定義ファイル [-sys サブシステム番号] (注)
[削除の場合]
  jobschsetgrp -del [プロジェクト名/]グループ名 [-sys サブシステム番号]  (注)
```

注)

“[-sys サブシステム番号]”はSystemwalker Operation Manager EEのオプションです。

機能説明

グループを登録/変更/削除します。

オプション

-ent

グループを定義ファイルの情報をもとに登録します。なお、同一グループ名がすでに存在する場合は登録できません。

プロジェクト名

グループの情報を格納する格納先のプロジェクト名を指定します。登録時の場合、jobschsetgrpコマンドの実行ユーザが更新アクセスできるプロジェクトのうち、ジョブネットが登録されているプロジェクトが1つしかない場合は、省略できます。変更/削除の場合、jobschsetgrpコマンドの実行ユーザが更新アクセスできるプロジェクトのうちグループが登録されているプロジェクトが1つしかない場合は、省略できます。

定義ファイル

グループ制御文およびグループ内ジョブネット制御文を格納したファイル名を指定します。

複数のグループを登録するときは、グループ制御文を複数記述します。

“グループ制御文”については、“4.13.1 グループ制御文”を参照してください。

“グループ内ジョブネット制御文”については、“4.13.2 グループ内ジョブネット制御文”を参照してください。

-che

グループを定義ファイルの情報をもとに変更します。グループを変更する場合、グループを登録する時に使用した定義ファイルを利用すると便利です。

-del

グループの削除を行います。グループの削除を行った場合、グループおよびグループ内ジョブネットの登録情報がすべて削除されます。

グループ名

削除するグループ名を指定します。

EE GEE

-sys サブシステム番号

複数サブシステム運用をしているシステムで、操作対象のサブシステムを指定するためのオプションです。操作対象とするサブシステム番号を、0から9の範囲で指定します。このオプションを省略した場合、サブシステム番号0が操作対象となります。

復帰値

復帰値	意味
0	正常終了
0以外	異常終了。エラーの内容はエラーメッセージを確認してください。

コマンド格納場所

Windows	Systemwalker Operation Managerインストール先ディレクトリ¥MpWalker.JM¥bin
Solaris HP-UX AIX Linux	/usr/bin

注意事項

- 本コマンドは、グループの変更をサーバに依頼するだけであり、本コマンドの実行完了が、グループの変更処理の完了を保証するものではありません(本コマンドの実行完了後、グループの変更処理が完了するまでに、しばらく時間がかかる場合があります)。

4.13.1　グループ制御文

グループの登録情報を設定する制御文です。

記述形式

```
GRP                              ---------- (1)
groupname グループ名              ---------- (2)  (*)
groupcomment グループ名称         ---------- (3)  (*)
powermode 電源未投入時の処置      ---------- (4)
jobnet グループ内ジョブネット数   ---------- (5)  (*)
operate 操作                     ---------- (6)
;
```

(*)：

必ず指定してください。

制御文の説明

(1) GRP

グループ制御文の開始を示すキーワードです。

(2) groupname グループ名

グループ名を50バイト以内で指定します。他のグループ名と重複はできません。

また、以下の記号、空白、半角カナ、およびユーザ定義文字を含むことはできません。
【Windows版】 /?*;,¥"<>|:
【UNIX版】 /?*;,

出口プログラムを利用する場合、以下の文字は指定できません。【Windows版】

& = ^ %

(3) groupcomment グループ名称

グループへのコメントを100バイト以内で指定します。空白、半角カナ、およびユーザ定義文字を含むことはできません。

(4) powermode 電源未投入時の処置

グループ起動予定時刻にジョブスケジューラサービス/デーモンが起動されていなかった場合の処置を指定します。省略した場合は、“RESET”となります。

RESET

ジョブスケジューラサービス/デーモンが起動されてもグループは起動されません。

SET

ジョブスケジューラサービス/デーモン起動時にグループが起動されます。

(5) jobnet グループ内ジョブネット数

グループを構成するジョブネットの数(数字)を指定します。

Systemwalker Operation Manager SEは、1～50まで指定できます。

EE GEE Systemwalker Operation Manager EEは、1～255まで指定できます。

(6) operate 操作

操作による停止または無効の状態を表します。省略値は“NONE”です。

PAUSE

停止

DISABLE

無効

NONE

登録時、変更時ともに実行待ちになります。

4.13.2　グループ内ジョブネット制御文

グループ内のジョブネットの登録情報を設定する制御文です。

記述形式

```
GNT                              ---------- (1)
jobnetname ジョブネット名        ---------- (2)  (*)  (注1)
jobnetnumber ジョブネット番号    ---------- (3)  (*)  (注2)
pre_jobnet 先行ジョブネット番号  ---------- (4)
starttimeignore 開始予定時刻無視 ---------- (5)
iconposition アイコン位置        ---------- (6)       (注3)
;
```

注1)

jobnetnameオペランドは、GNTキーワードのすぐ後に指定してください。

注2)

jobnetnumberオペランドは、jobnetnameオペランドのすぐ後に指定してください。

注3)

グループを構成するすべてのジョブネットに指定する必要があります。

(*)：

必ず指定してください。

制御文の説明

(1) GNT

グループ内ジョブネット制御文の開始を示すキーワードです。

(2) jobnetname ジョブネット名

すでに登録されているジョブネット名を50バイト以内で指定します。

終了処理ジョブネット名の“JSHEND”または“JSHFORCE”はグループ内に登録できません。【Windows版】

また、以下の記号、空白、半角カナ、およびユーザ定義文字を含むことはできません。
【Windows版】 /?*;,¥"<>|:'.
【UNIX版】 /?*;,'.

(3) jobnetnumber ジョブネット番号

ジョブネット名に対応する番号を割り当てます。ジョブネットの実行順序に従い、重複しないように番号を割り当てます。先行ジョブネットから後続ジョブネットへ昇順になるように番号を割り当てます。

Systemwalker Operation Manager SEは、1～50まで指定できます。

EE GEE

Systemwalker Operation Manager EEは、1～255まで指定できます。

(4) pre_jobnet 先行ジョブネット番号

先行ジョブネットのジョブネット番号を指定します。先行ジョブネット番号を複数指定する場合は、コンマ“,”で区切ります。指定が多くなる場合は、昇順に並べておくと効率よく確認できます。なお、先行ジョブネットに指定したジョブネットは、事前に定義されている必要があります。

(5) starttimeignore 開始予定時刻無視

ジョブネットの開始予定時刻を無視して実行させる場合、“ON”を指定します。

(6) iconposition アイコン位置

ジョブアイコンの位置情報を以下の形式で表します。数字はジョブフローの左上を基準（1,1)とした、グリッド位置を表しています。

横位置(1～255), 縦位置(1～255)

省略した場合、アイコン位置は自動的に配置されます。省略して登録した後、[グループの変更]ウィンドウで変更を行った場合、その時点でのアイコン位置情報が保持され、jobschprint -Rコマンドで本制御文が出力されます。

注意事項

グループには、以下の条件を満たすジョブネットを登録します。

- グループを登録するプロジェクトに属している。
- [起動条件]が[時刻起動]で、登録してある起動時刻が1つだけ(一日一回起動)である。
- メッセージ事象の条件として[ジョブネット実行中も有効]が指定されていない。
- グループ内に同一のジョブネットが存在しない。
- 他のグループに登録されていない。

4.14 jobschctljobコマンド

記述形式

jobschctljob [プロジェクト名/]ジョブネット名/ジョブ名 *control* [-sys サブシステム番号]（注）

注)

“[-sys サブシステム番号]”はSystemwalker Operation Manager EEのオプションです。

機能説明

ジョブを制御（強制終了／再起動／停止／停止解除／無効／無効解除)します。

オプション

プロジェクト名

制御の対象となるジョブの格納先のプロジェクト名を指定します。jobschctljobコマンドの実行ユーザが操作できるプロジェクトのうちジョブが登録されているプロジェクトが1つしかない場合は、省略できます。

ジョブネット名

制御の対象となるジョブの格納先のジョブネット名を指定します。

ジョブ名

制御の対象となるジョブ名を指定します。

control

以下に示す6つの制御項目のうちどれか1つを指定します。

制御の対象となるジョブの状態と操作の関係、および、親ジョブネットの状態と操作の関係については、“Systemwalker Operation Manager 運用ガイド”の“スケジュールジョブを操作する”を参照してください。

cancel

ジョブを強制終了します。

rerun

異常終了した、または、強制終了させたジョブを再起動します。

pause

ジョブを停止中の状態にします。

continue

ジョブを“停止”を行う前の状態にします。

disable

ジョブを無効状態にします。

enable

ジョブを“無効”を行う前の状態にします。

EE GEE **-sys サブシステム番号**

複数サブシステム運用をしているシステムで、操作対象のサブシステムを指定するためのオプションです。操作対象とするサブシステム番号を、0から9の範囲で指定します。このオプションを省略した場合、サブシステム番号0が操作対象となります。

復帰値

復帰値	意味
0	正常終了
0以外	異常終了。エラーの内容はエラーメッセージを確認してください。

コマンド格納場所

Windows	Systemwalker Operation Managerインストール先ディレクトリ¥MpWalker.JM¥bin
Solaris HP-UX AIX Linux	/usr/bin

注意事項

- ジョブの制御は、実行属性が“ジョブ実行制御”、および“PowerAIM”【UNIX版】のジョブネットに属しているジョブにしかできません。
- 制御を行うジョブには、“ジョブ名”が必ず指定されている必要があります。ジョブ名が省略されているジョブは、制御できません。
- 子ジョブネットの制御はできません。
- リンクジョブネットの制御はできません。
- ジョブネット内に指定されたジョブ名のジョブが複数ある場合、本コマンドは異常終了します。

・ 本コマンドは、ジョブの操作をサーバに依頼するものであり、コマンドの実行が完了しても、サーバ側でジョブの操作が完了したことを保証するものではありません。このため、コマンドの実行完了後に、jobschprint -pコマンドを入力して、ジョブの状態を確認してください。

4.15 jobschcontrolコマンド

記述形式

jobschcontrol [プロジェクト名/]ジョブネット名 *control*
[start用オプション] [rerun用オプション] [-sys サブシステム番号] (注)

注)

“ [-sys サブシステム番号]”はSystemwalker Operation Manager EEのオプションです。

機能説明

ジョブネットを制御(強制終了/起動/再起動/停止/停止解除/無効/無効解除/確認/終了取消/持越解除)します。

オプション

プロジェクト名

ジョブネット情報の格納先のプロジェクト名を指定します。jobschcontrolコマンドの実行ユーザが操作できるプロジェクトのうちジョブネットが登録されているプロジェクトが1つしかない場合は、省略できます。

ジョブネット名

制御の対象となるジョブネット名を指定します。

control

以下に示す10の制御項目のうちどれか1つを指定します。

cancel

ジョブネットを強制終了します。

start

ジョブネットを起動します。

rerun

前回、異常終了した、強制終了させた、または確認済にしたジョブネットを再起動します。

なお、子ジョブネットの再起動については、親ジョブネットの状態が実行中状態または警告状態の場合のみ可能です。

pause

ジョブネットを停止中の状態にします。

continue

ジョブネットを“停止”を行う前の状態にします。

disable

ジョブネットを無効状態にします。

enable

ジョブネットを“無効”を行う前の状態にします。

cope

異常終了したジョブネットを確認済状態にします。

[ジョブスケジューラ起動パラメタの定義]ウィンドウ－[利用機能１]シートの[異常時の確認操作]の[詳細設定]ボタンを押下して表示される[確認操作の詳細設定]ウィンドウで[強制終了を確認操作の対象とする]を指定した場合には、強制終了も制御の対象とします。

reinstate

ジョブネットの終了取消を行います。

revoke

ジョブネットの持越解除を行います。

rerun用オプション

controlでrerunを指定した場合のみ、以下に示す2つの制御項目が指定できます。rerun用オプションを省略した場合は、異常終了したジョブから再起動します。異常終了したジョブより前のジョブに再起動ポイントが設定されている場合には、そのジョブから起動します。

next

異常終了したジョブ(子ジョブネット)に対して、“次の再起動のみ無効”を設定してからジョブネットを再起動します(異常終了したジョブ(子ジョブネット)を飛ばして、次のジョブから起動します)。強制終了したジョブ(子ジョブネット)についても、異常終了したジョブと同様に扱います。

top

先頭ジョブからジョブネットを再起動します。

start用オプション

controlでstartを指定した場合のみ指定できます。

-p param

起動されるジョブネットの各ジョブに渡す可変パラメタを指定します。本オプションを指定すると、ジョブネットの起動時に、ジョブ定義中に記述された変数“@.VPARAM@”が、指定した文字列に置き換えられます。

可変パラメタは200バイト以内の文字列で指定します。半角カナおよびユーザ定義文字を含むことはできません。

本オプションは省略可能です。省略した場合、ジョブ定義中に記述された変数“@.VPARAM@”は空文字列に置き換えられてジョブが起動されます。

可変パラメタは、階層化されたジョブネットの2階層目以降のジョブネット(子ジョブネット)にも渡されます。

本オプションを複数指定することはできません。

なお、変数“@.VPARAM@”については、“Systemwalker Operation Manager 運用ガイド”の“可変パラメタを利用してジョブネットを起動する場合”を参照してください。

EE GEE **-sys サブシステム番号**

複数サブシステム運用をしているシステムで、操作対象のサブシステムを指定するためのオプションです。操作対象とするサブシステム番号を、0から9の範囲で指定します。このオプションを省略した場合、サブシステム番号0が操作対象となります。

復帰値

復帰値	意味
0	正常終了
0以外	異常終了。エラーの内容はエラーメッセージを確認してください。

コマンド格納場所

Windows	Systemwalker Operation Managerインストール先ディレクトリ¥MpWalker.JM¥bin

Solaris HP-UX AIX Linux	/usr/bin

注意事項

- 本コマンドは、ジョブネットの操作をサーバに依頼するものであり、コマンドの実行が完了しても、サーバ側で、ジョブネットの操作が完了したことを保証するものではありません。このため、コマンドの実行完了後に、jobschprint -nコマンドを入力して、ジョブネットの状態を確認してください。
- start用オプションで-p paramを指定する場合、以下の点に注意してください。
  - 指定する可変パラメタには、起動するジョブネットのジョブ定義中にある変数“@.VPARAM@”が置き換えられたときに、その定義項目の使用不可文字が含まれない、かつ文字数の上限値を超えない文字列を指定してください。置き換え後の結果、ジョブ定義が不正になった場合は、ジョブは実行されず、終了コード245または246で異常終了します。
- マスタリンクジョブネット、および、その階層下のジョブネットは、停止/停止解除/無効/無効解除の制御しかできません。
- rerun用オプションを指定する場合、以下の点に注意してください。
  - 子ジョブネットに対してrerun用オプションを指定した場合、rerun用オプションは無視されて、異常終了したジョブから再起動されます。
  - 異常終了したジョブより前のジョブに再起動ポイントが設定されている場合には、再起動ポイントが設定されたジョブから再起動されます。
- jobschprintコマンドによりジョブネットの終了状態を確認したにも関わらず、本コマンドで以下のメッセージが出力された場合は、ジョブネットの状態が完全に終了状態に遷移するまでのわずかなタイミングにより発生したものなので、しばらく待ってから再度実行してください。

  「ジョブネットは既に実行中または警告状態です」

  「ジョブネットは既に実行中状態または警告状態です」

  「ジョブネットは異常終了状態または強制終了状態または確認済状態でないため再起動できません」

  「本日実行済みでないため終了取消できません」

4.16 jobschctlgrpコマンド

記述形式

```
jobschctlgrp [プロジェクト名/]グループ名 control
        [-sys サブシステム番号]        (注)
```

注)

“ [-sys サブシステム番号]”はSystemwalker Operation Manager EEのオプションです。

機能説明

グループを制御(強制終了/起動/再起動/停止/停止解除/無効/無効解除/確認)します。

オプション

プロジェクト名

グループ情報の格納先のプロジェクト名を指定します。jobschctlgrpコマンドの実行ユーザが操作できるプロジェクトのうちグループが登録されているプロジェクトが1つしかない場合は、省略できます。

グループ名

制御の対象となるグループ名を指定します。

control

以下に示す8つの制御項目のうちどれか1つを指定します。

cancel

グループを強制終了します。

start

グループを起動します。

rerun

前回、異常終了した、強制終了させた、または確認済にしたグループを再起動します。

pause

グループを停止中の状態にします。

continue

グループを“停止”を行う前の状態にします。

disable

グループを無効状態にします。

enable

グループを“無効”を行う前の状態にします。

cope

異常終了したグループを確認済状態にします。

[ジョブスケジューラ起動パラメタの定義]ウィンドウ－[利用機能１]シートの[異常時の確認操作]の[詳細設定]ボタンを押下して表示される[確認操作の詳細設定]ウィンドウで[強制終了を確認操作の対象とする]を指定した場合には、強制終了も制御の対象とします。

EE GEE

-sys サブシステム番号

複数サブシステム運用をしているシステムで、操作対象のサブシステムを指定するためのオプションです。操作対象とするサブシステム番号を、0から9の範囲で指定します。このオプションを省略した場合、サブシステム番号0が操作対象となります。

復帰値

復帰値	意味
0	正常終了
0以外	異常終了。エラーの内容はエラーメッセージを確認してください。

コマンド格納場所

Windows	Systemwalker Operation Managerインストール先ディレクトリ¥MpWalker.JM¥bin
Solaris HP-UX AIX Linux	/usr/bin

注意事項

- 本コマンド入力後は、jobschprint -xコマンドを入力して、ジョブネットの状態を確認してください。
- 本コマンドは、グループの操作をサーバに依頼するだけのものです。コマンドの実行が完了しても、サーバ側で、グループへの操作が完了していることを保証するものではありません。
  そのため、本コマンドの完了後、しばらく経過してから、操作依頼された処理が実施される場合があります。

4.17 jobschoperateコマンド

記述形式

jobschoperate *control* -prj [プロジェクト名] [-sys サブシステム番号] （注）

注)

“ [-sys サブシステム番号]”はSystemwalker Operation Manager EEのオプションです。

機能説明

プロジェクト内のジョブネットおよびグループを一括制御(すべて停止/すべて停止解除/すべて無効/すべて無効解除)します。

ただし、以下のジョブネットおよびグループは、停止/停止解除/無効/無効解除されず、操作の対象外となります。

- グループ内のジョブネット
- 子ジョブネット/リンクジョブネット（第2～5階層のジョブネット）
- 複写先ジョブネット

オプション

control

以下に示す4つの制御項目のうちどれか1つを指定します。

pause

すべてのジョブネットおよびグループを停止中の状態にします。

continue

すべてのジョブネットおよびグループの停止状態を解除します。

disable

すべてのジョブネットおよびグループを無効状態にします。

enable

すべてのジョブネットおよびグループの無効状態を解除します。

-prj プロジェクト名

指定されたプロジェクト内のジョブネットおよびグループを一括制御します。

ジョブネット/グループを一括操作する対象のプロジェクト名を指定します。jobschoperate コマンドの実行ユーザが操作できるプロジェクトのうち、ジョブネットが登録されているプロジェクトが1つしかない場合は、省略できます。

EE GEE **-sys サブシステム番号**

複数サブシステム運用をしているシステムで、操作対象のサブシステムを指定するためのオプションです。操作対象とするサブシステム番号を、0から9の範囲で指定します。このオプションを省略した場合、サブシステム番号0が操作対象となります。

ジョブネット/グループの状態と操作の関係

操作対象の状態と操作可否の関係を以下に示します。

・ ジョブネット

		操作			
		停止	停止解除	無効	無効解除
ジョブネットの状態	正常終了	○	△	○	△
	疑似正常	○	△	○	△
	異常終了	○	△	○	△
	強制終了	○	△	○	△
	警告	×	△	×	△
	実行中/終了遅延	×	△	×	△
	実行待ち/開始遅延	○	△	○	△
	停止中	△	○	○	△
	無効状態	×	△	△	○
	確認済	○	△	○	△
	持ち越し	○	△	○	△

○：操作できます
×：操作できません（WARNINGが出力されます）
△：無視されます

・ グループ

		操作			
		停止	停止解除	無効	無効解除
グループの状態	正常終了	○	△	○	△
	疑似正常	○	△	○	△
	異常終了	○	△	○	△
	強制終了	○	△	○	△
	警告	×	△	×	△
	実行中	×	△	×	△
	実行待ち	○	△	○	△
	停止中	△	○	○	△
	無効状態	×	△	△	○
	確認済	○	△	○	△

○：操作できます
×：操作できません（WARNINGが出力されます）
△：無視されます

復帰値

復帰値	意味
0	操作対象すべてに操作できました。

復帰値	意味
	ジョブネット/グループの状態と操作の関係で、一括操作の対象が「○：操作できます」または「△：無視されます」に該当し、「○：操作できます」の操作対象に対して、すべて操作されたことを示します。
1	操作対象の中に操作できないものがありました。 ジョブネット/グループの状態と操作の関係で、下記に該当することを示します。 ・ 一括操作の対象に「×：操作できません」に該当するものが一つでもある または、 ・ 一括操作の対象のうち、「○：操作できます」に該当するにもかかわらず、変更要求中または操作要求中のために操作できなかった操作対象が一つでもある 処理は中断されず、制御できるジョブネット/グループはすべて処理されます。 複数のジョブネットおよびグループが制御できなかった場合においても、メッセージは1つだけ出力されます。
2	エラーが発生しました。処理は中断されます。

コマンド格納場所

Windows	Systemwalker Operation Managerインストール先ディレクトリ¥MpWalker.JM¥bin
Solaris Linux	/usr/bin

注意事項

- 本コマンドは、ジョブネットの操作をサーバに依頼するだけのものです。コマンドの実行が完了しても、サーバ側で、ジョブネットへの操作が完了していることを保証するものではありません。
  そのため、本コマンドの完了後、しばらく経過してから、操作依頼された処理が実施される場合があります。
- 本コマンド入力後は、「jobschprint -n」および「jobschprint -x」コマンドを入力して、ジョブネットとグループの状態を確認してください。
- 操作対象のジョブネット/グループのうち、一つでも操作に成功した場合、監査ログの実行結果はSUCCESS(成功)になります。

実行結果／出力形式

復帰値0 : メッセージは出力されません。

復帰値1,2 : エラー内容は標準エラー出力に出力されます。

使用例

サブシステム1のプロジェクト“prj1”に対し、一括停止する場合の例

```
jobschoperate pause -prj prj1 -sys 1
```

4.18　jobschmoveコマンド

記述形式

```
jobschmove -group [プロジェクト名/]グループ名 [移動先グループ名]
           [-sys サブシステム番号]                (注)
jobschmove -gnet [プロジェクト名/]グループ名 [移動先ジョブネット名]
```

```
                [-sys サブシステム番号]                (注)
jobschmove -net [プロジェクト名/]ジョブネット名 [移動先ジョブネット名]
                [-sys サブシステム番号]                (注)
```

注)

“ [-sys サブシステム番号]”はSystemwalker Operation Manager EEのオプションです。

機能説明

指定したグループおよびジョブネットの登録順序を変更します。プロジェクト名は、jobschmoveコマンドの実行ユーザが更新できるプロジェクトのうちジョブネットが登録されているプロジェクトが1つしかない場合は、省略できます。

オプション

-group [プロジェクト名/] グループ名 [移動先グループ名]

指定したグループの登録順序を変更します。

グループ名に指定されたグループを、移動先グループに指定されたグループの直後へ移動します。移動先グループを省略した場合は、先頭へ移動します。

-gnet [プロジェクト名/] グループ名 [移動先ジョブネット名]

指定したグループに属するジョブネットの登録順序を他のジョブネットの直後に変更します。

グループ名に指定されたグループの属するジョブネットを、移動先ジョブネットに指定されたジョブネットの直後へ移動します。移動先ジョブネットを省略した場合は、先頭へ移動します。移動の際、グループ内のジョブネットの順序は、先行ジョブネットが先になります。

-net [プロジェクト名/] ジョブネット名 [移動先ジョブネット名]

指定したジョブネットの登録順序を変更します。ジョブネット名に指定されたジョブネットを、移動先ジョブネットに指定されたジョブネットの直後へ移動します。移動先ジョブネットを省略した場合は、先頭へ移動します。

EE GEE

-sys サブシステム番号

複数サブシステム運用をしているシステムで、操作対象のサブシステムを指定するためのオプションです。操作対象とするサブシステム番号を、0から9の範囲で指定します。このオプションを省略した場合、サブシステム番号0が操作対象となります。

復帰値

復帰値	意味
0	正常終了
0以外	異常終了。エラーの内容はエラーメッセージを確認してください。

コマンド格納場所

Windows	Systemwalker Operation Managerインストール先ディレクトリ¥MpWalker.JM¥bin
Solaris HP-UX AIX Linux	/usr/bin

4.19 jobschmsgclearコマンド

記述形式

```
jobschmsgclear -all [プロジェクト名] [-sys サブシステム番号]　(注)
jobschmsgclear -net [プロジェクト名/]ジョブネット名
                [-sys サブシステム番号]                (注)
```

注)

“ [-sys サブシステム番号]”はSystemwalker Operation Manager EEのオプションです。

機能説明

発生済のメッセージ事象を無効にする場合に使用します。日付が変わったなどで、改めてすべてのメッセージ事象の発生を待ち合わせしたい場合に、jobschmsgclearコマンドを使用し、メッセージ事象発生済みの状態を初期化してください。

なお、発生したメッセージ事象のカウントやクリアのされ方については、“Systemwalker Operation Manager 運用ガイド”の“メッセージ事象とジョブネットの動作”を参照してください。

オプション

-all

コマンド起動ユーザの更新できるすべてのプロジェクト内に存在するすべてのジョブネットについて、発生済みのメッセージ事象を無効にします。

プロジェクト名

-allと同時に指定することで、指定されたプロジェクト内のすべてのジョブネットについて発生済のメッセージ事象を無効にします。

-net

指定したジョブネットについて、発生済みのメッセージ事象を無効にします。

プロジェクト名

発生したメッセージ事象を無効にするジョブネットが存在するプロジェクト名を指定します。なお、jobschmsgclearコマンドの実行ユーザが更新できるプロジェクトのうちジョブネットが登録されているプロジェクトが1つしかない場合は、省略できます。

ジョブネット名

メッセージ事象を無効にするジョブネット名を指定します。-netパラメタと一緒に指定します。

EE GEE

-sys サブシステム番号

複数サブシステム運用をしているシステムで、操作対象のサブシステムを指定するためのオプションです。操作対象とするサブシステム番号を、0から9の範囲で指定します。このオプションを省略した場合、サブシステム番号0が操作対象となります。

復帰値

復帰値	意味
0	正常終了
0以外	異常終了。エラーの内容はエラーメッセージを確認してください。

コマンド格納場所

Windows	Systemwalker Operation Managerインストール先ディレクトリ¥MpWalker.JM¥bin

Solaris HP-UX AIX Linux	/usr/bin

注意事項

- 本コマンドによって発生済みのメッセージ事象をクリアしても、[メッセージ事象一覧]ウィンドウでは、発生した日付を表示したままとなります。
- 停止中状態にてメッセージ事象が発行され、起動条件が満たされたジョブネットは、本コマンドによって発生済みのメッセージ事象を初期化することはできません。

4.20 jobschmsgeventコマンド

記述形式

【Windows版】

```
jobschmsgevent メッセージ事象名[:プロジェクト名[/ジョブネット名]]
[{ホスト名｜-f:ファイル名} [-jm]] [-t:種別] [-p param [-c suffix]]
[-sys {all|サブシステム番号}] (注)
```

【UNIX版】

```
jobschmsgevent メッセージ事象名[:プロジェクト名[/ジョブネット名]
[メッセージテキスト]] [-h ホスト名] [-t:種別] [-p param [-c suffix]]
[-sys {all | サブシステム番号}] (注)
```

注)

“[-sys { all | サブシステム番号}]”はSystemwalker Operation Manager EEのオプションです。

機能説明

メッセージ事象を発生させます。

本コマンドでメッセージ事象を発生させた場合、イベントログにソース名“MpJobsch”、イベントID“2301”、メッセージテキスト“nnnnメッセージ事象が発生しました”(nnnnはメッセージ事象名)のメッセージ事象が出力されます。【Windows版】

本コマンドで発生させたメッセージ事象は、プロジェクト名を省略した場合、すべてのプロジェクトのすべてのジョブネットに対して有効となります。

ホスト名およびファイル名の両方を省略した場合、自ホストに対してメッセージ事象を発生させます。【Windows版】

なお、発生させたメッセージ事象のカウントやクリアのされ方については、“Systemwalker Operation Manager 運用ガイド”の“メッセージ事象とジョブネットの動作”を参照してください。

オプション

メッセージ事象名

発生させるメッセージ事象名を12バイト以内で指定します。

プロジェクト名

特定のプロジェクト内のジョブネットに対してのみメッセージ事象を発生させる場合に、プロジェクト名を指定します。

ジョブネット名

特定のジョブネットに対してのみメッセージ事象を発生させる場合に、対象となるジョブネットを指定します。

メッセージテキスト【UNIX版】

任意の文字列を指定します。

ホスト名【Windows版】 -h ホスト名【UNIX版】

メッセージ事象を発生させるホスト名を指定します。ホスト名は64バイト以内の英数字で指定します。省略した場合、自ホストに対してメッセージ事象を発生させます。

なお、他ホストでメッセージ事象を発生させたい場合は、本オプションを使用せずに、ネットワークジョブを利用してメッセージ事象を発生させることを推奨します。

-f: ファイル名【Windows版】

発生させるサーバのホスト名一覧を記述したファイル名を指定します。

-jm【Windows版】

本オプションを指定した場合は、ホスト名で指定したメッセージ事象を発生させるホスト(メッセージ事象発行先)とjmnet(デフォルト:9352/tcp)のポート番号で通信します。

本オプションの指定が必要な場合については、“Systemwalker Operation Manager トラブルシューティングガイド”の“エラーメッセージ「RegisterEventSource() 1783:スタブは正しくないデータを受信しました。」が出力される【Windows版】”を参照してください。

-t: 種別【Windows版】

発生させるメッセージ事象名のイベントログに出力される種別を指定します。

省略値は“-t:INFO”になります。

以下のどれか1つを指定します。

INFO

“情報”を出力します。

WARN

“警告”を出力します。

ERROR

“エラー”を出力します。

-p *param*

メッセージ事象により起動されるジョブネットの各ジョブに渡す可変パラメタを指定します。本オプションを指定すると、メッセージ事象を受信したジョブネットの起動時に、ジョブ定義中に記述された変数“@.VPARAM@”を、指定した可変パラメタに置き換えます。

可変パラメタは200バイト以内の文字列で指定します。半角カナおよびユーザ定義文字を含むことはできません。

可変パラメタは省略可能です。省略した場合、ジョブ定義中に記述された変数“@.VPARAM@”は空文字列に置き換えられてジョブが起動されます。

本オプションで指定した可変パラメタを受信できるのは、以下の条件をすべて満たすジョブネットです。

- グループに登録されていない
- ジョブ実行制御属性である
- 起動条件がメッセージ事象発生時のみ起動であり、かつ、以下のいずれかの条件を満たしている
  - 1つのメッセージ事象を待ち合わせている

－ 登録された複数のメッセージ事象のうち、いずれかのメッセージ事象を待ち合わせている

第1階層のジョブネットに渡された可変パラメタは、階層化されたジョブネットの2階層目以降のジョブネット(子ジョブネット)に引き継がれます。

2階層目以降のジョブネット/ジョブに直接可変パラメタは渡せません。ただし、jobschchkmsgevtコマンドの使用時は、ジョブネット変数(CM_VPARAM)から受け取ることができます。ジョブネット変数については、“Systemwalker Operation Manager 運用ガイド”の“ジョブネット変数をジョブ間で利用する”を参照してください。

本オプションを複数指定することはできません。

-c *suffix*

複写起動するジョブネット名につけるサフィックスを、30バイト以内の文字列で指定します。本オプションを指定すると、可変パラメタを指定して起動される対象のジョブネットが、指定したサフィックスつきのジョブネット名に複写され、複写されたジョブネットが起動されます。階層化されたジョブネットの場合、2階層目以降のジョブネットも複写されます。

半角英数字およびハイフン(-)が使用できます。先頭文字は半角英数字のみ使用可能です。必ず-pオプションとともに指定し、-pオプションよりも後ろに指定します。

複写先ジョブネットのジョブネット名およびジョブネット名称は、“複写元のジョブネット名（またはジョブネット名称）＋アンダーバー(_)＋*suffix*で指定した文字列”となります。

複写は、ジョブネットが起動されるタイミング（階層化されたジョブネットの場合は、親ジョブネットが起動されるタイミング）で行われます。また、複写先ジョブネットは、次回、同じ名前のジョブネットが複写されるタイミングで上書きされます。

なお、複写起動されたジョブネットは自動削除されません。適宜削除してください。

グループ内ジョブネットに対しては、本オプションは無効となります。

EE GEE **-sys { all | サブシステム番号}**

複数サブシステム運用をしているシステムで、操作対象のサブシステムを指定するためのオプションです。このオプションを省略した場合、サブシステム番号0が操作対象となります。

ただし、サブシステム運用しているジョブスケジューラのジョブとして起動された場合は、起動元のサブシステム番号が操作対象となります。

all

すべてのサブシステムを操作対象とします。

サブシステム番号

操作対象とするサブシステム番号を、0から9の範囲で指定します。

復帰値

復帰値	意味
0	正常終了
0以外	異常終了。エラーの内容はエラーメッセージを確認してください。

コマンド格納場所

Windows	Systemwalker Operation Managerインストール先ディレクトリ¥MpWalker.JM¥bin
Solaris HP-UX AIX Linux	/usr/bin

注意事項

- コマンドを実行するユーザーIDが、メッセージ事象を発生させるホストにも必要です。
- メッセージ事象を待ち合わせているジョブネットが以下の状態の場合は、メッセージ事象が発生しても認識されません。
  - ー ジョブネットが実行中状態(注1)
  - ー ジョブネットが警告状態(注1)(注2)
  - ー ジョブネットが無効状態
  - ー ジョブネットが異常終了状態(注3)
  - ー ジョブネットが強制終了状態(注4)
  - ー メッセージ事象の累積の発生回数が255回を超えている場合(注5)
  - ー メッセージ事象を待ち合わせているジョブネットが非起動日の場合(注6)
  - ー [メッセージ事象発生時のみ起動]の[一日一回]が指定してあり、ジョブネットがすでに起動した場合(注7)

  注1)

  [ジョブネット実行中も有効]をチェックしている場合は認識されます。このとき、メッセージ事象がジョブネット実行中に発生した回数分、ジョブネットが繰り返し起動されます。なお、ジョブネットは同時には起動されません。

  注2)

  実行属性が“ジョブ実行制御”のジョブネットのみが警告状態になります。

  注3)

  グループに含まれるジョブネットの場合は認識されます。また、グループに含まれないジョブネットの場合でも、[確認操作の詳細設定]ウィンドウで[ジョブネットの確認操作を有効とする]をチェックしていない場合は認識されます。

  注4)

  グループに含まれるジョブネットの場合は認識されます。また、グループに含まれないジョブネットの場合でも、[確認操作の詳細設定]ウィンドウで[強制終了を確認操作の対象とする]をチェックしていない場合は認識されます。

  注5)

  [メッセージ事象発生時のみ起動]のジョブネットは、内部でメッセージ事象の発生回数を保持しています。この発生回数を累積の発生回数といいます。累積の発生回数は、メッセージ事象の発生により、1つ増加します。逆にメッセージ事象の発生により、ジョブネットが起動した場合、発生回数が1つ減少します。メッセージ事象の累積の発生回数が255を超えないよう注意してください。

  注6)

  [起動日のみ有効]をチェックしていない場合は認識されます。

  注7)

  [終了取消]操作後は再び認識されます。
- ホスト名一覧を記述する場合、1行に1つのホスト名を64バイト以内で指定します。【Windows版】
- ホスト名一覧を記述する場合、ホスト名は1カラム目から指定します。なお、ファイル中にコメントは記述できません。【Windows版】
- メッセージ事象を発生させるホストは、Windows版のSystemwalker Operation Managerサーバ機能がインストールされている必要があります。【Windows版】
- メッセージ事象を発生させるホストは、Windows版のServerサービスがあらかじめ起動されている必要があります。【Windows版】

- jobschmsgeventコマンドは指定されたホストのアプリケーションイベントログにイベントを記録する動作をします。このため、以下の点に注意が必要です。【Windows版】

  ー jobschmsgeventコマンドを実行するユーザーIDが、メッセージ事象を発生させるホストにも認証可能である必要があります。

  例えば、Systemwalker Operation Managerのジョブとして実行する場合は、"Systemwalker MpMjes"サービスのログオンアカウントが相手サーバにおいて認証可能であることを確認してください。

  ー 記録先のイベントログの容量がいっぱいとなっている場合、イベントの書き込みができずjobschmsgeventコマンドの実行が失敗する場合があります。

  運用を開始する前に、記録先のアプリケーションイベントログのプロパティで、ログサイズを十分に大きくするか、「必要に応じてイベントを上書きする」設定をし、イベントログの容量が不足しないようにしてください。

  イベントログの記録にはNetBIOSインタフェースが利用されるため、メッセージ連携するサーバ間にファイアウォールを構築し、NetBIOSインタフェースのポートを制限する場合、メッセージ事象の通知が正しく実施されなくなります。この場合はjobschmsgeventのホスト名指定の利用は取り止め、ネットワークジョブとして実行するようにしてください。

- -jmオプションを指定せずに、他ホストでメッセージ事象を発生させる場合は、Administratorユーザにて実行する必要があります。

  ネットワークジョブを利用すると、実行するユーザを意識する必要がなくなるため、ネットワークジョブを利用して、依頼先ホストでメッセージ事象を発生させることを推奨します。(ただし、実行するユーザがGuestsグループに所属するユーザの場合を除きます。下記の注意事項を参照してください。)【Windows版】

- 自ホストでメッセージ事象を発生させる場合は、jobschmsgeventコマンドを発行するユーザーIDがGuestsグループに所属する場合、メッセージ事象の発生に失敗する場合があります。Guestsグループ以外のユーザーIDでjobschmsgeventコマンドを発行してください。【Windows版】

- -hオプションで指定するホストには、UNIX版のSystemwalker Operation Managerサーバ機能がインストールされている必要があります。【UNIX版】

- EE版からSE版のSystemwalker Operation Managerに対してメッセージ事象を発生させる場合は、以下のスクリプトを実行してください。

  【Windows版】

  [jobschmsgevent.bat]

```
set JOBSCH_SUB_SYSTEM=
jobschmsgevent %*
setecode.exe %ERRORLEVEL%
```

  上記スクリプトを実行するためのジョブのコマンド名欄の例を示します。

```
ドライブ名:¥パス名¥jobschmsgevent.bat MsgA HostName
```

  【UNIX版】

  [jobschmsgevent.sh]

```
#!/bin/csh
unsetenv JOBSCH_SUB_SYSTEM
jobschmsgevent $*
exit $status
```

  上記スクリプトを実行するためのジョブのコマンド名欄の例を示します。

```
//パス名/jobschmsgevent.sh MsgA -h HostName
```

- -pオプションで可変パラメタを指定する場合、以下の点に注意してください。
  - 指定する可変パラメタには、起動するジョブネットのジョブ定義中にある変数“@.VPARAM@”が置き換えられたときに、その定義項目の使用不可文字が含まれない、かつ文字数の上限値を超えない文字列を指定してください。置き換え後の結果、ジョブ定義が不正になった場合は、ジョブは実行されず、終了コード245または246で異常終了します。
  - 可変パラメタを受け付けることができないジョブネットに対して、本オプションを指定してメッセージ事象を発生させないでください。
  - メッセージ事象を発生させるホストがV13.2.0以前の場合、本オプションは指定できません。
- -cオプションを指定してジョブネットを複写する場合、以下の点に注意してください。
  - 複写先のジョブネット名およびジョブネット名称には、アンダーバー(_)とサフィックスが付加されます。複写元のジョブネット名長、ジョブネット名称長、およびサフィックスの長さは、複写により作成されるジョブネット名長、ジョブネット名称長がそれぞれの上限値を超えない名前にしてください。上限値を超えた場合、ジョブネットの複写に失敗しジョブネットは起動されません。
  - EE版およびGEE版では、登録するジョブネット数の制限はありませんが、ジョブネットを大量に複写する場合は、“Systemwalker Operation Manager 運用ガイド”の“性能のチューニングについて”および“Systemwalker Operation Manager 解説書”の“参考情報(ノウハウなど)”を参考にして、スケジュールどおりに問題なくジョブネットが起動されるか、十分に性能検証を行ってから運用してください。
  - 複写先ジョブネットは自動削除されません。適宜削除してください。
- メッセージ事象を待ち合わせているジョブネット側で[起動日のみ有効]([ジョブネットのプロパティ]ウィンドウ−[メッセージ]シート)の指定がされている場合、jobschmsgeventコマンドの起動日の判定は、コマンド実行時のタイミングではなく、待ち合わせているジョブネットの起動日の判定処理を行っている際の現在時刻をもとに行われます。jobschmsgeventコマンド実行中に日変わり時刻が到来した際には、意図せずジョブネットが起動することがありますので、注意が必要です。

4.21 jobschcancelnetコマンド【Windows版】

記述形式

```
jobschcancelnet プロジェクト名 ジョブネット名 [-all]
          [-sys サブシステム番号]          (注)
```

注)

“[-sys サブシステム番号]”はSystemwalker Operation Manager EEのオプションです。

機能説明

ジョブネット起動キューにつながっているジョブネットの起動要求を削除します。ジョブネット起動APIを使用してジョブネットの起動要求を行った時に、すでに同一ジョブネットが実行中である場合、ジョブネットは、ジョブネット起動キューにつながります。ジョブネット起動キューにつながったジョブネットの起動要求を削除したい場合に、本コマンドを使用します。

本コマンドは、指定したジョブネットの起動要求を削除します。

オプション

プロジェクト名

起動キューから削除するジョブネットが所属するプロジェクト名を指定します。

ジョブネット名

削除するジョブネット名を指定します。

-all

指定されたジョブネットの起動要求を起動キューからすべて削除します。

本オプションを指定しない場合は、最初の起動要求を削除します。

EE **-sys サブシステム番号**

複数サブシステム運用をしているシステムで、操作対象のサブシステムを指定するためのオプションです。操作対象とするサブシステム番号を、0から9の範囲で指定します。このオプションを省略した場合、サブシステム番号0が操作対象となります。

ただし、サブシステム運用しているジョブスケジューラのジョブとして起動された場合は、起動元のサブシステム番号が操作対象となります。

復帰値

復帰値	意味
0	正常終了
0以外	異常終了。エラーの内容はエラーメッセージを確認してください。

コマンド格納場所

Windows	Systemwalker Operation Managerインストール先ディレクトリ¥MpWalker.JM¥bin

4.22 jobschdeftocsvコマンド

記述形式

```
jobschdeftocsv [-d] 制御文ファイル
```

機能説明

以下の制御文ファイルをCSV形式に変換します。変換結果は、標準出力に出力します。

- ジョブネット制御文およびジョブ制御文
- グループ制御文およびグループ内ジョブネット制御文
- 休日情報制御文
- 起動日雛形制御文

変換結果をファイルに保存すると、制御文情報を表計算ソフトで管理することができます。(注)

注)

表計算ソフトに取り込む前に、出力結果を表計算ソフトがサポートする文字コードに変換しておく必要があります。

オプション

-d

各項目をダブルクォーテーション(")で囲み、出力します。ダブルクォーテーション(")については、2つの連続するダブルクォーテーション("")に変換して出力します。

制御文にダブルクォーテーションを含む場合で、Job Designerに取り込むときに指定してください。また、表計算ソフトに取り込む場合にも、必要に応じて指定してください。

変換例（「a"b」の場合）

-dなしの場合： a"b

-dありの場合： "a""b"

制御文ファイル

変換元の制御文ファイルを指定します。以下の制御文を記述したファイルを指定します。

- ジョブネット制御文およびジョブ制御文
- グループ制御文およびグループ内ジョブネット制御文
- 休日情報制御文
- 起動日雛形制御文

本コマンドで指定する制御文ファイルは、jobschprint -r/-R/-rh/-rspコマンドの出力結果を使用するか、以下の例に従って作成してください。

◇ジョブネット制御文およびジョブ制御文の制御文ファイルの作成例

ジョブネット制御文およびジョブ制御文の制御文ファイルの例を以下に示します。各制御文の詳細は、“4.12.1 ジョブネット制御文”および“4.12.2 ジョブ制御文”を参照してください。

```
NET
jobnetname hanbai
jobnetcomment 販売管理
starttime 2000
execpattern 毎週金曜日
holidayshift IGN
defaulthost king
job 4
operate PAUSE
;
JOB
jobname kanri1
jobnumber 1
jobcomment 顧客照会
restartpoint ON
iconposition 1,1
operate NONE
;
JOB
jobname kanri2
jobnumber 2
jobcomment 入金明細
pre_job 1
iconposition 1,2
operate NONE
;
JOB
jobname kanri3
jobnumber 3
jobcomment データ検索
pre_job 1
iconposition 2,2
operate DISABLE
;
JOB
jobname kanri4
jobnumber 4
jobcomment バックアップ
pre_job 2,3
iconposition 1,3
```

```
operate NONE
;
```

◇グループ制御文およびグループ内ジョブネット制御文の制御文ファイルの作成例

グループ制御文およびグループ内ジョブネット制御文の制御文ファイルの例を以下に示します。各制御文の詳細は、“4.13.1 グループ制御文”および“4.13.2 グループ内ジョブネット制御文”を参照してください。

```
GRP
groupname shuuhou
groupcomment 週報
jobnet 2
;
GNT
jobnetname eigyou
jobnetnumber 1
;
GNT
jobnetname hanbai
jobnetnumber 2
pre_jobnet 1
;
```

◇休日情報制御文の制御文ファイルの作成例

休日情報制御文の制御文ファイルの例を以下に示します。各制御文の詳細は、“4.2.1 休日情報制御文”を参照してください。

```
HOL
calendarname 本社カレンダ
yearset 2010
monthset 11
dayset 1
yearset 2010
monthset 12
dayset 20
;
```

◇起動日雛形制御文の制御文ファイルの作成例

起動日雛形制御文の制御文ファイルの例を以下に示します。各制御文の詳細は、“4.3.1 起動日雛形制御文”を参照してください。

```
CAL
execpatname サンプル
monthset 1
dayset 1
monthset 2
dayset 10
;
```

復帰値

復帰値	意味
0	正常終了
0以外	異常終了。エラーの内容はエラーメッセージを確認してください。

コマンド格納場所

Windows	Systemwalker Operation Managerインストール先ディレクトリ¥MpWalker.JM¥bin
Solaris HP-UX AIX Linux	/usr/bin

注意事項

CSV形式のファイルを表計算ソフトで読み込む時、数字だけの文字列が以下のように変換される場合があります。

CSVファイル上での表記:0010 → 表計算ソフトでの表記:10

これは表計算ソフトが数字だけの文字列を、文字列ではなく数値データとして誤って認識するためにおこる現象です。このような場合は、セルの書式設定により該当セルが文字列として扱われるように変更したのち、CSVファイル上での表記にあわせて文字列を訂正してください。

出力例

各制御文ファイルの作成例で示した制御文ファイルを入力とした場合のコマンドの出力結果の例を以下に示します。

◇ジョブネット制御文情報およびジョブ制御文情報の出力例

```
NET
jobnetname,jobnetcomment,starttime,execpattern,holidayshift,defaulthost,job,operate
hanbai,販売管理,2000,毎週金曜日,IGN,king,4,PAUSE
JOB
jobname,jobnumber,jobcomment,restartpoint,pre_job,iconposition,operate
kanri1,1,顧客照会,ON,1;1,NONE
kanri2,2,入金明細,,1,1;2, NONE
kanri3,3,データ検索,,1,2;2,DISABLE
kanri4,4,バックアップ,,2;3,1;3,NONE
```

◇グループ制御文情報およびグループ内ジョブネット制御文情報の出力例

```
GRP
groupname,groupcomment,jobnet
shuuhou,週報,2
GNT
jobnetname,jobnetnumber,pre_jobnet
eigyou,1,
hanbai,2,1
```

◇休日情報制御文情報の出力例

```
HOL
calendarname,yearset,monthset,dayset,yearset,monthset,dayset
本社カレンダ,2010,11,1,2010,12,20
```

◇起動日雛形制御文情報の出力例

```
CAL
execpatname,monthset,dayset,monthset,dayset
サンプル,1,1,2,10
```

上記のファイルを表計算ソフトで表示すると、以下のように表示されます。

◇ジョブネット制御文情報およびジョブ制御文情報の表示例

NET							
jobnetname	jobnetcomment	starttime	execpatternrn	holidayshift	defaulthost	job	operate
hanbai	販売管理	2000	毎週金曜日	IGN	king	4	PAUSE
JOB							
jobname	jobnumber	jobcomment	restartpoint	pre_job	iconposition	operate	
kanri1	1	顧客照会	ON		1;1	NONE	
kanri2	2	入金明細		1	1;2	NONE	
kanri3	3	データ検索		1	2;2	DISABLE	
kanri4	4	バックアップ		2;3	1;3	NONE	

◇グループ制御文情報およびグループ内ジョブネット制御文情報の表示例

GRP		
groupname	groupcomment	jobnet
shuuhou	週報	2
GNT		
jobnetname	jobnetnumber	pre_jobnet
eigyou	1	
hanbai	2	1

◇休日情報制御文情報の表示例

HOL						
calendarname	yearset	monthset	dayset	yearset	monthset	dayset
本社カレンダ	2010	11	1	2010	12	20

◇起動日雛形制御文情報の表示例

CAL				
execpatname	monthset	dayset	monthset	dayset
サンプル	1	1	2	10

4.23 jobschcsvtodefコマンド

記述形式

```
jobschcsvtodef CSVファイル
```

機能説明

CSV形式のジョブネット、グループ、カレンダの休日および起動日雛形の登録情報を、制御文形式に変換します。変換結果は標準出力に出力します。

変換結果をファイルに保存すると、それぞれjobschsetnetコマンド、jobschsetgrpコマンド、jobschsetholコマンド、jobschsetcalコマンドに指定する定義ファイルとして使用できます。

オプション

CSVファイル

変換元のCSVファイルを指定します。以下のCSVファイルを指定します。

- ジョブネット定義CSVファイル

  ジョブネット制御文およびジョブ制御文を入力したCSVファイルです。

- グループ定義CSVファイル

  グループ制御文およびグループ内ジョブネット制御文を入力したCSVファイルです。

- カレンダの休日定義CSVファイル

  休日情報制御文を入力したCSVファイルです。

- 起動日雛形定義CSVファイル

  起動日雛形制御文を入力したCSVファイルです。

参考

ジョブネット定義CSVファイルは、Job Designerを利用すると、表計算ソフトを使って視覚的に編集することができます。詳細は、“Systemwalker Operation Manager Job Designer ユーザーズガイド”を参照してください。

各CSVファイルの記述形式を以下に示します。

◇ジョブネット定義CSVファイルの記述形式

ジョブネット定義CSVファイルは、以下の形式で作成します。

```
NET
制御文,制御文,・・・
オプションの値,オプションの値,・・・
JOB
制御文,制御文,・・・
オプションの値,オプションの値,・・・
オプションの値,オプションの値,・・・
        ・
```

1行目:

“NET”と入力します。“NET”は、ジョブネット制御文の開始を表すキーワードです。

2行目:

ジョブネット制御文のオプションを入力します。必須オプションは必ず入力してください。オプションの詳細については“4.12.1 ジョブネット制御文”を参照してください。

3行目:

2行目に入力したオプションの順番に合わせて、オプションの値を入力します。オプションの値は省略できます。

4行目:

“JOB”と入力します。“JOB”は、ジョブ制御文の開始を表すキーワードです。

5行目:

ジョブ制御文のオプションを入力します。必須オプションは必ず入力してください。オプションの詳細については“4.12.2 ジョブ制御文”を参照してください。

6行目:

5行目に入力したオプションの順番と合わせて、オプションの値を入力します。オプションの値は省略できます。

7行目以降:

記述形式は6行目と同じです。登録するジョブの数だけ行を追加してください。



注意

1つの制御文に対してオプションの値が複数ある場合は、区切り記号としてセミコロン(;)を使用してください。ただし、起動日雛形名が複数ある場合は、区切り記号として半角空白を使用してください。

◇ジョブネット定義CSVファイルの作成例

```
NET
jobnetname,jobnetcomment,starttime,execpattern,holidayshift,defaulthost,job,operate
hanbai,販売管理,2000,毎週金曜日,IGN,king,4,PAUSE
JOB
jobname,jobnumber,jobcomment,restartpoint,pre_job,iconposition,operate
kanri1,1,顧客照会,ON,1;1,NONE
kanri2,2,入金明細,,1,1;2, NONE
kanri3,3,データ検索,,1,2;2,DISABLE
kanri4,4,バックアップ,,2;3,1;3,NONE
```

◇グループ定義CSVファイルの記述形式

グループ定義CSVファイルは、以下の形式で作成します。

```
GRP
制御文,制御文,・・・
オプションの値,オプションの値,・・・
GNT
制御文,制御文,・・・
オプションの値,オプションの値,・・・
オプションの値,オプションの値,・・・
      ・
```

1行目:

“GRP”と入力します。“GRP”は、グループ制御文の開始を表すキーワードです。

2行目:

グループ制御文のオプションを入力します。必須オプションは必ず入力してください。オプションの詳細については“4.13.1 グループ制御文”を参照してください。

3行目:

2行目に入力したオプションの順番に合わせて、オプションの値を入力します。オプションの値は省略できます。

4行目:

“GNT”と入力します。“GNT”は、グループ内ジョブネット制御文の開始を表すキーワードです。

5行目:

グループ内ジョブネット制御文のオプションを入力します。必須オプションは必ず入力してください。オプションの詳細については“4.13.2 グループ内ジョブネット制御文”を参照してください。

6行目:

5行目に入力したオプションの順番と合わせて、オプションの値を入力します。オプションの値は省略できます。

7行目以降:

記述形式は6行目と同じです。登録するグループ内ジョブネットの数だけ行を追加してください。



注意

1つの制御文に対してオプションの値が複数ある場合は、区切り記号としてセミコロン(;)を使用してください。

◇グループ定義CSVファイルの作成例

```
GRP
groupname,groupcomment,jobnet
shuuhou,週報,2
GNT
jobnetname,jobnetnumber,pre_jobnet
eigyou,1,
hanbai,2,1
```

◇カレンダの休日定義CSVファイルの記述形式

カレンダの休日定義CSVファイルは、以下の形式で作成します。

```
HOL
制御文,制御文,・・・
オプションの値,オプションの値,・・・
```

1行目:

“HOL”と入力します。“HOL”は、休日情報制御文の開始を表すキーワードです。

2行目:

休日情報制御文のオプションを入力します。必須オプションは必ず入力してください。オプションの詳細については“休日情報制御文”を参照してください。

3行目:

2行目に入力したオプションの順番に合わせて、オプションの値を入力します。オプションの値は省略できます。



注意

1つの制御文に対してオプションの値が複数ある場合は、区切り記号としてセミコロン(;)を使用してください。

◇カレンダの休日定義CSVファイルの作成例

```
HOL
calendarname,yearset,monthset,dayset,yearset,monthset,dayset
本社カレンダ,2010,11,1,2010,12,20
```

◇起動日雛形定義CSVファイルの記述形式

起動日雛形定義CSVファイルは、以下の形式で作成します。

```
CAL
制御文,制御文,・・・
オプションの値,オプションの値,・・・
```

1行目:

“CAL”と入力します。“CAL”は、起動日雛形制御文の開始を表すキーワードです。

2行目:

起動日雛形制御文のオプションを入力します。必須オプションは必ず入力してください。オプションの詳細については“起動日雛形制御文”を参照してください。

3行目:

2行目に入力したオプションの順番に合わせて、オプションの値を入力します。オプションの値は省略できます。



注意

1つの制御文に対してオプションの値が複数ある場合は、区切り記号としてセミコロン(;)を使用してください。

◇起動日雛形定義CSVファイルの作成例

```
CAL
execpatname,monthset,dayset,monthset,dayset
サンプル,1,1,2,10
```

復帰値

復帰値	意味
0	正常終了
0以外	異常終了。エラーの内容はエラーメッセージを確認してください。

コマンド格納場所

Windows	Systemwalker Operation Managerインストール先ディレクトリ¥MpWalker.JM¥bin
Solaris HP-UX AIX Linux	/usr/bin

出力例

各CSVファイルの作成例で示したCSVファイルを入力とした場合のコマンドの出力結果の例を以下に示します。

◇ジョブネット制御文情報およびジョブ制御文情報の出力例

```
NET
jobnetname hanbai
jobnetcomment 販売管理
holidayshift IGN
execpattern 毎週金曜日
starttime 2000
defaulthost king
job 4
operate PAUSE
;
JOB
jobname kanri1
jobnumber 1
jobcomment 顧客照会
restartpoint ON
iconposition 1,1
operate NONE
;
JOB
jobname kanri2
jobnumber 2
jobcomment 入金明細
pre_job 1
iconposition 1,2
operate NONE
;
JOB
jobname kanri3
jobnumber 3
jobcomment データ検索
pre_job 1
iconposition 2,2
operate DISABLE
;
JOB
jobname kanri4
jobnumber 4
jobcomment バックアップ
pre_job 2,3
iconposition 1,3
operate NONE
;
```

◇グループ制御文情報およびグループ内ジョブネット制御文情報の出力例

```
GRP
groupname shuuhou
groupcomment 週報
jobnet 2
;
GNT
jobnetname eigyou
jobnetnumber 1
;
GNT
jobnetname hanbai
jobnetnumber 2
```

```
pre_jobnet 1
;
```

◇休日情報制御文情報の出力例

```
HOL
calendarname 本社カレンダ
yearset 2010
monthset 11
dayset 1
yearset 2010
monthset 12
dayset 20
;
```

◇起動日雛形制御文情報の出力例

```
CAL
execpatname サンプル
monthset 1
dayset 1
monthset 2
dayset 10
;
```

4.24 jobschchecktimeコマンド

記述形式

```
jobschchecktime -wt waittime [-lt maxwaittime] [-sys サブシステム番号]
```

注)

“[-sys サブシステム番号]”はSystemwalker Operation Manager EEのオプションです。

機能説明

オプションで指定した待ち合わせ時刻が来るまで待ちます。すでに待ち合わせ時刻を過ぎていた場合は、すぐに終了します。

オプションで最大待ち合わせ時刻を指定していた場合は、その時刻の前後で復帰値が変わります。

待ち合わせ時刻および最大待ち合わせ時刻は、日変わり時刻を基準にした時刻です。ネットワークジョブの場合は、実行サーバ側の日変わり時刻が採用されます。

サーバが動作していないときは使用できません。

オプション

-wt *waittime*

この時刻を待ち合わせ時刻にします。時と分を2桁ずつ指定します。0000から2359までの値が指定できます。日変わり時刻を考慮します。例えば、日変わり時刻が10時の場合は0900という指定は翌日の9時を表します。

-lt *maxwaiting*

この時刻を最大待ち合わせ時刻にします。時と分を2桁ずつ指定します。0000から2359までの値が指定できます。日変わり時刻を考慮します。例えば、日変わり時刻が10時の場合は0930という指定は翌日の9時30分を表します。待ち合わせ時刻より早くなるように指定した場合はエラーになります。

EE GEE

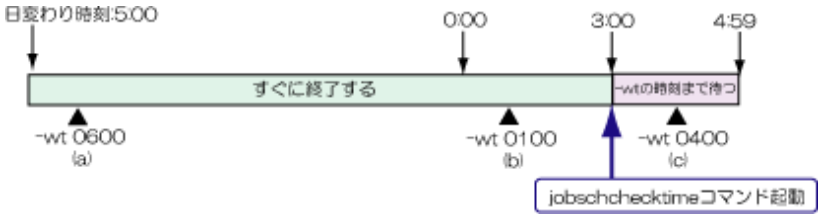
-sys サブシステム番号

複数サブシステム運用をしているシステムで、操作対象のサブシステムを指定するためのオプションです。操作対象とするサブシステム番号を、0から9の範囲で指定します。このオプションを省略した場合、サブシステム番号0が操作対象となります。

日変わり時刻を基準にした待ち合わせ時刻(-wt)とコマンド起動時刻について

待ち合わせ時刻(-wt)は、日変わり時刻を基準にした時刻です。

日変わり時刻と-wtで指定する待ち合わせ時刻、およびjobschchecktimeコマンド起動時の結果は、下図に示す関係があります。



jobschchecktimeコマンドが起動する時刻が、-wtで指定された待ち合わせ時刻より後であれば、コマンドはすぐに終了します。また、jobschchecktimeコマンドが起動する時刻が、-wtで指定された待ち合わせ時刻より前であれば、待ち合わせ時刻までコマンドの実行を待ちます。

例えば、日変わり時刻が5時で、jobschchecktime コマンドが起動する時刻が3時の場合の動きは以下のようになります。

a. 待ち合わせ時刻が6時(-wt 0600)の場合

   すでに待ち合わせ時刻を過ぎているので、すぐに終了します。

b. 待ち合わせ時刻が1時(-wt 0100)の場合

   すでに待ち合わせ時刻を過ぎているので、すぐに終了します。

c. 待ち合わせ時刻が4時(-wt 0400)の場合

   4時まで待ち合わせます。

ジョブネットの起動予定日とjobschchecktimeコマンドの待ち合わせ時刻(-wt)について

ジョブネットに登録されている時刻待ち合わせジョブ(jobschchecktimeコマンド)は、ジョブネットの起動予定日の待ち合わせ時刻を待ち合わせます。

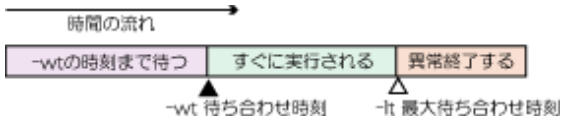
例：

```
ジョブネットの起動予定日時：9/1 20:00
時刻待ち合わせジョブ待ち合わせ時刻：22:00(-wt 2200)
```

上記のジョブ/ジョブネットにおいて、ジョブネットが異常終了または強制終了し、翌日9/2に再起動操作が行われた場合、時刻待ち合わせジョブ(jobschchecktimeコマンド)は、ジョブネット起動予定日の待ち合わせ時刻(9/1 22:00)を過ぎていることから、すぐに終了します。
再起動操作が行われた9/2の22:00を待ち合わせることはないので、注意が必要です。

待ち合わせ時刻(-wt)と最大待ち合わせ時刻(-lt)について

待ち合わせ時刻(-wt)と、最大待ち合わせ時刻(-lt)は、下図に示す関係があります。



jobschchecktimeコマンドが起動する時刻が、-wtで指定された待ち合わせ時刻より前であれば、待ち合わせ時刻までコマンドの実行を待ちます。

jobschchecktimeコマンドが起動する時刻が、-wtで指定された時刻以降、-ltで指定された時刻以前であれば、コマンドはすぐに実行されます。

jobschchecktimeコマンドが起動する時刻が、-ltで指定された時刻より後であれば、コマンドは復帰値1で異常終了します。

-lt 最大待ち合わせ時刻は、例えば以下のような使い方ができます。



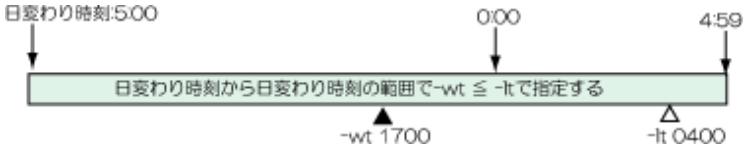
JOB 1：起動時刻9:00で数時間かかるジョブ
JOB 2：jobschchecktime -wt 1200 -lt 15:00
JOB 3：12:00 ～ 15:00の間に起動し始めなければいけないジョブ

上記のように、12:00から15:00の間に起動したいジョブJOB3があり、先行ジョブのJOB1は起動時刻が9:00であるが、数時間かかるような場合に、JOB1とJOB3の間に、最大待ち合わせ時刻を指定したjobschchecktimeコマンドのジョブJOB2を指定します。

JOB1が12:00より前に終了した場合は、JOB2は12:00になるまでコマンドの実行を待つので、JOB3は12:00より前には起動されません。JOB1が12:00から15:00の間に終了した場合は、JOB2のコマンドはすぐに実行され、終了するので、JOB3は12:00から15:00の間に起動します。JOB1の終了が15:00を超えた場合は、JOB2は異常終了するので、JOB3は実行されません。そのため、JOB3は必ず12:00から15:00に起動するように指定できます。

待ち合わせ時刻(-wt)／最大待ち合わせ時刻(-lt)の指定と日変わり時刻について

日変わり時刻と、待ち合わせ時刻(-wt)、最大待ち合わせ時刻(-lt)は、下図に示すように指定してください。



-wtおよび-ltに指定できる値は0000から2359ですが、日変わり時刻を基準に指定します。日変わり時刻の範囲で-wtの時刻が、-ltの時刻以前になるように指定します。

例えば、日変わり時刻が5:00の場合で、-wt 1700 -lt 0400と指定した場合、0400は日付を超えた翌日を指定したことになります。-wt 1700 -lt 1500と指定するなど、-ltの時刻の方が前の場合は、コマンドはエラーになり、ジョブは異常終了します。

復帰値

復帰値	意味
0	以下のいずれかであったことを示します。 ・ 待ち合わせ時刻以前に起動され、待ち合わせ時刻まで待った。 ・ 待ち合わせ時刻以降に起動され、最大待ち合わせ時刻が指定されていなかった。 ・ 待ち合わせ時刻と最大待ち合わせ時刻の間に起動された。
1	最大待ち合わせ時刻より後に起動されたことを示します。
21	オプションの指定に誤りがあることを示します。使用できないオプションを指定したり、オプションの値や組合せが不適切な場合にこの復帰値を返します。
22	システムまたはユーザの環境に関するエラーが発生したことを示します。

復帰値	意味
31	Systemwalker Operation Managerの環境に関する内部エラーが発生したことを示します。
35	その他の内部エラーが発生したことを示します。

コマンド格納場所

Windows	Systemwalker Operation Managerインストール先ディレクトリ¥MpWalker.JM¥bin
Solaris	/opt/FJSVJOBSC/bin
HP-UX	/opt/FHPJOBSCH/bin
AIX	/usr/FAIXJOBSC/bin
Linux	/opt/FJSVJOBSC/bin

使用例

待ち合わせ時刻を10:00、最大待ち合わせ時刻を11:00で待ち合わせる場合

```
jobschchecktime -wt 1000 -lt 1100
```

実行結果/出力形式

```
CT_WTIME=1000
CT_LTIME=1100
```

4.25 jobschchecklogコマンド

記述形式

(1)終端情報を元にファイルを解析する場合：

```
jobschchecklog -f filename -e regexp [-k key] [-s] [-v] [-z]
[-sys サブシステム番号]
```

(2)終端情報を削除する場合：

```
jobschchecklog -h -f filename [-k key] [-sys サブシステム番号]
```

(3)終端情報を扱わずファイル全体を解析する場合：

```
jobschchecklog -a -f filename -e regexp [-s] [-v] [-z]
[-sys サブシステム番号]
```

注)

“[-sys サブシステム番号]”はSystemwalker Operation Manager EEのオプションです。

機能説明

テキスト形式のファイルを1行ずつ解析し、正規表現が含まれるかどうか調べます。

オプションによって、解析を開始する位置をファイルの先頭にするか、前回の解析の直後にするかを指定することができます。

解析の終端は、ファイル名をユーザ名とキーを元にして終端情報として内部的に保存します。この終端情報は、オプションによって削除することができます。

ファイルに2048バイトを超える行が含まれている場合、2049バイト以降は無視します。

サーバが動作していないときは使用できません。

オプション

-a

ファイル全体を解析します。

-e *regexp*

指定した正規表現を解析に使用します。特殊文字として^$.*+?[]|()が使用できます。

-f *filename*

指定したファイルを解析の対象にします。

-h

終端情報を削除します。次回の解析は*filename*の先頭から行います。

-k *key*

指定したキーを終端情報の保存に使用します。指定しなかった場合は*filename*をキーとします。

-s

ファイルが存在しなかった場合に、正規表現が見つからなかった場合と同じ復帰値を返します。

-v

正規表現を含む行をすべて標準出力に出力します。

-z

正規表現が見つかった場合と見つからなかった場合の復帰値を逆にします。

EE GEE **-sys サブシステム番号**

複数サブシステム運用をしているシステムで、操作対象のサブシステムを指定するためのオプションです。操作対象とするサブシステム番号を、0から9の範囲で指定します。このオプションを省略した場合、サブシステム番号0が操作対象となります。

復帰値

復帰値	意味
0	記述形式(1)、(3)の場合に、-zを指定しなかったときは正規表現が見つかったことを示します。-zを指定したときは正規表現が見つからなかったことを示します。また、記述形式(2)の場合、処理が正常に終了したことを示します。
1	記述形式(1)、(3)の場合に、-zを指定しなかったときは正規表現が見つからなかったことを示します。-zを指定したときは正規表現が見つかったことを示します。
11	記述形式(1)、(3)の場合に、ファイルが見つからなかったことを示します。-sを指定した場合は、この復帰値を返すことはありません。
12	記述形式(1)、(3)の場合に、正規表現が不適切であることを示します。“[”と“]”の対応が取れていないときなどにこの復帰値を返します。

復帰値	意味
13	記述形式(1)、(3)の場合に、ファイルの読み込みに失敗したことを示します。アクセス権がなかったり、普通のファイルでなかったりした場合にこの復帰値を返します。
21	オプションの指定に誤りがあることを示します。使用できないオプションを指定したり、オプションの組合せが不適切な場合にこの復帰値を返します。
22	システムまたはユーザの環境に関するエラーが発生したことを示します。
31	Systemwalker Operation Managerの環境に関する内部エラーが発生したことを示します。
32	終端情報の保存に関する内部エラーが発生したことを示します。
35	その他の内部エラーが発生したことを示します。

コマンド格納場所

Windows	Systemwalker Operation Managerインストール先ディレクトリ¥MpWalker.JM¥bin
Solaris	/opt/FJSVJOBSC/bin
HP-UX	/opt/FHPJOBSCH/bin
AIX	/usr/FAIXJOBSC/bin
Linux	/opt/FJSVJOBSC/bin

使用例

C:¥TEMP¥file.txtに文字列「ABC」で始まる行が書き込まれたかどうかを調べる場合

```
jobschchecklog -f C:¥TEMP¥file.txt -e "^ABC.*" -k key1
```

実行結果/出力形式

- -vオプションを指定しない場合の出力例

```
CL_FILENAME=C:¥TEMP¥file.txt
CL_LOGCOUNT=1
CL_LOGTEXT=ABCDEF: logfile text
```

- -vオプションを指定した場合の出力例

```
ABCDEF: logfile text
```

注意事項

- 以下の場合はファイルが置き換えられたとみなし、終端情報を削除してファイルの先頭から解析します。
  - ファイルサイズが終端情報より小さい。
  - ファイル中の終端情報の位置にある文字が改行以外である。
- ファイルの最終行に改行が含まれない場合、-aが指定されていなければ1つ前の行で解析を終了します。-aが指定されていれば解析の対象にします。
- 運用上解析を行わなくなったファイルがある場合は、-hを使用して終端情報を削除してください。

4.26　chkfile / jobschchkfileコマンド

記述形式

【Windows版】

chkfile [-fwait [-sw]] [-fdelete [-sd]] [-fsize|-fupdate] [-i 監視間隔] [-a] ドライブ:ファイル名

chkfile [-wait|-update[-d:yyyymmdd][-t:hhmm]|-delete] [-i 監視間隔] {ドライブ:ファイル名|-file:ドライブ:リストファイル名 [-AND|-OR]}

chkfile [-z] ドライブ:ファイル名

【UNIX版】

jobschchkfile [-fwait [-sw]] [-fdelete [-sd]] [-fsize|-fupdate] [-i 監視間隔] ファイル名

jobschchkfile [-wait|-update[-d:yyyymmdd] [-t:hhmm]|-delete] [-i 監視間隔] {ファイル名|-file:リストファイル名 [-AND|-OR]}

jobschchkfile [-z] ファイル名

機能説明

ファイルの有無の確認、およびファイルの作成／更新／削除／サイズ変更を待つためのコマンドです。

オプション

-fwait

ファイルが存在しない場合は、作成されるまで待ちます。コマンドの開始時点でファイルが存在した場合は復帰しません。一度ファイルが削除された状態が確認され、再度作成されると復帰します。

-fdelete、-fupdate、-fsizeと組み合わせて待ち合わせることもできます。組み合わせた場合は、いずれかの条件を満たしたときに復帰します。

-sw

-fwaitとともに指定します。

指定すると、コマンドの開始時点でファイルが存在した場合にすぐに復帰します。

-fdelete

ファイルが存在する場合は、ファイルが削除されるまで待ちます。コマンドの開始時点でファイルが存在しなかった場合は復帰しません。一度ファイルが作成された状態が確認され、再度削除されると復帰します。

-fwait、-fupdate、-fsizeと組み合わせて待ち合わせることもできます。組み合わせた場合は、いずれかの条件を満たしたときに復帰します。

-sd

-fdeleteとともに指定します。

指定すると、コマンドの開始時点でファイルが存在しなかった場合にすぐに復帰します。

-fupdate

ファイルが存在した場合は、更新日付が変更されるまで待ちます。コマンドの開始時点でファイルが存在しなかった場合は、ファイルが作成されても復帰しません。一度ファイルが作成された状態が確認され、その後更新日付が更新された状態が確認されると復帰します。

-fwait、-fdeleteと組み合わせて待ち合わせることもできます。組み合わせた場合は、いずれかの条件を満たしたときに復帰します。-fsizeと組み合わせることはできません。

-fsize

ファイルのサイズが増減されるまで待ちます。ファイルが作成されたり削除されたりしても復帰しません。ファイルが更新されても、サイズに増減がない場合は復帰しません。サイズに増減がない更新を待ち合わせたい場合は、-fupdateを使用してください。

-fwait、-fdeleteと組み合わせて待ち合わせることもできます。組み合わせた場合は、いずれかの条件を満たしたときに復帰します。-fupdateと組み合わせることはできません。

-wait

ファイルが存在しない場合は、作成されるまで待ちます。コマンドの開始時点でファイルが存在した場合は、すぐに復帰します。

-delete、-update、-fwait、-fdelete、-fupdate、-fsizeと組み合わせて待ち合わせることはできません。

-delete

ファイルが存在する場合は、ファイルが削除されるまで待ちます。コマンドの開始時点でファイルが存在しなかった場合は、すぐに復帰します。

-wait、-update、-fwait、-fdelete、-fupdate、-fsizeと組み合わせて待ち合わせることはできません。

-update

ファイルが存在しない場合は、作成されるまで待ちます。

ファイルが存在した場合は、更新日付が変更されるまで待ちます。

-wait、-delete、-fwait、-fdelete、-fupdate、-fsizeと組み合わせて待ち合わせることはできません。

-d:yyyymmdd

-updateとともに指定します。

ファイルが更新されたと判断する基準の年月日を指定します。

例えば2014年6月5日の場合は、-d:20140605と指定します。

省略された場合は、現在の年月日以降となります。

-t:hhmm

-updateとともに指定します。

ファイルが更新されたと判断する基準の時分を指定します。

例えば13時35分の場合は、-t:1335と指定します。

省略された場合は、現在の時刻以降となります。

-i 監視間隔

ファイルの状態を確認する間隔を1秒～3600秒で指定します。省略された場合は、1秒間隔で確認します。

-a【Windows版】

-fwait、-fupdate、-fsizeとともに使用します。

ファイルの状態を確認する際、対象のファイルが他のプログラムからアクセスされなくなるまで待ちます。例えば、-fwaitでファイルが作成されるまで待つ場合、他のプログラムがファイルを作成中（アクセス中）であれば本コマンドは終了しません。次の監視間隔後の状態確認でファイルが作成済で、他のプログラムからアクセスされなくなっていたら終了します。

-z

ファイルの有無を確認し、ファイルが存在しなければ復帰値0で終了します。本オプションを指定せずにファイルの有無を確認した場合、ファイルが存在すれば復帰値0で終了します。

ドライブ:ファイル名【Windows版】

対象とするファイル名をフルパスで指定します。

-fwait、-fdelete、-fupdate、-fsize、-wait、-delete、-updateのいずれのオプションもない場合は、有無を確認するファイル名を指定します。

-fwait、-fdelete、-fupdate、-fsizeを指定した場合、ファイル名の一部または全体にワイルドカード“*”を使用することも可能です。ワイルドカードはファイル名だけに指定できます。ドライブ名やディレクトリ名には指定できません。

ファイル名【UNIX版】

対象とするファイル名を、フルパスで指定します。

-fwait、-fdelete、-fupdate、-fsize、-wait、-delete、-updateのいずれのオプションもない場合は、有無を確認するファイル名を指定します。

-fwait、-fdelete、-fupdate、-fsizeを指定した場合は、ファイル名の一部または全体にワイルドカード“*”を使用することも可能です。ただし、ワイルドカード“*”は、シェルによって特別な意味を持つ文字であるため、ワイルドカードを使用する場合は、“*”の直前にバックスラッシュ“¥”を記述してください。ワイルドカードはファイル名だけに指定できます。ディレクトリ名には指定できません。

-file:ドライブ:リストファイル名【Windows版】

-wait、-delete、-updateとともに使用します。

対象とするファイルを定義したファイル(リストファイル)名を指定します。

-wait、-delete、-updateのいずれのオプションもない場合は、有無を確認するファイル名を記述したファイル名を指定します。

なお、リストファイルには、複数のファイル名を記述することができます。

リストファイル名にはワイルドカードを指定できません。リストファイル名に記述するファイル名にもワイルドカードは指定できません。

-file:リストファイル名【UNIX版】

-wait、-delete、-updateとともに使用します。

対象とするファイルを定義したファイル(リストファイル)名を指定します。

-wait、-delete、-updateのいずれのオプションもない場合は、有無を確認するファイル名を記述したファイル名を指定します。

なお、リストファイルには、複数のファイル名を記述することができます。

リストファイル名にはワイルドカードを指定できません。リストファイル名に記述するファイル名にもワイルドカードは指定できません。

-AND|OR

-fileオプションでリストファイル名を指定した場合に、リストファイル中のすべてのファイルを対象にするか、任意の1ファイルを対象にするかを指定します。

省略するとリストファイル中のすべてファイルが対象となります。-wait、-delete、-updateの各オプションにおける動作を以下に示します。

- -wait オプションの場合：

-AND	リストファイル中のすべてのファイルが作成されるまで待ちます。 すべてのファイルが既に存在する場合は、即復帰します。

-OR	リストファイル中のどれか1つのファイルが作成されるまで待ちます。ファイルが既に1つでも存在する場合は、即復帰します。

- -delete オプションの場合：

-AND	リストファイル中のすべてのファイルが削除されるまで待ちます。ファイルが1つも存在しない場合は、復帰値1で異常終了します。
-OR	リストファイル中のどれか1つのファイルが削除されるまで待ちます。ファイルが既に1つでも存在しない場合は、即復帰します。

- -update オプションの場合：

-AND	リストファイル中のすべてのファイルが更新または作成されるまで待ちます。
-OR	リストファイル中のどれか1つのファイルが更新または作成されるまで待ちます。

復帰値

復帰値	意味
0	ファイルが作成/更新/削除されました。 または、-zオプションの指定がないときにファイルが存在しました。 または、-zオプションの指定があるときにファイルが存在しませんでした。
1	-zオプションの指定がないときにファイルが存在しませんでした。 または、-zオプションの指定があるときにファイルが存在しました。
10	内部エラーが発生しました。

コマンド格納場所

Windows	Systemwalker Operation Managerインストール先ディレクトリ¥MpWalker.JM¥bin
Solaris HP-UX AIX Linux	/usr/bin

使用例1

- d:¥tempディレクトリ配下に“file”で始まる名前のファイルが作成または削除されるのを待つ例【Windows版】

```
chkfile -fwait -fdelete d:¥temp¥file*
```

- /var/tmpディレクトリ配下に“file”で始まる名前のファイルが作成または削除されるのを待つ例【UNIX版】

```
jobschchkfile -fwait -fdelete /var/tmp/file¥*
```

使用例2

d:¥temp¥list_file内に記述されているすべてのファイル(c:¥a.txt, d:¥b.txt, d:¥c.txt)が、2014年12月5日13時30分以降に更新されるのを待つ例【Windows版】

```
chkfile -update -d:20141205 -t:1330 -file:d:¥temp¥list_file -AND
```

[リストファイル“d:¥temp¥list_file”の記載例]

```
c:¥a.txt
d:¥b.txt
d:¥c.txt
```

/var/tmp/list_file内に記述されているすべてのファイル(/home/a.dat, /home/b.dat, /home/c.dat)が、2014年12月5日13時30分以降に更新されるのを待つ例【UNIX版】

```
jobschchkfile -update -d:20141205 -t:1330 -file:/var/tmp/list_file -AND
```

[リストファイル“/var/tmp/list_file”の記載例]

```
/home/a.dat
/home/b.dat
/home/c.dat
```



参考

ジョブとして登録する

ファイル作成／更新／削除／サイズ変更を契機としてジョブネットを自動起動させる場合、chkfileコマンドやjobschchkfileコマンドをジョブとしてスケジュールします。ジョブネット内の先頭ジョブとして登録することにより、ファイル作成／更新／削除／サイズ変更を待ち合わせて後続の処理を行うことができます。

実行結果/出力形式

- chkfileの出力例【Windows版】

```
CF_FILENAME=d:¥temp¥file1130
CF_CHKTYPE=fwait
CF_FILESIZE=1024
CF_FILEYEAR=2014
CF_FILEDATE=1130
CF_FILETIME=2103
```

- jobschchkfileの出力例【UNIX版】

```
CF_FILENAME=/var/tmp/file1130
CF_CHKTYPE=fwait
CF_FILESIZE=1024
CF_FILEYEAR=2014
CF_FILEDATE=1130
CF_FILETIME=2103
```

注意事項

- -fwait、-fdelete、-fupdate、-fsize、-wait、-delete、-updateを指定して、ジョブとしてジョブスケジューラに登録した場合、ファイルが作成／更新／削除／サイズ変更されるまで、実行中のままとなります。投入キューのジョブ経過時間の制限値や実行多重度を確認して、問題があればキューの設定値を変更してください。
- 本コマンドは、-aオプションを使用することで、ファイルへのアクセス完了を待ち合わせることができます。しかし、ファイル転送途中にアクセスが一時的に中断されるような転送プログラムなどの場合、ファイル転送の終了を保証することができません。このような場合は、送信側で目的のファイルを送信後に、送信完了を通知するための小サイズのファイルを送信して、このファイルの更新を待ち合わせることで目的のファイルの転送が完了したことを認識するようにしてください。【Windows版】

- 本コマンドは、-iで指定した監視間隔でファイルの状態を確認し、前に監視したときの状態と次に監視したときの状態との差異で、ファイルの変化を検知します。そのため、監視間隔よりも短い時間にファイルの作成／削除／更新／サイズ変更などの変化が発生した場合、それらを正しく検知しない場合があります。例えば、前の監視から次の監視までの間で、ファイルが作成されてすぐに削除されると、ファイルの作成も削除も検知されません。
- 同一の監視間隔内に複数のファイルの変化を検知した場合、最初に検知されたファイルについての情報が出力されます。
- リストファイルには、1行につき1ファイル名を記述してください。空白やタブを含むことはできません。
- Solaris版、HP-UX版、AIX版、Linux x86版では、対象とするファイルのサイズは、2Gバイトを超えないようにしてください。
- 本コマンドは、NFS上のファイルなど、ネットワーク経由のファイルを想定していません。そのようなファイルを指定して実行すると、ネットワーク経由特有の問題が発生する場合があります。

  異常発生時は、再度コマンドを実行する、または、前回履歴の情報を基にネットワーク、および、接続先サーバの状況確認などを行ってください。

4.27 jobschnetmemoコマンド

記述形式

(1) 登録の場合：

```
jobschnetmemo -ent [プロジェクト名/]ジョブネット名 メモ内容 [-sys サブシステム番号] (注)
```

(2) 削除の場合

```
jobschnetmemo -del [プロジェクト名/]ジョブネット名 [-sys サブシステム番号] (注)
```

(3) 内容出力の場合

```
jobschnetmemo -out [プロジェクト名/]ジョブネット名 [-sys サブシステム番号] (注)
```

(4) 検索の場合

```
jobschnetmemo -find [-p プロジェクト名] -e 正規表現 [-m] [-sys サブシステム番号] (注)
```

注)

“[-sys サブシステム番号]”はSystemwalker Operation Manager EEのオプションです。

機能説明

ジョブネットにメモを登録/削除します。メモの内容を出力することもできます。

検索時は、正規表現を含むメモが登録されているジョブネット名の一覧が出力されます。

オプション

プロジェクト名【登録/削除の場合】

ジョブネット情報の格納先のプロジェクト名を指定します。jobschnetmemoコマンドの実行ユーザが操作、更新できるプロジェクトのうちジョブネットが登録されているプロジェクトが1つしかない場合は、省略できます。

プロジェクト名【内容出力の場合】

ジョブネット情報の格納先のプロジェクト名を指定します。jobschnetmemoコマンドの実行ユーザが参照できるプロジェクトのうちジョブネットが登録されているプロジェクトが1つしかない場合は、省略できます。

-p プロジェクト名【検索の場合】

ジョブネット情報の格納先のプロジェクト名を指定します。省略した場合は、jobschnetmemoコマンドの実行ユーザが参照できるプロジェクト内のジョブネット情報すべてに対して検索します。

ジョブネット名

登録、削除および内容出力の対象となるジョブネット名を指定します。

メモ内容

メモとして登録する文字列を128バイト以内で指定します。以下を指定することはできません。

- 先頭、末尾の半角空白、全角空白
- 半角カナ
- ユーザ定義文字

-e 正規表現

正規表現を256バイト以内で指定します。特殊文字として、^$.*+?[]|()が使用できます。

-m

ジョブネット名の後ろに半角空白で区切ってメモの内容を出力します。省略した場合は、プロジェクト名とジョブネット名のみが出力されます。

EE GEE

-sys サブシステム番号

複数サブシステム運用をしているシステムで、操作対象のサブシステムを指定するためのオプションです。操作対象とするサブシステム番号を、0から9の範囲で指定します。このオプションを省略した場合、サブシステム番号0が操作対象となります。

復帰値

復帰値	意味
0	登録/削除/内容出力の場合に、処理が正常に終了したことを示します。検索の場合は、正規表現を含むメモが登録されているジョブネットが1つ以上見つかったことを示します。
1	検索の場合に、正規表現を含むメモが登録されているジョブネットが見つからなかったことを示します。
11	指定されたプロジェクトが存在しないか、アクセス権が不足していることを示します。あるいは、プロジェクトを省略した場合に、アクセス権のあるプロジェクトが存在しない、または、アクセス権のあるプロジェクトが複数存在したことを示します。
12	登録/削除/内容出力の場合に、指定されたジョブネットが見つからなかったことを示します。
13	登録の場合に、メモ内容が不適切であることを示します。メモ内容の長さが長かったり、使用できない文字が含まれていたときなどにこの復帰値を返します。
14	検索の場合に、正規表現が不適切であることを示します。正規表現の長さが長かったり、正規表現が不適切な場合にこの復帰値を返します。

復帰値	意味
21	オプションの指定に誤りがあることを示します。使用できないオプションを指定したり、オプションの組合せが不適切な場合にこの復帰値を返します。
22	システムまたはユーザの環境に関するエラーが発生したことを示します。
31	Systemwalker Operation Managerの環境に関する内部エラーが発生したことを示します。
32	ジョブネットの情報の読み込みや書き込みに失敗したことを示します。

コマンド格納場所

Windows	Systemwalker Operation Managerインストール先ディレクトリ¥MpWalker.JM¥bin
Solaris HP-UX AIX Linux	/usr/bin

注意事項

- 正規表現が空白文字を含む場合、あるいは正規表現に特殊文字を使用する場合は、正規表現全体をダブルクォーテーション(")で囲ってください。正規表現にダブルクォーテーションを記述する場合は、ダブルクォーテーションの直前に“¥”を指定することでダブルクォーテーションを文字として扱うことが可能です。なお、直前に“¥”を記述したダブルクォーテーションは、正規表現全体を括るためのダブルクォーテーションとして扱われません。

使用例

登録の場合の使用例を以下に示します。

```
jobschnetmemo -ent 総務部門/kyuuyo 給与・賞与の計算 -sys 1
```

削除の場合の使用例を以下に示します。

```
jobschnetmemo -del 総務部門/kyuuyo -sys 1
```

内容出力の場合の使用例を以下に示します。

```
jobschnetmemo -out 総務部門/kyuuyo -sys 1
```

検索の場合の使用例を以下に示します。

```
jobschnetmemo -find -p 総務部門 -e "データ.*管理" -m -sys 1
```

出力例

内容出力の場合の例を以下に示します。

```
給与・賞与の計算
```

検索の場合に-mオプションを指定して実行したときの例を以下に示します。

```
総務部門 suitou 出納データの管理
総務部門 uriage 売り上げデータ管理
```

・
・

4.28　jobschconvertコマンド

記述形式

【Windows版】

```
jobschconvert [-sys { all | サブシステム番号}]　(注)
```

【UNIX版】

```
jobschconvert [SJIS|EUC] [-sys { all | サブシステム番号}]　(注)
```

注)

“[-sys { all | サブシステム番号}] ”はSystemwalker Operation Manager EEのオプションです。

機能説明

旧バージョンのジョブスケジューラのデータベースを本バージョンで使用できるデータベースに変換します。

オプション

EE GEE **-sys { all | サブシステム番号}**

複数サブシステム運用をしているシステムで、操作対象のサブシステムを指定するためのオプションです。このオプションを省略した場合、サブシステム番号0が操作対象となります。

all

すべてのサブシステムを操作対象とします。

サブシステム番号

操作対象とするサブシステム番号を、0から9の範囲で指定します。

SJIS【UNIX版】

データベース中に含まれている文字コードをSJISに変換します。

EUC【UNIX版】

データベース中に含まれている文字コードをEUCに変換します。

復帰値

復帰値	意味
0	正常終了
0以外	異常終了。エラーの内容はエラーメッセージを確認してください。

コマンド格納場所

Windows	Systemwalker Operation Managerインストール先ディレクトリ¥MpWalker.JM¥bin
Solaris HP-UX AIX Linux	/usr/bin

注意事項

- jobschconvertコマンドは、システム管理者(“Administrators”グループに属するユーザ/スーパーユーザ)のみ実行できます。一般利用者は、jobschconvertコマンドを使用することはできません。
- 本コマンドは、Systemwalker Operation Managerの以下のコンポーネントが停止状態である時にのみ起動可能です。
  - ジョブスケジューラサービス/デーモン
  - カレンダサービス/デーモン
- カレンダ名として、/?*;,¥"<>|:などが指定されていたものについては、移行されません。
- 変換前の文字コードは自動判定します。【UNIX版】
- オプションを省略した場合、データベース中に含まれている文字コードは変換しません。【UNIX版】
- 本バージョンで使用できるデータベースに対して本コマンドを実行した場合は、データベース中に含まれている文字コードの変換のみを行います。【UNIX版】
- 変換前の文字コードがUTF-8の場合、EUCオプションおよびSJISオプションを指定してもコード変換は行われません。【UNIX版】

4.29 jobschchkmsgevtコマンド

記述形式

```
jobschchkmsgevt メッセージ事象名 [-i 監視間隔]
```

機能説明

メッセージ事象待ち合わせジョブで呼び出されるコマンドです。

本コマンドがジョブとして実行されているジョブネットに対して発生するメッセージ事象(プロジェクト名やジョブネット名を省略して発生するメッセージ事象、プロジェクトやジョブネットを指定して発生するメッセージ事象)を待ち合わせます。

コマンド実行開始時点からのメッセージを待ち合わせ、待ち合わせ対象のメッセージ事象が発生すると終了します。

なお、発生したメッセージ事象のカウントやクリアのされ方については、“Systemwalker Operation Manager 運用ガイド”の“メッセージ事象とジョブネットの動作”を参照してください。

オプション

メッセージ事象名

待ち合わせるメッセージ事象名を12バイト以内で指定します。

空白、半角カナ、ユーザ定義文字および以下の半角文字を指定することはできません。
, :

-i 監視間隔

メッセージ事象の発生を確認する間隔を1秒～3600秒で指定します。省略された場合は60秒間隔で確認します。

復帰値

復帰値	意味
0	正常終了
0以外	異常終了。エラーの内容はエラーメッセージを確認してください。

コマンド格納場所

Windows	Systemwalker Operation Managerインストール先ディレクトリ¥MpWalker.JM¥bin
Solaris HP-UX AIX Linux	/usr/bin

注意事項

- 本コマンドは、Systemwalker Operation Managerのジョブ(メッセージ事象待ち合わせジョブ)として登録された場合にだけ実行することができます。コマンドプロンプトから直接実行するなど、Systemwalker Operation Managerのジョブ以外の方法で実行することはできません。
- 本コマンドが実行されているジョブネット以外に対して発生するメッセージ事象を待ち合わせることはできません。
- 本コマンドは、スケジュールサーバで発生するメッセージ事象のみを待ち合わせるコマンドです。ネットワークジョブや分散実行ジョブで実行することはできません。
- 本コマンドの同時実行数に制限はありませんが、本コマンド(メッセージ事象待ち合わせジョブ)が大量に実行されている場合、メッセージ事象発生履歴ファイルにアクセスが集中するため、コマンド(ジョブ)の終了が遅れる場合があります。
- 同一の監視間隔内にメッセージ事象が複数回発生した場合は、一番最初に発生したメッセージ事象が実行結果に出力されます。

使用例

メッセージ事象“ABC”を待ち合わせる場合のジョブの定義例を以下に示します。

```
jobschchkmsgevt ABC
```

実行結果/出力形式

```
CM_MSGEVENT=ABC
```

4.30 jobschchknetvarコマンド

記述形式

```
jobschchknetvar 変数名 [-num 判定用演算子,比較値[,判定用演算子2,比較値2] | -char 比較方法,比較文字列] [-z]
```

機能説明

ジョブネット変数判定ジョブで呼び出されるコマンドです。

指定したジョブネット変数に対し、判定条件を指定して値を判定します。値を数値とみなして大小比較する方法と、値を文字列とみなして文字列比較する方法とがあります。

判定条件を指定しない場合、変数の存在有無が判定され、ジョブネット変数の情報が標準出力に表示されます。判定対象の変数名の変数が存在しなかった場合、常に判定条件は一致しません。

判定に使用したジョブネット変数の情報は、「変数名=変数値」の形式で標準出力に表示されます。判定対象の変数が存在しなかった場合は、何も表示されません。

オプション

変数名

判定対象とする変数名を指定します。

判定の対象とするジョブネット変数名を64バイト以内で指定します。

ジョブネット変数名の形式は「ジョブネット変数のプレフィックス＋“.”(半角ドット)＋ジョブの出力情報に出力された名前(“NAME=VALUE”形式の“NAME”)」です。

-num、-charいずれのオプションもない場合は、変数の存在を確認する変数名を指定し、存在すれば条件一致となります。指定した変数が存在し、その値が空文字の場合は、「変数が存在する」扱いとなります。

-num 判定用演算子,比較値[,判定用演算子2,比較値2]

ジョブネット変数の値を数値とみなして比較します。

判定演算子、比較値だけを指定した場合は、比較値に対する比較条件を判定します。

判定演算子2、比較値2も指定した場合は、比較値から比較値2までの範囲条件を判定します。

- 数値の比較条件の場合

  “判定用演算子”は、以下の6種類です。

  - EQ(EQual)

    “比較値”と等しい

  - NE(Not Equal)

    “比較値”と等しくない

  - GT(Greater Than)

    “比較値”より大きい

  - GE(Greater than Equal to)

    “比較値”以上

  - LT(Less Than)

    “比較値”より小さい

  - LE(Less than Equal to)

    “比較値”以下

  変数の値と比較する比較値には-2147483648から2147483647までの整数値を設定します。

- 数値の範囲条件の場合

  “判定用演算子1”には、以下のいずれかを指定します。

  - GT(Greater Than)

    “比較値1”より大きい

  - GE(Greater than Equal to)

    “比較値1”以上

  “判定用演算子2”には、以下のいずれかを指定します。

  - LT(Less Than)

    “比較値2”より小さい

  - LE(Less than Equal to)

    “比較値2”以下

  変数の値と比較する“比較値1”および“比較値2”には、-2147483648から2147483647までの整数値を指定し、“比較値1”には“比較値2”より小さい数値を指定します。

-char 比較方法,比較文字列

ジョブネット変数の値を文字列とみなして比較します。

比較方法には、以下のいずれかを指定します。

- EQ(EQual)

  変数の値が比較文字列と等しい

- NE(Not Equal)

  変数の値が比較文字列と等しくない

- IN(INclude)

  変数の値が比較文字列を含む

- NI(Not Include)

  変数の値が比較文字列を含まない

比較文字列には、半角カナおよびユーザ定義文字を除く、512バイト以内の文字列を指定します。

比較文字列に何も入力しなかった場合は、以下の動作となります。

- 比較方法がEQのとき

  変数の値が空文字列のときに判定条件が一致します。

- 比較方法がNEのとき

  変数の値が空文字列以外のときに判定条件が一致します。

- 比較方法がINのとき

  常に判定条件が一致しません。

- 比較方法がNIのとき

  常に判定条件が一致します。

-z

判定結果の復帰値を逆にします。

- -zオプションを指定しない場合

  判定条件が一致すれば0で復帰し、判定条件が一致しなければ1で復帰します。、

- -zオプションを指定した場合

  判定条件が一致すれば1で復帰し、判定条件が一致しなければ0で復帰します。

復帰値

復帰値	意味
0	正常終了 -zオプションが指定されていない状態で判定条件が一致しました。 -zオプションが指定されている状態で判定条件が一致しませんでした。
1	正常終了 -zオプションが指定されていない状態で判定条件が一致しませんでした。 -zオプションが指定されている状態で判定条件が一致しました。
2以上	異常終了

コマンド格納場所

Windows	Systemwalker Operation Managerインストール先ディレクトリ¥MpWalker.JM¥bin
Solaris HP-UX AIX Linux	/usr/bin

注意事項

- 本コマンドは、Systemwalker Operation Managerのジョブとして登録された場合にだけ実行することができます。コマンドプロンプトから直接実行するなど、Systemwalker Operation Managerのジョブ以外の方法で実行することはできません。
- 本コマンドは、スケジュールサーバ上で設定されるジョブネット変数を判定するコマンドです。ネットワークジョブや分散実行ジョブで実行することはできません。
- システムが自動で設定するジョブネット変数は指定できません。

使用例

ジョブネット変数“JOB.ABC”の値が、文字列“xyz”と等しいか判定する場合のジョブの定義例を以下に示します。

```
jobschchknetvar JOB.ABC -char EQ,xyz
```

実行結果/出力形式

```
JOB.ABC=xyz
```

4.31 jobschsleepコマンド

記述形式

```
jobschsleep 待ち合わせ時間
```

機能説明

待ち合わせ時間で指定した時間を待ち合わせます。

オプション

待ち合わせ時間

待ち合わせる時間を0(秒)から86400(秒)の範囲で指定します。0を指定すると、すぐに復帰します。

復帰値

復帰値	意味
0	正常終了
0以外	異常終了

コマンド格納場所

Windows	Systemwalker Operation Managerインストール先ディレクトリ¥MpWalker.JM¥bin

Solaris HP-UX AIX Linux	/usr/bin

使用例

600秒待ち合わせる場合の使用例を以下に示します。

```
jobschsleep 600
```

実行結果/出力形式

```
SL_SLEEPSEC=600
```

4.32 jobschrestexecコマンド【Windows版】【Linux版】

記述形式

```
jobschrestexec 接続情報ファイル
```

機能説明

REST実行ジョブで呼び出されるコマンドです。

RESTインタフェースが提供されているクラウドサービス上のジョブ(API)の実行/完了待ち合わせ/強制終了を行います。

実行依頼したジョブのレスポンス結果のキーの値をジョブネット変数に設定し、後続ジョブと連携できます。

オプション

接続情報ファイル

コマンドに指定するパラメタを記載したファイル名を指定します。

256バイト以内のフルパスで指定します。

接続情報ファイルの詳細については、“20.9.1 接続情報ファイル”を参照してください。

復帰値

復帰値	意味
0	正常終了
1	コマンドの実行時の引数に誤りがあります。
2	接続情報ファイルのパラメタ設定値に誤りがあります。
3	コマンドの実行時に行うファイル操作に失敗しました。
4	指定された接続先URLへのHTTPリクエストが失敗しました。
5	指定された接続先URLへのHTTPリクエストがタイムアウトしました。
6	リクエストの完了待ち合わせがタイムアウトしました。 (リクエストの完了待ち合わせの設定をしている場合)
7	リクエストの実行結果が指定された正常終了値ではありません。 (リクエストの完了待ち合わせの設定をしている場合)
8	リクエストのレスポンス結果の解析に失敗しました。

復帰値	意味
9	Systemwalker Operation Managerのジョブとして登録されていないため、実行されませんでした。
10	認証情報の取得に失敗しました。
12	HTTPリクエストの署名に失敗しました。
99	内部エラーが発生しました。

コマンド格納場所

Windows	Systemwalker Operation Managerインストール先ディレクトリ¥MpWalker.JM¥bin
Linux	/usr/bin

使用例

接続情報ファイルのパスが/connectionfiles/connectioninfo.jsonの場合のジョブの定義例を以下に示します。

```
jobschrestexec /connectionfiles/connectioninfo.json
```

実行結果/出力形式

- クラウドサービス上のジョブへ実行依頼をした場合の出力例

```
HttpStatusCode=200
HttpResHeader=/response/resheader.json
HttpResBody=/response /resbody.json
```

- 実行依頼したクラウドサービス上のジョブの完了を待ち合わせる場合の出力例

```
HttpStatusCode=200
HttpResHeader=/response/resheader.json
HttpResBody=/response/resbody.json
ChkResultHttpStatusCode=200
ChkResultHttpResHeader=/response/monheader.json
ChkResultHttpResBody=/response/monbody.json
```

HttpResHeader、HttpResBody、ChkResultHttpResHeader、ChkResultHttpResBodyは、レスポンスの出力先の指定がない場合は出力されません。

- 強制終了リクエストをする設定の場合に強制終了を行う場合の出力例

【Windows版】

```
HttpStatusCode=200
HttpResHeader=/response/resheader.json
HttpResBody=/response/resbody.json
ジョブ実行制御：ジョブ jobschrestexec(jobno.hostname) は強制終了操作、または、ジョブ実行制御サービスの停止によって強制終了しました。終了コードは 0x40000222 です。
```

【Linux版】

```
HttpStatusCode=200
HttpResHeader=/response/resheader.json
HttpResBody=/response/resbody.json
```

・ コマンドの異常終了時の例

実行するHTTPリクエストに応じたレスポンスのステータスコードと、エラーメッセージが以下の形式で出力されます。

```
HttpStatusCode=xxxx(実行リクエストのレスポンスのステータスコード)
HttpResHeader=/response/resheader.json
HttpResBody=/response/resbody.json
ChkResultHttpStatusCode=xxxx(監視リクエストのレスポンスのステータスコード)
ChkResultHttpResHeader=/response/monheader.json
ChkResultHttpResBody=/response/monbody.json
エラーメッセージ（注）
```

注）該当のエラーメッセージは、以下のとおりです。

- MpJobsch: ERROR: 0609: HTTP request failed. RequestType=XXXX StatusCode=YYYY
- MpJobsch: ERROR: 8609: HTTP request failed. RequestType=XXXX StatusCode=YYYY
- MpJobsch: ERROR: 0613: The wait time for completion of the job on the cloud service has elapsed.
- MpJobsch: ERROR: 6613: The wait time for completion of the job on the cloud service has elapsed.
- MpJobsch: ERROR: 0614: The execution result of the job on the cloud service is not the specified normal exit value. ResultValue=XXXX
- MpJobsch: ERROR: 7614: The execution result of the job on the cloud service is not the specified normal exit value. ResultValue=XXXX
- MpJobsch: ERROR: 0615: The specified key is not found in the response result. ParamName=XXXX
- MpJobsch: ERROR: 7615: The specified key is not found in the response result. ParamName=XXXX

なお、レスポンス結果の指定キーをジョブネット変数に設定する指定をした場合は、以下の形式で出力されます。

```
指定のジョブネット変数名=レスポンス結果の指定のキーの値
```

注意事項

- 本コマンドは、Systemwalker Operation Managerのジョブ(REST実行ジョブ)として登録された場合にだけ実行することができます。コマンドプロンプトから直接実行するなど、Systemwalker Operation Managerのジョブ以外の方法で実行することはできません。
- 実行依頼したクラウドサービス上のジョブの完了の待ち合わせは、クラウドサービス側に実行ジョブの結果取得用APIが提供されている場合のみ可能です。
- REST実行ジョブの強制終了に合わせた、クラウドサービス上の実行中のジョブの強制終了は、クラウドサービス側に実行ジョブの強制終了用APIが提供されている場合のみ可能です。
- 強制終了リクエストをしない設定でREST実行ジョブを強制終了した場合、REST実行ジョブは強制終了されますが、実行中のクラウドサービス側のジョブは強制終了されません。
- 以下は、APIのレスポンスヘッダのContent-Typeおよびレスポンスボディの形式が application/json、application/xml、text/xmlの場合にだけ利用できます。
  - 実行依頼したクラウドサービスのジョブの完了待ち合わせ
  - REST実行ジョブの強制終了に合わせたクラウドサービス上のジョブの強制終了
  - レスポンス結果のジョブネット変数設定によるジョブ間連携
- HTTPレスポンスヘッダや HTTPレスポンスボディの内容は、標準出力には出力されません。接続情報ファイルに出力結果ファイルのパスを指定することでファイル出力できます。出力結果ファイルの文字コードは、

Systemwalker Operation Manager のインストール環境における既定の文字コードが使用されます。ただし、既定の文字コードでの出力に失敗した場合は、UTF-8 で出力されます。

- HTTPリクエストのステータスコードが200番台以外の場合、REST実行ジョブはHTTPリクエスト失敗のエラーとなります。
- 強制終了リクエストをする設定の場合、強制終了リクエスト失敗時にはシステムログを出力し、管理者に異常を通知します。
- 強制終了リクエストをする設定の場合でも、クラウドサービス上のジョブ実行依頼後にREST実行ジョブが何らかの原因で異常終了した場合、クラウドサービス上のジョブの強制終了は保証されません。
- 本コマンドは、ネットワークジョブや分散実行ジョブで実行することはできません。

第5章 ジョブスケジューラコマンド(印刷用)

本章では、Systemwalker Operation Managerが提供するジョブスケジューラコマンド(印刷用)について説明します。なお、コマンドは、Systemwalker Operation Managerのサーバ機能がインストールされているサーバで起動してください。

注意

WindowsのUAC機能を有効にしている環境でコマンドを実行する場合、OSにログインしたユーザの権限によって権限の昇格を求めるダイアログが表示され、対処が必要となります。

詳細は、“Systemwalker Operation Manager 運用ガイド”の、“Windowsでコマンドを実行する場合の注意事項【Windows版】”を参照してください。

注意

ジョブスケジューラコマンド実行時の注意事項

コマンド実行時には、以下の条件を確認してください。

- 標準出力、標準エラー出力がクローズされている環境から、コマンドを実行しないでください。
- Operation Managerサーバインストール時に指定した、Operation Managerデーモンの動作言語種別とLANG環境変数が同じであること。(例えば、Solarisの場合でSystemwalker Operation Managerの動作言語種別をSJISとしてインストールした時は、LANG環境変数にja_JP.PCKが設定されている環境から、ジョブスケジューラのコマンドを実行する必要があります。)【UNIX版】

コマンドの実行ユーザは、Windowsサーバにログオン/UNIXサーバにログインしているユーザとなります。コマンドをジョブスケジューラに登録して実行した場合、各コマンドのプロジェクト名の省略値は、コマンドを登録しているジョブネットが所属しているプロジェクトのプロジェクト名になります。

5.1 コマンド一覧

ジョブスケジューラコマンド(印刷用)で使用するコマンドには以下のものがあります。

jobschprintコマンド

ジョブスケジューラが管理している情報を標準出力(stdout)に出力します。jobschprintコマンドには、以下に示すオプションがあります。

( )内に大文字オプションが付加してあるオプションは、大文字と小文字どちらを指定してもコマンドの処理内容は同じです。ただし、-Rと-rだけは、コマンドの処理内容が異なるのでご注意ください。

-j(-J):

ジョブネットおよびジョブの一覧を出力します。

-n(-N):

ジョブネットの現在の状態を出力します。

-a(-A):

ジョブネットのプロパティ情報を出力します。

-e(-E):

実行中のジョブの一覧を出力します。

-f(-F):

ジョブネットの履歴情報を出力します。

-g(-G):

ジョブネットの履歴情報を日付指定により出力します。

-i(-I):

ジョブの登録情報を出力します。

-l(-L):

ジョブネットの運用予定一覧を出力します。ジョブネット名称も出力します。

-m(-M):

ジョブネットの運用予定一覧を出力します。ジョブネット名称は出力しません。

-p(-P):

ジョブの現在の状態を出力します。

-q(-Q):

ジョブネット起動キューにつながっているジョブネットの一覧を出力します。【Windows版】

-R:

グループ情報を制御文の形式で出力します。

-r:

ジョブネット情報を制御文の形式で出力します。

-rh:

カレンダの休日情報を制御文の形式で出力します。

-rsp:

起動日雛形情報を制御文の形式で出力します。

-s(-S):

ジョブネットの起動日一覧(3年分)を出力します。

-w(-W):

実行待ちのジョブの一覧を出力します。

-x(-X):

グループの現在の状態を出力します。

-y(-Y):

グループ内のジョブネットの一覧を出力します。

-b:

起動日雛形名の一覧を出力します。

-c:

日付設定の起動日雛形を出力します。

-d:

曜日設定の起動日雛形を出力します。

-h:

カレンダ情報を出力します。

-u:

ジョブスケジューラの利用者情報を出力します。

-v:

カレンダ名の一覧を出力します。

-o(-O):

プロジェクトの一覧を出力します。

注意

jobschprintコマンドの出力結果中の日付の表示形式は、以下のようになります。

【Windows版】

Windowsサーバのコントロールパネルの指定値に従います。

【UNIX版】

システムのLANGがシフトJIS、EUCまたはUTF-8の場合(日本語環境の場合)はyyyy/mm/dd、LANGがCなどの場合(日本語以外の環境の場合)はmm/dd/yyyyとなります。

jobschprintcsvコマンド

ジョブスケジューラが管理している情報をCSV形式で標準出力(stdout)に出力します。

jobschprintcsvコマンドには、以下に示すオプションがあります。

-n:

ジョブネットの現在の状態を出力します。

-g:

ジョブネットの履歴情報を日付指定により出力します。

-l:

ジョブネットの運用予定一覧を出力します。

-f:

ジョブネットの履歴情報を相対日指定により出力します。

stemprintprojectコマンド

プロジェクト情報を制御文の形式で出力します。

5.2 jobschprint -j(-J)コマンド

記述形式

jobschprint {-jl-J} [-sys サブシステム番号]　　(注)

注)

“[-sys サブシステム番号]”はSystemwalker Operation Manager EE/GEEのオプションです。

機能説明

登録されているジョブネットおよびジョブの一覧を出力します。

オプション

-jl-J

コマンドの実行ユーザが参照できるジョブネットおよびジョブの一覧を出力します。

EE GEE **-sys サブシステム番号**

複数サブシステム運用をしているシステムで、操作対象のサブシステムを指定するためのオプションです。操作対象とするサブシステム番号を、0から9の範囲で指定します。このオプションを省略した場合、サブシステム番号0が操作対象となります。

復帰値

復帰値	意味
0	正常終了
0以外	異常終了。エラーの内容はエラーメッセージを確認してください。

コマンド格納場所

Windows	Systemwalker Operation Managerインストール先ディレクトリ¥MpWalker.JM¥bin
Solaris HP-UX AIX Linux	/usr/bin

出力項目の説明

出力項目について説明します。

User Name

jobschprintコマンドの実行ユーザ名です。

Project Name

プロジェクト名です。

Job Net

ジョブネット名です。

Job Name

ジョブとして登録されているコマンド(コマンド名、バッチファイル名/シェルスクリプト名、プログラム名、またはジョブ実行制御のJCL名)です。

子ジョブネットをジョブとして出力する場合には、子ジョブネットのジョブネット名を出力します。

リンクジョブネットをジョブとして出力する場合には、リンクジョブネットのジョブネット名を出力します。

バッチフレームワークジョブをジョブとして出力する場合には、バッチジョブ定義名を出力します。

実行属性が“ジョブ実行制御”および“旧バージョン互換(旧・標準)”の場合に出力されます。【UNIX版】

GEE ジョブとして登録されているグローバルサーバジョブの業務IDまたはホストのジョブ名を表示します。

Job Param

コマンドに渡すパラメタです。

バッチフレームワークジョブをジョブとして出力する場合は出力されません。

実行属性が“ジョブ実行制御”の場合に出力されます。

Job Name JES

ジョブ実行制御のジョブ名です。

実行属性が“Interstage”の場合は出力されません。【UNIX版】

Job Comment

ジョブのコメントです。

子ジョブネットをジョブとして出力する場合には、子ジョブネットのジョブネット名称を出力します。

リンクジョブネットをジョブとして出力する場合には、リンクジョブネットのジョブネット名称を出力します。

Previous Job Name

先行ジョブ名です。

先行ジョブが子ジョブネットの場合には、先行ジョブとなっている子ジョブネットのジョブネット名を出力します。

先行ジョブがリンクジョブネットの場合には、先行ジョブとなっているリンクジョブネットのジョブネット名を出力します。

実行属性が“ジョブ実行制御”および“旧バージョン互換(旧・標準)”の場合に出力されます。【UNIX版】



先行ジョブ名です。先行ジョブがグローバルサーバジョブの場合は業務IDまたはホストのジョブ名を表示します。

Previous Job End Code

先行ジョブの終了コードによる起動条件を表示します。起動条件を比較条件とした場合と範囲条件とした場合では出力形式が異なります。判定用演算子については、“4.12.2 ジョブ制御文”を参照してください。

比較条件の場合:

判定用演算子 比較値

Previous Job End Code: GT 10

範囲条件の場合:

判定用演算子1 比較値1 and 判定用演算子2 比較値2

Previous Job End Code: GT 10 and LT 20

Job Attr

リカバリジョブのモードです。

ジョブがリカバリジョブの場合のみ出力されます。【Windows版】

Recovery Only

リカバリジョブです。

Recovery Rerun

リカバリジョブで、本リカバリジョブの終了後、先行ジョブを再実行します。

Work Unit Name

ジョブとして登録されているワークユニット名です。

実行属性が“PowerAIM”または“Interstage”の場合に出力されます。【UNIX版】

Previous System Name

先行となる業務システム名です。

実行属性が“PowerAIM”の場合に出力されます。【UNIX版】

System Name

業務システム名/システム名です。

実行属性が“PowerAIM”または“Interstage”の場合に出力されます。【UNIX版】

Previous Work Unit Name【UNIX版】

先行となるワークユニット名です。実行属性が“PowerAIM”の場合に出力されます。

Execution Day Mode

ジョブ(子ジョブネット、およびリンクジョブネット)の“起動日のみ有効”指定の有無(ON/OFF)です。アイコン種別が子ジョブネット、およびリンクジョブネットの場合に出力されます。

ON

“起動日のみ有効”を指定しています。

OFF

“起動日のみ有効”を指定していません。

Start Job

ジョブ(子ジョブネット、およびリンクジョブネット)が、起動条件が成立する/起動操作する場合に起動対象であるかどうかです。

アイコン種別が子ジョブネットおよびリンクジョブネットで、“起動日のみ有効”指定が“ON”の場合に出力されます。

ON

起動対象です。

OFF

起動対象ではありません。

Rerun Job

ジョブ（子ジョブネット、およびリンクジョブネット）が、再起動操作する場合に再起動対象であるかどうかです。

アイコン種別が子ジョブネットおよびリンクジョブネットで、“起動日のみ有効”指定が“ON”の場合に出力されます。

ON

再起動対象です。

OFF

再起動対象ではありません。

Or Mode

“先行ジョブをOR条件で待つ”指定の有無(ON/OFF)です。

ON

“先行ジョブをOR条件で待つ”を指定しています。

OFF

“先行ジョブをOR条件で待つ”を指定していません。

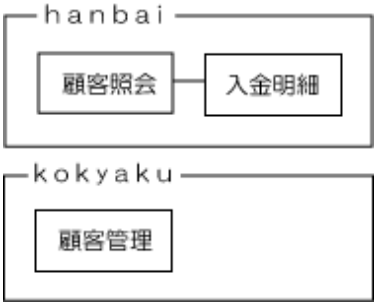
実行属性が“ジョブ実行制御”の場合に出力されます。

出力例

コマンドの出力結果“Job List”の例を以下に示します。

```
Job List                    date: 2002/02/25 time: 15:20

User Name: suzuki

Project Name: 総務部門

Job Net: hanbai

Job Name: kanri1.bat
Job Param: -file data1.dat
Job Name JES: kanri1
Job Comment: 顧客照会
Job Name: kanri2.bat
Job Param: -file data2.dat
Job Name JES: j123
Job Comment: 入金明細
Previous Job Name: kanri1.bat
Previous Job End Code: GT 10

Project Name: 総務部門

Job Net: kokyaku

Job Name: kanri.bat
Job Param: -file data2.dat
Job Name JES: j124
Job Comment: 顧客管理

Job Net: keiri
```

上記の例におけるジョブネットの構成を下図に示します。



5.3 jobschprint -n(-N)コマンド

記述形式

```
jobschprint {-n|-N} [-toplevel] [-long] [-sys サブシステム番号]      (注)
```

注)

“[-sys サブシステム番号]”はSystemwalker Operation Manager EEのオプションです。

機能説明

登録されているジョブネットについて、現在の状態を出力します。

オプション

-n|-N

コマンドの実行ユーザが参照できるジョブネットについて現在の状態を出力します。

-toplevel

子ジョブネット、およびリンクジョブネットを出力しません。階層化されていないジョブネットおよび階層化されているジョブネットのうち第1階層のものを出力します。

-long

出力項目“Job Net”および“Job Net Comment”を最大値まで出力します。

EE GEE

-sys サブシステム番号

複数サブシステム運用をしているシステムで、操作対象のサブシステムを指定するためのオプションです。操作対象とするサブシステム番号を、0から9の範囲で指定します。このオプションを省略した場合、サブシステム番号0が操作対象となります。

復帰値

復帰値	意味
0	正常終了
0以外	異常終了。エラーの内容はエラーメッセージを確認してください。

コマンド格納場所

Windows	Systemwalker Operation Managerインストール先ディレクトリ¥MpWalker.JM¥bin
Solaris HP-UX AIX Linux	/usr/bin

出力項目の説明

出力項目について説明します。

User Name

jobschprintコマンドの実行ユーザ名です。

Project Name

プロジェクト名です。

Job Net

ジョブネット名です。

先頭から8バイト分まで表示します。-longオプションを指定した場合は、50バイト分すべて表示します。

Job Net Comment

ジョブネット名称です。

先頭から24バイト分まで表示します。-longオプションを指定した場合は、100バイト分すべて表示します。

Status

ジョブネットの状態です。

以下に示した13種類の状態から選択表示されます。

Normal:

正常終了

Pseudo:

疑似正常

Abend:

異常終了

Forced:

強制終了

Warning:

警告

Execute:

実行中

Wait:

実行待ち

Pause:

停止中

Disable:

無効状態

Coped:

確認済

CarryOver:

持ち越し

StartDelay:

開始遅延

EndDelay:

終了遅延

[ジョブスケジューラ起動パラメタの定義]ウィンドウで疑似正常を無効にしている場合は、“疑似正常”状態は“正常終了”状態として表示されることに注意してください。

Scheduled Time

起動予定時刻です。

Start Time

前回のジョブネット開始時刻です。

End Time

前回のジョブネット終了時刻です。

Completion Code

ジョブネットの終了コードです。

注意事項

- UTF-8コードの環境では、ジョブネット名およびジョブネット名称に日本語を使用している場合、出力項目の位置がずれて出力されることがあります。【UNIX版】
- マスタスケジュール管理機能を有効にしている場合は、stemprint -n(-N)コマンドでジョブネットの現在の状態を出力してください。stemprint -n(-N)コマンドを利用することで、状態を出力したい運用日やプロジェクトを絞り込むことができます。stemprint -n(-N)コマンドの詳細は、“6.12 stemprint -n(-N)コマンド”を参照してください。

使用例

```
jobschprint -n -toplevel -sys 2
```

出力例

コマンドの出力結果“Job Net Status List”の例を以下に示します。

```
  Job Net Status List            date: 2000/10/25    time: 16:28

User Name: suzuki

Project Name: 営業

Job Net Job Net Comment Status  Scheduled Time   Start Time       End Time       Completion Code
hanbai  販売管理        Abend                   2000/10/25 10:22 2000/10/25 10:38 10
kokyaku 顧客売上管理    Wait    2000/10/25 18:00
eigyou  営業週報作成    Disable 2000/10/26       2000/10/18 23:00 2000/10/18 23:12


Project Name: 総務

Job Net Job Net Comment Status  Scheduled Time Start Time        End Time        Completion Code
suitou  出納データ管理  Execute 2000/10/28     2000/10/25 16:00
kyuuyo  給与・賞与計算  Normal  2000/11/10     2000/10/10 23:00  2000/10/10 23:50 0
```

5.4 jobschprint -a(-A)コマンド

記述形式

```
jobschprint {-a|-A} [-sys サブシステム番号]    (注)
```

注)

“[-sys サブシステム番号]”はSystemwalker Operation Manager EEのオプションです。

機能説明

登録されているジョブネットのプロパティ情報を出力します。

オプション

-a|-A

コマンドの実行ユーザが参照できるジョブネットについてプロパティ情報を出力します。

EE GEE

-sys サブシステム番号

複数サブシステム運用をしているシステムで、操作対象のサブシステムを指定するためのオプションです。操作対象とするサブシステム番号を、0から9の範囲で指定します。このオプションを省略した場合、サブシステム番号0が操作対象となります。

復帰値

復帰値	意味
0	正常終了
0以外	異常終了。エラーの内容はエラーメッセージを確認してください。

コマンド格納場所

Windows	Systemwalker Operation Managerインストール先ディレクトリ¥MpWalker.JM¥bin
Solaris HP-UX AIX Linux	/usr/bin

注意事項

毎月カレンダ・毎年カレンダ・一時変更関連の情報は出力されません。

詳細な情報を採取する場合は、jobschprint -rコマンドにより採取してください。

出力項目の説明

出力項目について説明します。

User Name

jobschprintコマンドの実行ユーザ名です。

Project Name

プロジェクト名です。

Job Net

ジョブネット名です。

Job Net Comment

ジョブネット名称です。

<No Execution>

起動条件“なし”指定の有無(ON/OFF)を表示します。

ON:

“なし”を指定しています。

OFF:

“なし”を指定していません。

<Single Time>

起動条件“単一起動”指定の有無(ON/OFF)を表示します。

ON:

“単一起動”を指定しています。

なお、“ON”の時、以下の項目が表示されます。

Day Change Time:

日変わり時刻です。

Start Time:

ジョブネットの起動時刻として指定した時刻です。

Stop Time:

ジョブネットの停止時刻として指定した時刻です。実行属性が“Interstage”の場合に出力されます。

Alarm Time:

ジョブネットの終了予定時刻として指定した時刻です。

OFF:

“単一起動”を指定していません。

<Multi Time>

起動条件“複数起動”指定の有無(ON/OFF)を表示します。

ON:

“複数起動”を指定しています。

なお、“ON”の時、以下の項目が表示されます。

Day Change Time:

日変わり時刻です。

Multi Time1～6:

1日に複数回起動するように指定された時刻です。

OFF:

“複数起動”を指定していません。

<Interval>

起動条件“一定間隔”指定の有無(ON/OFF)を表示します。

ON:

“一定間隔”を指定しています。

なお、“ON”の時、以下の項目が表示されます。

Day Change Time:

日変わり時刻です。

Start Time:

ジョブネットの起動時刻として指定した時刻です。

End Time:

ジョブネットの終了時刻として指定した時刻です。

Interval:

起動間隔時間を表示します。

OFF:

“一定間隔”を指定していません。

<IPL>

起動条件“サーバ起動時”指定の有無(ON/OFF)を表示します。

ON:

“サーバ起動時”を指定しています。

なお、“ON”の時、以下の項目が表示されます。

Day Change Time:

日変わり時刻です。

OFF:

“サーバ起動時”を指定していません。

<MSG>

起動条件“メッセージ起動”指定の有無(ON/OFF)を表示します。

ON:

“メッセージ起動”を指定しています。

なお、“ON”の時、以下の項目が表示されます。

Day Change Time:

日変わり時刻です。

Wait Condition:

メッセージ事象の待ち合わせ条件です。“AND”または“OR”が表示されます。ANDは登録したメッセージ事象のすべてが発生した場合にジョブネットを起動します。ORは登録したメッセージ事象のうち1つでも発生した場合にジョブネットを起動します。

Mode(order):

“ジョブネット実行中も有効”を指定している場合は“ON”、指定していない場合は“OFF”が表示されます。“ジョブネット実行中も有効”を指定した場合、ジョブネット実行中にメッセージ事象が発生しても、発生した回数分ジョブネットが起動されます。

Mode(Execution Day):

“起動日のみ有効”を指定している場合は“ON”、指定していない場合は“OFF”が表示されます。ジョブネットの起動予定日にメッセージ事象が発生した場合、“起動日のみ有効”を指定しているジョブネットは起動されます。

Mode(Time):

“起動時刻との条件”が表示されます。起動時刻の到来かつ、メッセージ事象が発生した時にジョブネットが起動される場合は、“AND”が表示されます。起動時刻の到来または、待ち合わせていたメッセージ事象の発生のどちらか一方の起動条件が満たされた時にジョブネットが起動される場合は、“OR”が表示されます。

Carry Over:

日変わり時刻が到来した時に持ち越し処理を行うように指定されている場合に“ON”が表示されます。

Message:

登録したメッセージ事象が表示されます。発生したメッセージの前に“*”が表示されます。ジョブネットが起動されると“*”は消えます。

OFF:

“メッセージ起動”を指定していません。

<Exec_Attr>

ジョブネット登録時に指定した“実行属性”です。

Normal:

ジョブスケジューラがジョブネットを実行します。【UNIX版】

JES:

ジョブ実行制御がジョブネットを実行します。

Interstage:

Interstageがジョブネットを実行します。

PAIM:

PowerAIMがジョブネットを実行します。【UNIX版】

なお、“JES”または“Interstage”の時は、以下の項目が表示されます。

HOST:

デフォルトホスト名です。実行サブシステム名を指定したジョブネットの場合、指定した実行サブシステム名がそのまま表示されます。【UNIX版】

<Copy Start>

ジョネット起動時に複写起動するか、複写起動しないかの指定です。

ON:

複写起動します。

OFF:

複写起動しません。

Save Number

複写先ジョブネットの保存数を表示します。

<Default_Host>

デフォルトホスト名です。実行サブシステム名を指定したジョブネットの場合、指定した実行サブシステム名がそのまま表示されます。【Windows版】

<Power Mode>

電源未投入時の処置です。起動条件に“単一起動”が設定された状態でジョブネットの起動時刻を過ぎてジョブスケジューラサービスが起動された場合の処置です。

ON:

ジョブスケジューラサービス起動時にジョブネットを起動します。

OFF:

ジョブスケジューラサービスが起動されてもジョブネットは起動されません。

<IPL Mode>

サーバ起動時の処置です。起動条件に“サーバ起動時”が設定された状態で1日にジョブスケジューラサービスが複数回起動された場合の処置です。

Everytime:

ジョブスケジューラサービスが起動されるごとにジョブネットを起動します。

Once:

ジョブスケジューラサービスが複数回起動されてもジョブネットは最初の1回のみ起動します。

<MSG Mode>

起動条件に“メッセージ起動”が設定された状態で待ち合わせているメッセージ事象が1日に複数回発生した場合の処置です。

Everytime:

待ち合わせているメッセージ事象が発生するたびにジョブネットを起動します。

Once:

待ち合わせているメッセージ事象が1日に複数回発生してもジョブネットは最初の1回のみ起動します。

なお、“Once”の時は、以下の項目が表示されます。

Msg Start Time:

ジョブネットの開始予定時刻として指定された時刻です。指定されていない場合は空白になります。

<Estimate Time>

ジョブネットの実行予測時間です。単位は“分”です。

<Standard Job Net>

基準ジョブネットとして宣言されている場合に“ON”が表示されます。

<Schedule Pattern>

基準日情報“起動日雛形”指定の有無(ON/OFF)を表示します。

ON:

“起動日雛形”を指定しています。

なお、“ON”の時は、以下の項目が表示されます。

Pattern Name:

基準日情報として指定した起動日雛形名です。

OFF:

“起動日雛形”を指定していません。

<Schedule Pattern Mode>:

複数の起動日雛形を指定している時の組み合わせ条件です。表示内容の意味を以下に示します。

AND:

すべての起動日雛形で起動日設定されている日のみを起動日とします。

OR:

少なくとも1つの起動日雛形で起動日設定されている日を起動日とします。

<Base Job Net>

基準日情報“他ジョブネット”指定の有無(ON/OFF)を表示します。

ON:

“他ジョブネット”を指定しています。

なお、“ON”の時は、以下の項目が表示されます。

Base Job Net Name:

基準日情報として、他ジョブネットを指定した時の基準ジョブネット名です。

OFF:

“他ジョブネット”を指定していません。

<Annually Calendar>

基準日情報“毎年カレンダ”指定の有無(ON/OFF)を表示します。

ON:

“毎年カレンダ”を指定しています。

OFF:

“毎年カレンダ”を指定していません。

<Monthly Calendar>

基準日情報“毎月カレンダ”指定の有無(ON/OFF)を表示します。

ON:

“毎月カレンダ”を指定しています。

OFF:

“毎月カレンダ”を指定していません。

<Offset Mode>

基準日からシフトする日数を数える方法です。

Execday:

稼働日でオフセット日数を数えます。

Day:

暦日でオフセット日数を数えます。

Offset Day

基準日からシフトするオフセット日数です。

Offset Holiday

基準日が休日の場合の扱い（ON/OFF）を表示します。

ON:

基準日が休日の場合は、稼働日とみなします。

OFF:

基準日が休日の場合は、休日とみなします。

<Holiday Calendar>

休日情報“カレンダ”指定の有無(ON/OFF)を表示します。

ON:

“カレンダ”を指定しています。

OFF:

“カレンダ”を指定していません。

なお、<Holiday Calendar>が“ON”の場合は、以下の項目が表示されます。

Calendar Name:

ジョブネット登録時に指定したカレンダ名です。

<Holiday Mode>:

複数のカレンダを指定している時の組み合わせ方です。表示内容の意味を以下に示します。

AND:

すべてのカレンダで休日設定されている日のみを休日とします。

OR:

少なくとも1つのカレンダで休日設定されている日を休日とします。

<Holiday Shift>

休日の取り扱いおよび運用期間です。表示内容の意味を以下に示します。

Before Shift:

ジョブネットの起動日を前日に変更します。

After Shift:

ジョブネットの起動日を翌日に変更します。

Cancel:

当日を休日として認識し、ジョブネットを起動しません。

Ignore:

当日を起動日として認識し、ジョブネットを起動します。

Shift Limit

休日シフト時に休日が連続した時に起動を抑止する日数が表示されます。

Enable Term

設定された起動日の有効期限です。

Disable Term

設定された起動日の無効期限です。

<Business Pattern>

基準日情報“営業日”指定の有無（ON/OFF）です。

ON

営業日を指定しています。

なお、“ON”の時は、以下の項目が表示されます。

Business Base

指定した月初め日です。表示内容の意味を以下に示します。

1～28

1日から28日です。

Last

月末です。

Business Day

指定した営業日です。複数指定した場合には、複数表示します。1つも指定していない場合には、表示しません。

1～31

第1営業日（月初め）から数えた営業日です。

-1～-31

第-1営業日（月締め）から逆算した営業日です。

Business Limit

“月度内の営業日のみを起動日とする”の有無（ON/OFF）です。

ON

“月度内の営業日のみを起動日とする”を指定しています。

OFF

“月度内の営業日のみを起動日とする”を指定していません。

OFF

営業日を指定していません。

<Carryover Mode>

“運用指示”の指定値を表示します。マスタスケジュール管理環境が構築され、日ごと管理のスケジュールが行われている場合にのみ表示します。

CANCEL：

“打ち切り”を指定しています。

OVER：

“完了待ち”を指定しています。

PARALLEL：

“並列走行”を指定しています。

ACTION：

“オペレータ対処”を指定しています。

出力例

コマンドの出力結果“Information of Job Net”の例を以下に示します。

```
Information of Job Net          date: 2002/02/25 time: 18:58

User Name: suzuki

Project Name: 営業部門

Job Net: hanbai
Job Net Comment: 販売管理
<No Execution>： OFF
<Single Time>: ON
  Day Change Time: 00:00
  Start Time: 08:00
  Alarm Time: 10:00
<Multi Time>: OFF
<Interval>: OFF
<IPL>: OFF
<MSG Mode>: OFF
<Estimate Time>: 60
<MSG>: OFF
<Exec_Attr>: JES
<Copy Start>: ON
  Save Number: 10
<Default_Host>: king
```

```
<Power Mode>: OFF
<IPL Mode>: Everytime
<Schedule Pattern>: ON
  Pattern Name: 毎日
<Schedule Pattern Mode>: AND
<Base Job Net>: OFF
<Annually Calendar>: OFF
<Monthly Calendar>: OFF
<Offset Mode>: Execday
  Offset Day: 0
  Offset Holiday: OFF
<Holiday Calendar>: ON
  Calendar Name: 西営業所
<Holiday Mode>: AND
<Holiday Shift>: Cancel
  Shift Limit: 0
  Enable Term: 2002/03/01,2002/03/31
  Disable Term:
<Business Pattern>: ON

Business Base: 20
  Business Day: 1
  Business Day: 3
  Business Day: 5
  Business Limit: ON
<Carryover Mode>: CANCEL
```

5.5 jobschprint -e(-E)コマンド

記述形式

```
jobschprint {-e|-E} [-jes] [-long] [-sys サブシステム番号]     (注)
```

注)

“[-sys サブシステム番号]”はSystemwalker Operation Manager EE/GEEのオプションです。

機能説明

実行中ジョブの一覧を出力します。

オプション

-e|-E

コマンドの実行ユーザが参照できるジョブについて実行中ジョブの一覧を出力します。

-jes

実行属性が“ジョブ実行制御”のジョブについては、出力項目“Job Name”にジョブ名を出力します。

-long

出力項目“Project Name”、“Job Net”、“Job Net Comment”および“Job Name”を最大値まで出力します。

EE GEE **-sys サブシステム番号**

複数サブシステム運用をしているシステムで、操作対象のサブシステムを指定するためのオプションです。操作対象とするサブシステム番号を、0から9の範囲で指定します。このオプションを省略した場合、サブシステム番号0が操作対象となります。

復帰値

復帰値	意味
0	正常終了
0以外	異常終了。エラーの内容はエラーメッセージを確認してください。

コマンド格納場所

Windows	Systemwalker Operation Managerインストール先ディレクトリ¥MpWalker.JM¥bin
Solaris HP-UX AIX Linux	/usr/bin

出力項目の説明

出力項目について説明します。

User Name

jobschprintコマンドの実行ユーザ名です。

Project Name

プロジェクト名です。

先頭から20バイト分まで表示します。-longオプションを指定した場合は、50バイト分すべて表示します。

Job Net

ジョブネット名です。

先頭から8バイト分まで表示します。-longオプションを指定した場合は、50バイト分すべて表示します。

Job Net Comment

ジョブネット名称です。

先頭から24バイト分まで表示します。-longオプションを指定した場合は、100バイト分すべて表示します。

Job Name

実行中のジョブです。実行中のジョブのコマンド(コマンド名、バッチファイル名/シェルスクリプト名、プログラム名またはジョブ実行制御のJCL名)です。先頭から24バイト分まで表示します。-longオプションを指定した場合は、300バイト分すべて表示します。

また、実行属性が“PowerAIM”または“Interstage”の場合はワークユニット名を表示します。

子ジョブネットをジョブとして出力する場合には、子ジョブネットのジョブネット名を出力します。リンクジョブネットをジョブとして出力する場合には、リンクジョブネットのジョブネット名を出力します。

GEE 実行中のグローバルサーバジョブの業務IDまたはホストのジョブ名を表示します。

-jesオプションを指定した場合、実行属性が“ジョブ実行制御”のジョブについては、ジョブ名を先頭から24バイト分まで表示します。-longオプションと-jesオプションを同時に指定した場合、実行属性が“ジョブ実行制御”のジョブについては、ジョブ名を64バイト分すべて表示します。なお、以下の場合には何も出力されません。

- ジョブ名が省略されている
- ジョブ実行制御属性以外の実行属性のジョブ
- 子ジョブネット

・ リンクジョブネット

Start Time

ジョブの実行開始時刻です。“西暦/月/日 時:分:秒”の形式で表示します。

子ジョブネットをジョブとして出力する場合には、子ジョブネットの開始時刻を出力します。

リンクジョブネットをジョブとして出力する場合には、リンクジョブネットの開始時刻を出力します。

注意事項【UNIX版】

・ UTF-8コードの環境では、プロジェクト名、ジョブネット名およびジョブネット名称に日本語を使用している場合、出力項目の位置がずれて出力されることがあります。

使用例

```
jobschprint -e -sys 2
```

出力例

コマンドの出力結果“Job Net Execution List”の例を以下に示します。

```
  Job Net Execution List          date: 1998/10/25 time: 16:58

User Name : suzuki

Project Name   Job Net Job Net Comment     Job Name           Start Time
総務部門       kyuuyo   給与・賞与計算      kyuuyo.bat         1998/10/25 18:00:21
               data     出納データ管理      suitou.bat         1998/10/25 13:00:00
                                           jobschmsgevent.exe 1998/10/25 16:57:00
```

5.6 jobschprint -f(-F)コマンド

記述形式

```
jobschprint {-f|-F} P [-vparam] [-long] [-sys サブシステム番号]    (注)
```

注)

“[-sys サブシステム番号]”はSystemwalker Operation Manager EEのオプションです。

機能説明

登録されているジョブネットの実行結果の履歴情報を出力します。なお、5日前より古い履歴情報を出力する場合は、ジョブネットの実行情報を日付指定により出力する“5.7 jobschprint -g(-G)コマンド”を参照してください。

オプション

-f|-F

コマンドの実行ユーザが参照できるジョブネットの履歴情報を出力します。

P

何日前の履歴情報を出力するかを0から5までの数字で指定します。

-vparam

可変パラメタを指定して起動されたジョブネットの場合、可変パラメタを表示します。

-long

出力項目“Job Net”および“Job Net Comment”を最大値まで出力します。

EE GEE **-sys サブシステム番号**

複数サブシステム運用をしているシステムで、操作対象のサブシステムを指定するためのオプションです。操作対象とするサブシステム番号を、0から9の範囲で指定します。このオプションを省略した場合、サブシステム番号0が操作対象となります。

復帰値

復帰値	意味
0	正常終了
0以外	異常終了。エラーの内容はエラーメッセージを確認してください。

コマンド格納場所

Windows	Systemwalker Operation Managerインストール先ディレクトリ¥MpWalker.JM¥bin
Solaris HP-UX AIX Linux	/usr/bin

出力項目の説明

出力項目について説明します。

User Name

jobschprintコマンドの実行ユーザ名です。

Day Change Time

日変わり時刻です。

Project Name

プロジェクト名です。

Job Net

ジョブネット名です。

先頭から8バイト分まで表示します。-longオプションを指定した場合は、50バイト分すべて表示します。

Job Net Comment

ジョブネット名称です。

先頭から24バイト分まで表示します。-longオプションを指定した場合は、100バイト分すべて表示します。

Status

ジョブネットの状態です。

以下に示した8種類の状態から選択表示されます。

Normal:

正常終了(終了コード:0)

Pseudo:

疑似正常(終了コード:1～正常終了コード上限値)

Abend:

異常終了(終了コード:正常終了コード上限値+1～255)

Forced:

強制終了(終了コード:256)

Execute:

実行中

E_Cancel:

実行拒否

Closed:

実行中断(終了コード:239)

Skipped:

スキップ

Start Time

ジョブネットが起動した時刻です。“西暦/月/日 時:分:秒”の形式で表示します。

End Time

ジョブネットが終了した時刻です。“西暦/月/日 時:分:秒”の形式で表示します。

End Code

ジョブネットの終了コードです。

Variable Parameter

可変パラメタを指定して起動されたジョブネットの可変パラメタです。-vparamオプションを指定した場合のみ表示されます。

注意事項

- UTF-8コードの環境では、ジョブネット名およびジョブネット名称に日本語を使用している場合、出力項目の位置がずれて出力されることがあります。【UNIX版】
- マスタスケジュール管理機能を有効にしている場合は、stemprint -f(-F)コマンドで履歴情報を出力してください。本コマンドの場合、マスタスケジュール管理機能で指定した[スケジュール情報を保管する世代数]を超えて管理対象外となった運用日の履歴情報は出力できません。stemprint -f(-F)コマンドの詳細は、“6.11 stemprint -f(-F)/-g(-G)コマンド”を参照してください。

使用例

```
jobschprint -f 0 -vparam -sys 2
```

出力例

コマンドの出力結果“History Information of Job Net”の例を以下に示します。

```
History Information of Job Net  date: 2007/08/13  time: 16:45

User Name: suzuki

Day Change Time: 00:00
```

```
Project Name: 総務部門

Job Net  Job Net Comment          Status   Start Time          End Time            End Code  Variable Parameter
kyuuyo   給与・賞与計算           Normal   2007/08/13 10:00:01 2007/08/13 11:33:32 0         ”200707”
backup   バックアップ業務         Normal   2007/08/13 10:30:01 2007/08/13 11:49:26 0
uriage   売上データ管理           Normal   2007/08/13 12:48:42 2007/08/13 12:50:56 0         ”sales01.dat”
kanri    顧客管理                 Abend    2007/08/13 13:32:45 2007/08/13 13:42:37 2
data     出納データ管理           Forced   2007/08/13 16:27:21 2007/08/13 16:31:21 256
data     出納データ管理           Execute  2007/08/13 16:44:30
```

5.7 jobschprint -g(-G)コマンド

記述形式

jobschprint {-g|-G} *YYYY MM DD* [-vparam] [-long] [-sys サブシステム番号]　(注)

注)

“[-sys サブシステム番号]”はSystemwalker Operation Manager EEのオプションです。

機能説明

登録されているジョブネットの実行結果の履歴情報を日単位で出力します。出力の対象日を、空白で区切った年月日の形式で指定できます。

オプション

-g|-G

コマンドの実行ユーザが参照できるジョブネットについて履歴情報を出力します。

YYYY

西暦を4桁の数字で指定します。

MM

出力したい日の月を2桁以内の数字で指定します。

DD

出力したい日を2桁以内の数字で指定します。

-vparam

可変パラメタを指定して起動されたジョブネットの場合、可変パラメタを表示します。

-long

出力項目“Job Net”および“Job Net Comment”を最大値まで出力します。

EE GEE

-sys サブシステム番号

複数サブシステム運用をしているシステムで、操作対象のサブシステムを指定するためのオプションです。操作対象とするサブシステム番号を、0から9の範囲で指定します。このオプションを省略した場合、サブシステム番号0が操作対象となります。

復帰値

復帰値	意味
0	正常終了
0以外	異常終了。エラーの内容はエラーメッセージを確認してください。

コマンド格納場所

Windows	Systemwalker Operation Managerインストール先ディレクトリ¥MpWalker.JM¥bin
Solaris HP-UX AIX Linux	/usr/bin

出力項目の説明

出力項目について説明します。

User Name

jobschprintコマンドの実行ユーザ名です。

Day Change Time

日変わり時刻です。

Project Name

プロジェクト名です。

Job Net

ジョブネット名です。

先頭から8バイト分まで表示します。-longオプションを指定した場合は、50バイト分すべて表示します。

Job Net Comment

ジョブネット名称です。

先頭から24バイト分まで表示します。-longオプションを指定した場合は、100バイト分すべて表示します。

Status

ジョブネットの状態です。

以下に示した8種類の状態から選択表示されます。

Normal:

正常終了

Pseudo:

疑似正常

Abend:

異常終了

Forced:

強制終了(終了コード:256)

Execute:

実行中

E_Cancel:

実行拒否

Closed:

実行中断(終了コード:239)

Skipped:

スキップ

Start Time

ジョブネットが起動した時刻です。“西暦/月/日 時:分:秒”の形式で表示します。

End Time

ジョブネットが終了した時刻です。“西暦/月/日 時:分:秒”の形式で表示します。

End Code

ジョブネットの終了コードです。

Variable Parameter

可変パラメタを指定して起動されたジョブネットの可変パラメタです。-vparamオプションを指定した場合のみ表示されます。

注意事項

- サイクリックに管理されているジョブスケジューラのログファイル(jobdb1.log/jobdb2.log/jobdb3.log)のうち、現在使用中のログファイルと1世代前のログファイルの履歴情報を出力します。
- UTF-8コードの環境では、ジョブネット名およびジョブネット名称に日本語を使用している場合、出力項目の位置がずれて出力されることがあります。【UNIX版】
- マスタスケジュール管理機能を有効にしている場合は、stemprint -g(-G)コマンドで履歴情報を出力してください。本コマンドの場合、マスタスケジュール管理機能で指定した[スケジュール情報を保管する世代数]を超えて管理対象外となった運用日の履歴情報は出力できません。stemprint -g(-G)コマンドの詳細は、“6.11 stemprint -f(-F)/-g(-G)コマンド”を参照してください。

使用例

```
jobschprint -g 2007 10 24 -vparam -sys 2
```

出力例

コマンドの出力結果“History Information of Job Net”の例を以下に示します。

```
History Information of Job Net  date: 2007/08/13  time: 16:45

User Name: suzuki

Day Change Time: 00:00

Project Name: 総務部門

Job Net  Job Net Comment          Status   Start Time          End Time            End Code  Variable Parameter
kyuuyo   給与・賞与計算            Normal   2007/08/13 10:00:01 2007/08/13 11:33:32 0         ”200707”
backup   バックアップ業務          Normal   2007/08/13 10:30:01 2007/08/13 11:49:26 0
uriage   売上データ管理            Normal   2007/08/13 12:48:42 2007/08/13 12:50:56 0         ”sales01.dat”
kanri    顧客管理                  Abend    2007/08/13 13:32:45 2007/08/13 13:42:37 2
data     出納データ管理            Forced   2007/08/13 16:27:21 2007/08/13 16:31:21 256
data     出納データ管理            Execute  2007/08/13 16:44:30
```

5.8 jobschprint -i(-I)コマンド

記述形式

jobschprint {-i|-I} [-sys サブシステム番号]　(注)

注)

“[-sys サブシステム番号]”はSystemwalker Operation Manager EE/GEEのオプションです。

機能説明

ジョブを登録した時に設定した情報(ジョブの登録情報)を出力します。なお、“標準出力/エラー出力を保存しない”、“再起動ポイント”、“終了コードに関わらず正常”、“次の再起動のみ無効”の値については、登録時の値よりもジョブの監視画面で変更された値を優先します。

オプション

-i|-I

コマンドの実行ユーザが参照できるジョブについての登録情報を出力します。

EE GEE

-sys サブシステム番号

複数サブシステム運用をしているシステムで、操作対象のサブシステムを指定するためのオプションです。操作対象とするサブシステム番号を、0から9の範囲で指定します。このオプションを省略した場合、サブシステム番号0が操作対象となります。

復帰値

復帰値	意味
0	正常終了
0以外	異常終了。エラーの内容はエラーメッセージを確認してください。

コマンド格納場所

Windows	Systemwalker Operation Managerインストール先ディレクトリ¥MpWalker.JM¥bin
Solaris HP-UX AIX Linux	/usr/bin

出力項目の説明

出力項目について説明します。

User Name

jobschprintコマンドの実行ユーザ名です。

Project Name

プロジェクト名です。

Job Net

ジョブネット名です。

Job Name

ジョブとして登録されているコマンド(コマンド名、バッチファイル名/シェルスクリプト名、プログラム名またはジョブ実行制御のJCL名)です。

子ジョブネットをジョブとして出力する場合には、子ジョブネットのジョブネット名を出力します。

リンクジョブネットをジョブとして出力する場合には、リンクジョブネットのジョブネット名を出力します。

バッチフレームワークジョブをジョブとして出力する場合には、バッチジョブ定義名を出力します。

実行属性が“ジョブ実行制御”または“旧バージョン互換(旧・標準)”の場合に出力されます。【UNIX版】

Job Param

コマンドのパラメタを出力します。

バッチフレームワークジョブをジョブとして出力する場合には出力されません。

Job Name JES

ジョブ実行制御で使用するジョブ名です。

子ジョブネットをジョブとして出力する場合には出力されません。

リンクジョブネットをジョブとして出力する場合には出力されません。

実行属性が“Interstage”の場合は出力されません。【UNIX版】

Job Comment

ジョブのコメントです。

子ジョブネットをジョブとして出力する場合には、子ジョブネットのジョブネット名称を出力します。

リンクジョブネットをジョブとして出力する場合には、リンクジョブネットのジョブネット名称を出力します。

Directory

実行属性が“ジョブ実行制御”の場合、qsub(ジョブの投入)を実行するディレクトリ名です。

子ジョブネットをジョブとして出力する場合には出力されません。

リンクジョブネットをジョブとして出力する場合には出力されません。

バッチフレームワークジョブをジョブとして出力する場合には出力されません。

実行属性が“PowerAIM”または“Interstage”の場合は出力されません。【UNIX版】

実行属性が“旧バージョン互換(旧・標準)”の場合、コマンドを実行するディレクトリ名です。【UNIX版】

System Name

業務システム名/システム名です。

実行属性が“PowerAIM”または“Interstage”の場合に出力されます。【UNIX版】

Work Unit Name

ジョブとして登録されているワークユニット名です。

実行属性が“PowerAIM”または“Interstage”の場合に出力されます。【UNIX版】

Host

ジョブの投入先ホスト名です。

子ジョブネットをジョブとして出力する場合には出力されません。

リンクジョブネットをジョブとして出力する場合には出力されません。

バッチフレームワークジョブをジョブとして出力する場合には出力されません。

実行属性が“ジョブ実行制御”または“Interstage”の場合に出力されます。【UNIX版】

Jobfile Transfer

“実行コマンドの転送”指定の有無(ON/OFF)を表示します。実行属性が“ジョブ実行制御”の場合に出力されます。

子ジョブネットをジョブとして出力する場合には出力されません。

リンクジョブネットをジョブとして出力する場合には出力されません。

バッチフレームワークジョブをジョブとして出力する場合には出力されません。

ON:

“実行コマンドの転送”を指定しています。

OFF:

“実行コマンドの転送”を指定していません。

Queue

ジョブの投入先キュー名です。

子ジョブネットをジョブとして出力する場合には出力されません。

リンクジョブネットをジョブとして出力する場合には出力されません。

バッチフレームワークジョブをジョブとして出力する場合には、ジョブキュー名を出力します。

実行属性が“ジョブ実行制御”または“Interstage”の場合に出力されます。【UNIX版】

Resource

ジョブが利用する資源名(リソース名)です。

子ジョブネットをジョブとして出力する場合には出力されません。

リンクジョブネットをジョブとして出力する場合には出力されません。

バッチフレームワークジョブをジョブとして出力する場合には出力されません。

実行属性が“ジョブ実行制御”または“Interstage”の場合に出力されます。【UNIX版】

Resource_Attr

ジョブが利用する資源(リソース)の割当て方法です。

子ジョブネットをジョブとして出力する場合には出力されません。

リンクジョブネットをジョブとして出力する場合には出力されません。

バッチフレームワークジョブをジョブとして出力する場合には出力されません。

実行属性が“ジョブ実行制御”または“Interstage”の場合に出力されます。【UNIX版】

Sh:

共有を指定しています。

Ex:

排他を指定しています。

Ignore Complete Code

“終了コードに関わらず正常”指定の有無(ON/OFF)を表示します。子ジョブネットをジョブとして出力する場合には出力されません。

リンクジョブネットをジョブとして出力する場合には出力されません。

ON:

“終了コードに関わらず正常”を指定しています。本ジョブの終了コードが“正常終了コード上限値”より大きい値でも後続のジョブは実行されます。

OFF:

“終了コードに関わらず正常”を指定していません。

No Output

“標準出力/エラー出力を保存しない”指定の有無(ON/OFF)を表示します。子ジョブネットをジョブとして出力する場合には出力されません。

リンクジョブネットをジョブとして出力する場合には出力されません。

実行属性が“ジョブ実行制御”の場合に出力されます。【Windows版】

実行属性が“Interstage”の場合は出力されません。【UNIX版】

ON:

“標準出力/エラー出力を保存しない”を指定しています。本ジョブの標準出力およびエラー出力は保存されません。

OFF:

“標準出力/エラー出力を保存しない”を指定していません。

バッチフレームワークジョブをジョブとして出力する場合には、“標準出力/標準エラー出力をジョブログファイルに保存しない”指定の有無(ON/OFF)を出力します。

ON：

“標準出力/標準エラー出力をジョブログファイルに保存しない”を指定しています。

OFF：

“標準出力/標準エラー出力をジョブログファイルに保存しない”を指定していません。

Restart Point

“再起動ポイント”指定の有無(ON/OFF)を表示します。

実行属性が“Interstage”の場合は出力されません。【UNIX版】

ON:

“再起動ポイント”を指定しています。ジョブネットの再起動時には本ジョブから起動されます。

OFF:

“再起動ポイント”を指定していません。

Normal Limit

“正常終了コード上限値”です。ジョブが正常に終了(疑似正常を含む)したか、異常終了したかを判断する値です。

子ジョブネットをジョブとして出力する場合には出力されません。

リンクジョブネットをジョブとして出力する場合には出力されません。

Prediction Time

ジョブの開始から終了までの実行予測時間を表示します。

子ジョブネットをジョブとして出力する場合には出力されません。

リンクジョブネットをジョブとして出力する場合には出力されません。

実行属性が“ジョブ実行制御”の場合に出力されます。【UNIX版】

Job Icon

ジョブのアイコン種別を表示します。表示内容および表示の意味を以下に示します。

表示内容	表示の意味(アイコン種別)
Command	コマンド実行
Message	メッセージ事象発生
SrvStart	サービス開始
SrvStop	サービス停止
FileWait	ファイル待ち合わせ
Backup	バックアップ連携
CL_Appl	クライアントアプリケーション起動
SV_Appl	サーバアプリケーション起動
CL_Press	クライアントファイル圧縮／伸長
SV_Press	サーバファイル圧縮／伸長
CL_Control	クライアントファイル操作
SV_Control	サーバファイル操作
CL_Copy	クライアントファイル転送
SV_Copy	サーバファイル転送
CL_Pon	クライアント電源投入
CL_Poff	クライアント電源切断
R/3	SAP ERP
SCS	SAPクラウドサービス連携【Windows版】【Linux版】
Interstage	Interstage
BatchJob	バッチフレームワークジョブ【EE】
EBS	Oracle EBS コンカレント
CL_RenameFile	クライアントファイル操作(ファイル名変更)
SV_RenameFile	サーバファイル操作(ファイル名変更)
CL_DelFile	クライアントファイル操作(ファイル削除)
SV_DelFile	サーバファイル操作(ファイル削除)
CL_MakeDir	クライアントファイル操作(ディレクトリ作成)
SV_MakeDir	サーバファイル操作(ディレクトリ作成)
CL_RemoveDir	クライアントファイル操作(ディレクトリ削除)
SV_RemoveDir	サーバファイル操作(ディレクトリ削除)
CheckLog	ログファイル解析
WaitTime	時刻待ち合わせ
CheckSrv	サービス状態確認
JobNet	ジョブネット
LinkJobNet	リンクジョブネット
WaitMessage	メッセージ事象待ち合わせ
Sleep	指定時間待ち合わせ
REST	REST API実行【Windows版】【Linux版】

表示内容	表示の意味(アイコン種別)
FileExist	ファイル存在判定
CheckNetVar	ジョブネット変数判定
MF-MSP	MSP【GEE】
MF-XSP	XSP【GEE】
User001(～User100)	ユーザ定義1(～ユーザ定義100)

Exec User

ジョブの実行ユーザ名を出力します。

バッチフレームワークジョブをジョブとして出力する場合には出力されません。

Job Env

ジョブの実行時の環境変数を出力します。出力形式は以下です。

Job Env*X*：環境変数名＝環境変数値数

X：1～32の整数

バッチフレームワークジョブをジョブとして出力する場合には、バッチジョブ定義の置換えパラメタを出力します。出力形式は以下です。

Job Env*X*: パラメタ名＝パラメタ値

X：1～32の整数

Next Time Disable

“次の再起動のみ無効”指定の有無(ON/OFF)を表示します。

実行属性が“Interstage”の場合は出力されません。【UNIX版】

ON:

“次の再起動のみ無効”を指定しています。

OFF:

“次の再起動のみ無効”を指定していません。

Previous Job Name

先行ジョブ名です。

先行ジョブ名が子ジョブネットの場合には、先行ジョブとなっている子ジョブネットのジョブネット名を出力します。

先行ジョブ名がリンクジョブネットの場合には、先行ジョブとなっているリンクジョブネットのジョブネット名を出力します。

実行属性が“PowerAIM”または“Interstage”の場合は出力されません。【UNIX版】

Previous Job End Code

先行ジョブの終了コードによる起動条件を表示します。起動条件を比較条件とした場合と範囲条件とした場合では出力形式が異なります。判定用演算子については、“4.12.2 ジョブ制御文”を参照してください。

比較条件の場合:

判定用演算子 比較値

Previous Job End Code: GT 10

範囲条件の場合:

判定用演算子1 比較値1 and 判定用演算子2 比較値2

Previous Job End Code: GT 10 and LT 20

Stdout File

ジョブの標準出力の出力先ファイル名を出力します。

バッチフレームワークジョブをジョブとして出力する場合には出力されません。

Stderr File

ジョブの標準エラー出力の出力先ファイル名を出力します。

バッチフレームワークジョブをジョブとして出力する場合には出力されません。

Stderr To Stdout File

“標準エラー出力を標準出力ファイルに出力する”の指定の有無(ON/OFF)を出力します。

バッチフレームワークジョブをジョブとして出力する場合には出力されません。

ON：

“標準エラー出力を標準出力ファイルに出力する”を指定しています。

OFF：

“標準エラー出力を標準出力ファイルに出力する”を指定していません。

Limit Time

ジョブの打ち切り時間を分単位で出力します。

Exec Priority

ジョブの実行優先度を出力します。

バッチフレームワークジョブをジョブとして出力する場合には、ジョブキュー内実行優先順位を出力します。

Job Attr

リカバリジョブのモードを表示します。

ジョブがリカバリジョブの場合のみ出力されます。【Windows版】

Recovery Only:

リカバリジョブです。先行ジョブの再起動は行いません。

Recovery Rerun:

リカバリジョブで、本リカバリジョブの終了後、先行ジョブを再実行します。

Previous System Name【UNIX版】

先行となる業務システム名です。実行属性が“PowerAIM”の場合に出力されます。

Previous Work Unit Name【UNIX版】

先行となるワークユニット名です。実行属性が“PowerAIM”の場合に出力されます。

GEE

Job ID

グローバルサーバジョブの業務IDまたはホストのジョブ名を表示します。

User Definition

グローバルサーバのユーザ名と対応して登録した、UNIXシステム上のグループ名を表示します。

Exec Host

ジョブを実行するグローバルサーバのホスト名を表示します。

Job Definition

SymfoNETでの業務定義の有無です。

ON:

SymfoNETに業務定義済みです。

OFF:

SymfoNETに業務定義が未定義です。

File Name

業務定義が未定義の場合に実行するファイル名を表示します。OS種別がXSPの場合はホスト上のファイル名またはJCM名、MSPの場合はデータセット名となります。

Retry

グローバルサーバに接続できなかった時にリトライをする間隔を表示します。リトライ間隔ごとに10回リトライを行います。

Alternate Host

実行ホスト名に指定したホストに接続できない場合に、代わりに接続するホスト名を表示します。

Option

ジョブに渡すパラメタを表示します。

Job Kind

起動する業務の種別を表示します。グローバルサーバのOS種別がXSPの場合のみ表示されます。

- BATCH

  バッチ業務を起動します。

- JCM

  JCM(ジョブ制御マクロ)業務を起動します。

Job Log File

ジョブログファイル名を出力します。

バッチフレームワークジョブをジョブとして出力する場合にのみ出力します。指定がない場合は出力しません。

Beginning Step Name

開始ステップ名を出力します。

バッチフレームワークジョブをジョブとして出力する場合にのみ出力します。指定がない場合は出力しません。

End code of last step

“最後に実行されたジョブステップの終了コードをジョブの終了コードとする”の指定の有無(ON/OFF)を出力します。

バッチフレームワークジョブをジョブとして出力する場合にのみ出力します。

ON：

“最後に実行されたジョブステップの終了コードをジョブの終了コードとする”を指定しています。

OFF：

“最後に実行されたジョブステップの終了コードをジョブの終了コードとする”を指定していません。

Execution Day Mode

アイコン種別が子ジョブネット、およびリンクジョブネットの場合に、ジョブ(子ジョブネット、およびリンクジョブネット)の“起動日のみ有効”指定の有無(ON/OFF)を表示します。

ON

“起動日のみ有効”を指定しています。

OFF

“起動日のみ有効”を指定していません。

Netvar Prefix

ジョブネット変数名のプレフィックスを表示します。

バッチフレームワークジョブをジョブとして出力する場合には出力されません。

実行属性が“ジョブ実行制御”の場合に出力されます。

Or Mode

“先行ジョブをOR条件で待つ”指定の有無(ON/OFF)を表示します。

ON

“先行ジョブをOR条件で待つ”を指定しています。

OFF

“先行ジョブをOR条件で待つ”を指定していません。

実行属性が“ジョブ実行制御”の場合に出力されます。

出力例

コマンドの出力結果“Information of Job”の例を以下に示します。

```
Information of Job date: 2002/03/26 time: 10:00

User Name: Suzuki

Project Name: 総務部門

Job Net: kanri

Job Name: kanri1.bat
Job Param: -file data1.dat
Job Name JES: kanri1
Job Comment: 顧客照会
Directory: c:¥tmp
Host: queen
Jobfile Transfer: ON
Queue: queue3
Resource: resource1
Resource_Attr: Sh
Ignore Complete Code: ON
No Output: ON
```

```
Restart Point: OFF
Normal Limit: 10
Prediction Time: 10
Job Icon: Command
Exec User: administrator
Job Env1: TEMP_PATH = d:¥temp
Job Env2: DATA_PATH = d:¥kanri¥data
Next Time Disable: OFF
Stdout File: c:¥tmp¥outfile.log
Stderr File: c:¥tmp¥errfile.log
Stderr To Stdout File: OFF
Limit Time: 30
Exec Priority: 0

Project Name: 総務部門

Job Net: hanbai

Job Name: hanbai1.bat
Job Param: -file data2.dat
Job Name JES: hanbai1
Job Comment: 集金反映
Directory: c:¥tmp
Host: atlas
Jobfile Transfer: ON
Queue: queue2
Resource: resource2
Resource_Attr: ExIgnore Complete Code: OFF
No Output: ON
Restart Point: OFF
Normal Limit: 1
Prediction Time: 20
Job Icon: Command
Exec User: administrator
Job Env1: DATA_PATH = d:¥hanbai¥data
Next Time Disable: OFF
Stdout File: c:¥tmp¥err.log
Stderr File:
Stderr To Stdout File: ON
Limit Time: 5
Exec Priority: 0
Job Name: cldnet
Job Comment: cldnet
Restart Point: OFF
Job Icon: JobNet
Next Time Disable: OFF
Execution Day Mode: OFF
```

5.9 jobschprint -l(-L)コマンド

記述形式

```
jobschprint {-l|-L} YYYY MM [-long] [-sys サブシステム番号]     (注)
```

注)

“[-sys サブシステム番号]”はSystemwalker Operation Manager EEのオプションです。

機能説明

登録されているジョブネットについて、指定した年月の運用予定一覧を出力します。

オプション

-l|-L

コマンドの実行ユーザが参照できるジョブネットについて、指定した年月の運用予定一覧を出力します。ジョブネット名称も出力します。

なお、複写起動されたジョブネットは出力されません。

YYYY

西暦を4桁の数字で指定します。指定できる年は、今年、来年および再来年です。年変わり直後であっても昨年は指定できません。

MM

出力したい日の月を2桁以内の数字で指定します。

-long

出力項目“Job Net”および“Job Net Comment”を最大値まで出力します。

EE GEE

-sys サブシステム番号

複数サブシステム運用をしているシステムで、操作対象のサブシステムを指定するためのオプションです。操作対象とするサブシステム番号を、0から9の範囲で指定します。このオプションを省略した場合、サブシステム番号0が操作対象となります。

復帰値

復帰値	意味
0	正常終了
0以外	異常終了。エラーの内容はエラーメッセージを確認してください。

コマンド格納場所

Windows	Systemwalker Operation Managerインストール先ディレクトリ¥MpWalker.JM¥bin
Solaris HP-UX AIX Linux	/usr/bin

出力項目の説明

出力項目について説明します。

“*”が付いている日はジョブネットの起動日です。“@”が上に付いている日は日曜日です。

User Name

jobschprintコマンドの実行ユーザ名です。

Project Name

プロジェクト名です。

Job Net

ジョブネット名です。

先頭から8バイト分まで表示します。-longオプションを指定した場合は、50バイト分すべて表示します。

Job Net Comment

ジョブネット名称です。

先頭から24バイト分まで表示します。-longオプションを指定した場合は、100バイト分すべて表示します。

注意事項【UNIX版】

- UTF-8コードの環境では、ジョブネット名およびジョブネット名称に日本語を使用している場合、出力項目の位置がずれて出力されることがあります。

使用例

```
jobschprint -l 2007 10 -sys 2
```

出力例

コマンドの出力結果“Job Net Execution Day Matrix”の例を以下に示します。

```
  Job Net Execution Day Matrix    date: 1998/10/25  time: 16:52

User Name: suzuki

Project Name: 営業部門

1998/11

-------------------------------------------------------------------------------------
        |                |     @            @            @            @
        |                | 0                10                20                30
Job Net |Job Net Comment | 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0
-------------------------------------------------------------------------------------
hanbai  |販売管理        | *            *            *            *            *
kokyaku |顧客売上管理    | *            *            *            *            *
eigyou  |営業週報作成    |       *            *            *            *
-------------------------------------------------------------------------------------
```

5.10 jobschprint -m(-M)コマンド

記述形式

```
jobschprint {-m|-M} YYYY MM [-long] [-sys サブシステム番号]  (注)
```

注)

“[-sys サブシステム番号]”はSystemwalker Operation Manager EEのオプションです。

機能説明

登録されているジョブネットについて、指定した年月の運用予定一覧を出力します。

オプション

-m|-M

コマンドの実行ユーザが参照できるジョブネットについて、指定した年月の運用予定一覧を出力します。

なお、複写起動されたジョブネットは出力されません。

YYYY

西暦を4桁の数字で指定します。指定できる年は、今年、来年および再来年です。年変わり直後であっても昨年は指定できません。

MM

出力したい日の月を2桁以内の数字で指定します。

-long

出力項目“Job Net”を最大値まで出力します。

EE GEE **-sys サブシステム番号**

複数サブシステム運用をしているシステムで、操作対象のサブシステムを指定するためのオプションです。操作対象とするサブシステム番号を、0から9の範囲で指定します。このオプションを省略した場合、サブシステム番号0が操作対象となります。

復帰値

復帰値	意味
0	正常終了
0以外	異常終了。エラーの内容はエラーメッセージを確認してください。

コマンド格納場所

Windows	Systemwalker Operation Managerインストール先ディレクトリ¥MpWalker.JM¥bin
Solaris HP-UX AIX Linux	/usr/bin

出力項目の説明

出力項目について説明します。

“*”が付いている日はジョブネットの起動日です。“@”が上に付いている日は日曜日です。

User Name

jobschprintコマンドの実行ユーザ名です。

Project Name

プロジェクト名です。

Job Net

ジョブネット名です。

先頭から8バイト分まで表示します。-longオプションを指定した場合は、50バイト分すべて表示します。

注意事項【UNIX版】

- UTF-8コードの環境では、ジョブネット名に日本語を使用している場合、出力項目の位置がずれて出力されることがあります。

使用例

```
jobschprint -m 2007 10 -sys 2
```

出力例

コマンドの出力結果“Job Net Execution Day Matrix”の例 を以下に示します。

```
Job Net Execution Day Matrix    date: 1998/10/25 time: 16:52

User Name: suzuki

Project Name: 営業部門

1998/11

------------------------------------------------------------------------
       |    @            @            @            @
       | 0               10               20               30
Job Net| 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0
------------------------------------------------------------------------
hanbai | *            *            *            *            *
kokyaku| *            *            *            *            *
eigyou |      *            *            *            *
------------------------------------------------------------------------
```

5.11 jobschprint -p(-P)コマンド

記述形式

```
jobschprint {-p|-P} [-jes] [-long] [-sys サブシステム番号]  (注)
```

注)

“[-sys サブシステム番号]”はSystemwalker Operation Manager EE/GEEのオプションです。

機能説明

登録されているジョブについて、現在の状態を出力します。

オプション

-p|-P

コマンドの実行ユーザが参照できるジョブについて現在の状態を出力します。

-jes

実行属性が“ジョブ実行制御”のジョブについては、出力項目“Job Name”にジョブ名を出力します。

-long

出力項目“Job Net”および“Job Name”を最大値まで出力します。

EE GEE **-sys サブシステム番号**

複数サブシステム運用をしているシステムで、操作対象のサブシステムを指定するためのオプションです。操作対象とするサブシステム番号を、0から9の範囲で指定します。このオプションを省略した場合、サブシステム番号0が操作対象となります。

復帰値

復帰値	意味
0	正常終了
0以外	異常終了。エラーの内容はエラーメッセージを確認してください。

コマンド格納場所

Windows	Systemwalker Operation Managerインストール先ディレクトリ¥MpWalker.JM¥bin
Solaris HP-UX AIX Linux	/usr/bin

出力項目の説明

出力項目について説明します。

User Name

jobschprintコマンドの実行ユーザ名です。

Project Name

プロジェクト名です。

Job Net

ジョブネット名です。

先頭から8バイト分まで表示します。-longオプションを指定した場合は、50バイト分すべて表示します。

Job Name

登録されているジョブです。ジョブとして登録されているコマンド(コマンド名、バッチファイル名/シェルスクリプト名、プログラム名またはジョブ実行制御のJCL名)です。実行属性が“Interstage”の場合は、ワークユニット名です。

子ジョブネットをジョブとして出力する場合には、子ジョブネットのジョブネット名を出力します。

リンクジョブネットをジョブとして出力する場合には、リンクジョブネットのジョブネット名を出力します。

先頭から24バイト分まで表示します。なお、-longオプションを指定した場合は、300バイト分すべて表示します。

GEE

登録されているグローバルサーバジョブの業務IDまたはホストのジョブ名を表示します。

-jesオプションを指定した場合、実行属性が“ジョブ実行制御”のジョブについては、ジョブ名を先頭から24バイト分まで表示します。-longオプションと-jesオプションを同時に指定した場合、実行属性が“ジョブ実行制御”のジョブについては、ジョブ名を64バイト分すべて表示します。なお、以下の場合には何も出力されません。

- ジョブ名が省略されている
- ジョブ実行制御属性以外の実行属性のジョブ
- 子ジョブネット
- リンクジョブネット

Status

ジョブの状態です。

子ジョブネットをジョブとして表示する場合には、子ジョブネットの状態を表示します。

リンクジョブネットをジョブとして表示する場合には、リンクジョブネットの状態を表示します。

以下に示した13種類の状態から選択表示されます。

Normal:

正常終了

Pseudo:

疑似正常

Abend:

異常終了

Forced:

強制終了

Execute:

実行中(子ジョブネットで状態が終了遅延の場合も、実行中として表示されます)

Delay:

実行遅延(子ジョブネット場合は実行遅延になりません)

Warning:

警告(子ジョブネットの場合のみ)

Wait:

実行待ち

Pause:

停止中

Disable:

無効状態

Pass & W:

実行待ちのジョブがパスされた状態

Pass & P:

停止中のジョブがパスされた状態

Pass & D:

無効状態のジョブがパスされた状態

[ジョブスケジューラ起動パラメタの定義]ウィンドウで疑似正常を無効にしている場合は、“疑似正常”状態は“正常終了”状態として表示されることに注意してください。

Start Time

ジョブの開始時刻です。

子ジョブネットをジョブとして出力する場合には、子ジョブネットの開始時刻を出力します。

リンクジョブネットをジョブとして出力する場合には、リンクジョブネットの開始時刻を出力します。

End Time

ジョブの終了時刻です。

子ジョブネットをジョブとして出力する場合には、子ジョブネットの終了時刻を出力します。

リンクジョブネットをジョブとして出力する場合には、リンクジョブネットの終了時刻を出力します。

Completion Code

ジョブの終了コードです。

子ジョブネットをジョブとして出力する場合には、子ジョブネットの終了コードを出力します。

リンクジョブネットをジョブとして出力する場合には、リンクジョブネットの終了コードを出力します。

注意事項【UNIX版】

- UTF-8コードの環境では、ジョブネット名およびジョブネット名称に日本語を使用している場合、出力項目の位置がずれて出力されることがあります。

使用例

```
jobschprint -p -sys 2
```

出力例

コマンドの出力結果“Job Status List”の例を以下に示します。

```
  Job Status List              date: 1998/10/25   time: 17:31

User Name: suzuki

Project Name: 総務部門

Job Net  Job Name      Status    Start Time        End Time         Completion Code
hanbai   kanri1.bat    Normal    1998/10/25 17:20  1998/10/25 17:25 0
         kanri2.bat    Execute   1998 10/25 17:30
         kanri3.bat    Pass & W
         kanri4.bat    Wait
kokyaku  ko_kanri.bat  Normal    1998/10/21 20:24  1998/10/21 20:56 0
eigyou   shuuhou.bat   Wait
kyuuyo   kyuuyo.bat    Normal    1998/10/10 22:00  1998/10/10 22:48 0
data     suitou.bat    Execute   1998/10/25 17:29
kanri    kokyaku.bat   Abend     1998/10/25 15:00  1998/10/25 15:14 2
```

5.12 jobschprint -q(-Q)コマンド【Windows版】

記述形式

```
jobschprint {-q|-Q} [-long] [-sys サブシステム番号]  (注)
```

注)

“[-sys サブシステム番号]”はSystemwalker Operation Manager EEのオプションです。

機能説明

ジョブネット起動APIによる起動要求で、現在、ジョブネット起動キューにつながっているジョブネットの一覧を表示します。

オプション

-q|-Q

コマンドの実行ユーザが参照できるジョブネットについてジョブネット起動キューにつながっているジョブネットの一覧を出力します。

-long

出力項目“Project Name”および“Job Net”を最大値まで出力します。

EE **-sys サブシステム番号**

複数サブシステム運用をしているシステムで、操作対象のサブシステムを指定するためのオプションです。操作対象とするサブシステム番号を、0から9の範囲で指定します。このオプションを省略した場合、サブシステム番号0が操作対象となります。

復帰値

復帰値	意味
0	正常終了
0以外	異常終了。エラーの内容はエラーメッセージを確認してください。

コマンド格納場所

Windows	Systemwalker Operation Managerインストール先ディレクトリ¥MpWalker.JM¥bin
Solaris HP-UX AIX Linux	/usr/bin

出力項目の説明

出力項目について説明します。

User Name

jobschprintコマンドの実行ユーザ名です。

Project Name

プロジェクト名です。

先頭から20バイト分まで表示します。-longオプションを指定した場合は、50バイト分すべて表示します。

Job Net

ジョブネット名です。

先頭から8バイト分まで表示します。-longオプションを指定した場合は、50バイト分すべて表示します。

Parameter

ジョブに渡すパラメタ(先頭30バイト)です。

使用例

```
jobschprint -q -sys 2
```

出力例

コマンドの出力結果“Queuing Job Net List”の例を以下に示します。

```
  Queuing Job Net List            date: 1998/10/25  time: 17:12

User Name : suzuki
```

```
Project Name      Job Net      Parameter
総務部門          kanri
営業部門          backup       d:¥users¥default
営業部門          backup       e:¥users¥default
総務部門          kintai
```

5.13 jobschprint -Rコマンド

記述形式

jobschprint -R [プロジェクト名/]グループ名 [-operate] [-sys サブシステム番号] (注)

注)

“[-sys サブシステム番号]”はSystemwalker Operation Manager EEのオプションです。

機能説明

グループ制御文およびグループ内ジョブネット制御文で指定された情報を出力します。

オプション

-R

登録されているグループの情報をjobschsetgrpコマンドで指定するグループ制御文の記述形式で出力します。

プロジェクト名

グループの情報が格納されているプロジェクト名を指定します。jobschprintコマンドの実行ユーザが参照可能なプロジェクトのうち、グループが登録されているプロジェクトが1つしか存在しない場合は省略できます。

グループ名

制御文を展開するグループ名を指定します。

-operate

グループ制御文に、operateオペランド（操作による停止または無効の状態）を出力します。出力される値は、“PAUSE”、“DISABLE”、“NONE”のいずれかです。本オプションを省略した場合、operateオペランドは出力されません。

EE GEE

-sys サブシステム番号

複数サブシステム運用をしているシステムで、操作対象のサブシステムを指定するためのオプションです。操作対象とするサブシステム番号を、0から9の範囲で指定します。このオプションを省略した場合、サブシステム番号0が操作対象となります。

復帰値

復帰値	意味
0	正常終了
0以外	異常終了。エラーの内容はエラーメッセージを確認してください。

コマンド格納場所

Windows	Systemwalker Operation Managerインストール先ディレクトリ¥MpWalker.JM¥bin

Solaris HP-UX AIX Linux	/usr/bin

出力項目の説明

出力項目については、“4.13.1 グループ制御文”および“4.13.2 グループ内ジョブネット制御文”を参照してください。

出力例

コマンドの出力結果の例を以下に示します。

```
GRP
groupname shuuhou
groupcomment 週報
jobnet 2

operate PAUSE
;
GNT
jobnetname eigyou
jobnetnumber 1
iconposition 1,1
;
GNT
jobnetname hanbai
jobnetnumber 2
pre_jobnet 1
iconposition 2,1
;
```

5.14 jobschprint -rコマンド

記述形式

```
jobschprint -r [プロジェクト名/]ジョブネット名 [-detail]
            [-operate] [-sysサブシステム番号]                (注)
```

注)

“[-sys サブシステム番号]”はSystemwalker Operation Manager EE/GEEのオプションです。

機能説明

ジョブネット制御文およびジョブ制御文で指定された情報を出力します。

オプション

-r

登録されているジョブネットの情報をjobschsetnetコマンドで指定するジョブネット制御文の記述形式で出力します。

プロジェクト名

ジョブネットの情報が格納されているプロジェクト名を指定します。jobschprintコマンドの実行ユーザが参照可能なプロジェクトのうち、ジョブネットが登録されているプロジェクトが1つしか存在しない場合は省略できます。

ジョブネット名

情報を出力するジョブネット名を指定します。

-detail

jobschprintrオペランド(3年分の起動日ビットパターン)が出力されます。特異日を含むジョブネットの起動日をjobschsetnetコマンドで登録させたい場合に指定します。

-operate

operateオペランド（操作による停止または無効の状態）を出力します。出力される値は、“PAUSE”、“DISABLE”、“NONE”のいずれかです。本オプションを省略した場合、operateオペランドは出力されません。

EE GEE

-sys サブシステム番号

複数サブシステム運用をしているシステムで、操作対象のサブシステムを指定するためのオプションです。操作対象とするサブシステム番号を、0から9の範囲で指定します。このオプションを省略した場合、サブシステム番号0が操作対象となります。

復帰値

復帰値	意味
0	正常終了
0以外	異常終了。エラーの内容はエラーメッセージを確認してください。

コマンド格納場所

Windows	Systemwalker Operation Managerインストール先ディレクトリ¥MpWalker.JM¥bin
Solaris HP-UX AIX Linux	/usr/bin

出力項目の説明

出力項目については、“4.12.1 ジョブネット制御文”および“4.12.2 ジョブ制御文”を参照してください。

出力例

コマンドの出力結果の例を以下に示します。

```
NET
jobnetname hanbai
jobnetcomment 販売管理
starttime 2000
holidayshift IGN
execattr JES
copystart ON
jobnetsavenumber 10
defaulthost queen
operate NONE
```

```
job 2
;
JOB
jobname kanri1.bat
jobnumber 1
jobparam -file data2.dat
jobcomment 入金明細
pre_job 1
jobicon FWAT
execuser administrator
stdoutfile c:¥tmp¥exec.log
stderrfile c:¥tmp¥error.log
priority -10
envname INPUT
envvalue FILE
envname OUTPUT
envvalue DAT
iconposition 1,1
operate NONE
;
JOB
jobname child1
jobnumber 2
pre_job 1
jobicon JNET
iconposition 1,2
;
```

5.15 jobschprint -s(-S)コマンド

記述形式

```
jobschprint {-s|-S} [-sys サブシステム番号]  (注)
```

注)

“[-sys サブシステム番号]”はSystemwalker Operation Manager EEのオプションです。

機能説明

登録されているジョブネットについて、3年分(今年、来年および再来年)の起動日一覧を出力します。

オプション

-s|-S

コマンドの実行ユーザが参照できるジョブネットについて3年分の起動日一覧を出力します。

なお、複写起動されたジョブネットは出力されません。

EE GEE **-sys サブシステム番号**

複数サブシステム運用をしているシステムで、操作対象のサブシステムを指定するためのオプションです。操作対象とするサブシステム番号を、0から9の範囲で指定します。このオプションを省略した場合、サブシステム番号0が操作対象となります。

復帰値

復帰値	意味
0	正常終了
0以外	異常終了。エラーの内容はエラーメッセージを確認してください。

コマンド格納場所

Windows	Systemwalker Operation Managerインストール先ディレクトリ¥MpWalker.JM¥bin
Solaris HP-UX AIX Linux	/usr/bin

出力項目の説明

出力項目の説明を以下に示します。

日付の下に“*”が付いている日がジョブネットの起動日です。

User Name

jobschprintコマンドの実行ユーザ名です。

Project Name

プロジェクト名です。

Job Net

ジョブネット名です。

Job Net Comment

ジョブネット名称です。

Assortment

起動日雛形名です。

複数の起動日雛形が選択されている場合は、半角空白で区切って出力します。

Assortment Mode

起動日雛形の組み合わせ条件です。起動日雛形が複数選択されている場合に出力します。

AND:

すべての起動日雛形で起動日設定されている日のみを起動日とします。

OR:

少なくとも1つの起動日雛形で起動日設定されている日を起動日とします。

Scheduled Time

起動予定時刻です。

出力例

コマンドの出力結果“Job Net Execution Day List”の例を以下に示します。

```
Job Net Execution Day List     date: 2009/11/14  time: 11:40
```

```
User Name: Administrator

Project Name: 営業部門

Job Net: hanbai
Job Net Comment: 販売管理
Assortment: 毎週月曜日 毎週金曜日
Assortment Mode: OR
Scheduled Time: 20:00

2009/01

 Sun  Mon  Tue  Wed  Thu  Fri  Sat
                    1    2    3
                         *
  4    5    6    7    8    9   10
       *                   *
 11   12   13   14   15   16   17
       *                   *
 18   19   20   21   22   23   24
       *                   *
 25   26   27   28   29   30   31
       *                   *

2009/02

 Sun  Mon  Tue  Wed  Thu  Fri  Sat
  1    2    3    4    5    6    7
       *                   *
  8    9   10   11   12   13   14
       *                   *
 15   16   17   18   19   20   21
       *                   *
```

5.16 jobschprint -w(-W)コマンド

記述形式

jobschprint {-w|-W} [-jes] [-long] [-sys サブシステム番号]　(注)

注)

“[-sys サブシステム番号]”はSystemwalker Operation Manager EE/GEEのオプションです。

機能説明

実行待ちジョブの一覧を出力します。ただし、実行待ちジョブがパスされた場合は、一覧に出力されません。

ジョブネットが実行待ちまたは開始遅延状態で、かつ、ジョブが実行待ち以外の状態(正常終了など)の場合のジョブネットは、ジョブネット一覧を出力します。

オプション

-w|-W

コマンドの実行ユーザが参照できるジョブについて実行待ちジョブの一覧を出力します。

-jes

実行属性が“ジョブ実行制御”のジョブについては、出力項目“Job Name”にジョブ名を出力します。

-long

出力項目“Project Name”、“Job Net”、“Job Net Comment”および“Job Name”を最大値まで出力します。

EE GEE **-sys サブシステム番号**

複数サブシステム運用をしているシステムで、操作対象のサブシステムを指定するためのオプションです。操作対象とするサブシステム番号を、0から9の範囲で指定します。このオプションを省略した場合、サブシステム番号0が操作対象となります。

復帰値

復帰値	意味
0	正常終了
0以外	異常終了。エラーの内容はエラーメッセージを確認してください。

コマンド格納場所

Windows	Systemwalker Operation Managerインストール先ディレクトリ¥MpWalker.JM¥bin
Solaris HP-UX AIX Linux	/usr/bin

出力項目の説明

出力項目の説明を以下に示します。

User Name

jobschprintコマンドの実行ユーザ名です。

Project Name

プロジェクト名です。

先頭から20バイト分まで表示します。-longオプションを指定した場合は、50バイト分すべて表示します。

Job Net

ジョブネット名です。

先頭から8バイト分まで表示します。-longオプションを指定した場合は、50バイト分すべて表示します。

Job Net Comment

ジョブネット名称です。

先頭から24バイト分まで表示します。-longオプションを指定した場合は、100バイト分すべて表示します。

Job Name

実行待ちのジョブです。実行待ちのジョブのコマンド(コマンド名、バッチファイル名/シェルスクリプト名、プログラム名またはジョブ実行制御のJCL名)です。

子ジョブネットをジョブとして出力する場合には、子ジョブネットのジョブネット名を出力します。

リンクジョブネットをジョブとして出力する場合には、リンクジョブネットのジョブネット名を出力します。

先頭から24バイト分まで表示します。なお、-longオプションを指定した場合は、300バイト分すべて表示します。

また、実行属性が“Interstage” または“PowerAIM”【UNIX版】の場合は、ワークユニット名です。

GEE

実行待ちのグローバルサーバジョブの業務IDまたはホストのジョブ名を表示します。

-jesオプションを指定した場合、実行属性が“ジョブ実行制御”のジョブについては、ジョブ名を先頭から24バイト分まで表示します。-longオプションと-jesオプションを同時に指定した場合、実行属性が“ジョブ実行制御”のジョブについては、ジョブ名を64バイト分すべて表示します。なお、以下の場合には何も出力されません。

- ジョブ名が省略されている
- ジョブ実行制御属性以外の実行属性のジョブ
- 子ジョブネット
- リンクジョブネット

Scheduled Time

実行待ちのジョブを含むジョブネットの起動予定時刻です。“西暦/月/日 時:分”の形式で表示されます。

注意事項【UNIX版】

- UTF-8コードの環境では、プロジェクト名、ジョブネット名、ジョブネット名称およびジョブ名に日本語を使用している場合、出力項目の位置がずれて出力されることがあります。

使用例

```
jobschprint -w -sys 2
```

出力例

コマンドの出力結果“Job Net Wait List”の例を以下に示します。なお、起動予定がない場合は、空白が表示されます。

```
  Job Net Wait List                    date: 1998/10/25 time: 17:02

User Name : suzuki

Project Name  Job Net   Job Net Comment   Job Name         Scheduled Time
総務部門      hanbai    販売管理          kanri1.bat
                                          kanri2.bat
                                          kanri3.bat
              kokyaku   顧客売上管理      ko_kanri.bat     1998/10/25 18:00
              eigyou    営業週報作成      syuhou.bat       1998/10/25 18:00
              kyuuyo    給与・賞与計算    kyuuyo.bat       1998/11/10 22:00
```

5.17 jobschprint -x(-X)コマンド

記述形式

```
jobschprint {-x|-X} [-long] [-sys サブシステム番号]  (注)
```

注)

“[-sys サブシステム番号]”はSystemwalker Operation Manager EEのオプションです。

機能説明

登録されているグループについて、現在の状態を出力します。

オプション

-x|-X

コマンドの実行ユーザが参照できるグループについて現在の状態を出力します。

-long

出力項目“Group”および“Group Comment”を最大値まで出力します。

EE GEE **-sys サブシステム番号**

複数サブシステム運用をしているシステムで、操作対象のサブシステムを指定するためのオプションです。操作対象とするサブシステム番号を、0から9の範囲で指定します。このオプションを省略した場合、サブシステム番号0が操作対象となります。

復帰値

復帰値	意味
0	正常終了
0以外	異常終了。エラーの内容はエラーメッセージを確認してください。

コマンド格納場所

Windows	Systemwalker Operation Managerインストール先ディレクトリ¥MpWalker.JM¥bin
Solaris HP-UX AIX Linux	/usr/bin

出力項目の説明

出力項目の説明を以下に示します。

User Name

jobschprintコマンドの実行ユーザ名です。

Project Name

プロジェクト名です。

Group

グループ名です。

先頭から8バイト分まで表示します。-longオプションを指定した場合は、50バイト分すべて表示します。

Group Comment

グループ名称です。

先頭から24バイト分まで表示します。-longオプションを指定した場合は、100バイト分すべて表示します。

Status

グループの状態です。以下に示した10種類の状態から選択表示されます。

Normal:

正常終了

Pseudo:

疑似正常

Abend:

異常終了

Forced:

強制終了

Execute:

実行中

Wait:

実行待ち

Pause:

停止中

Disable:

無効状態

Coped:

確認済

Warning:

警告

[ジョブスケジューラ起動パラメタの定義]ウィンドウで疑似正常を無効にしている場合は、“疑似正常”状態は“正常終了”状態として表示されることに注意してください。

Scheduled Time

グループの次の起動予定日時です。

Start Time

前回のグループが起動した時刻です。

End Time

前回のグループが終了した時刻です。

Completion Code

前回のグループの終了コードです。グループの終了コードはグループ内ジョブネットの終了コードの最大値です。

注意事項【UNIX版】

- UTF-8コードの環境では、グループ名およびグループ名称に日本語を使用している場合、出力項目の位置がずれて出力されることがあります。

使用例

```
jobschprint -x -sys 2
```

出力例

コマンドの出力結果“Group Status List”の例を以下に示します。

```
Group Status List          date: 2000/10/25  time: 17:27
```

```
User Name: suzuki

Project Name:総務部門

Group   Group Comment Status  Scheduled Time   Start Time        End Time     Completion Code
keiri   経理          Normal  2000/11/10       2000/10/10 18:00  2000/10/10 20:27 0
syuuhou 週報          Wait    2000/10/25 20:00
eigyou  営業          Wait    2000/10/25 23:00
```

5.18 jobschprint -y(-Y)コマンド

記述形式

jobschprint {-y|-Y} [-long] [-sys サブシステム番号]　(注)

注)

“[-sys サブシステム番号]”はSystemwalker Operation Manager EEのオプションです。

機能説明

登録されているグループ内のジョブネットの一覧を出力します。

オプション

-y|-Y

コマンドの実行ユーザが参照できるグループ内のジョブネットの一覧を出力します。

-long

出力項目“Group”および“Job Net”を最大値まで出力します。

EE GEE

-sys サブシステム番号

複数サブシステム運用をしているシステムで、操作対象のサブシステムを指定するためのオプションです。操作対象とするサブシステム番号を、0から9の範囲で指定します。このオプションを省略した場合、サブシステム番号0が操作対象となります。

復帰値

復帰値	意味
0	正常終了
0以外	異常終了。エラーの内容はエラーメッセージを確認してください。

コマンド格納場所

Windows	Systemwalker Operation Managerインストール先ディレクトリ¥MpWalker.JM¥bin
Solaris HP-UX AIX Linux	/usr/bin

出力項目の説明

出力項目の説明を以下に示します。

User Name

jobschprintコマンドの実行ユーザ名です。

Project Name

プロジェクト名です。

Group

グループ名です。

先頭から8バイト分まで表示します。-longオプションを指定した場合は、50バイト分すべて表示します。

Job Net

ジョブネット名です。

先頭から8バイト分まで表示します。-longオプションを指定した場合は、50バイト分すべて表示します。

Schedule

本日のスケジュールの有無を表示します。

YES:

本日、スケジュールされています。

NO:

本日、スケジュールされていません。

Status

ジョブネットの状態です。以下に示した11種類の状態から選択表示されます。

Normal:

正常終了

Pseudo:

疑似正常

Abend:

異常終了

Forced:

強制終了

Warning:

警告

Execute:

実行中

Wait:

実行待ち

Pause:

停止中

Disable:

無効状態

StartDelay:

開始遅延

EndDelay:

終了遅延

[ジョブスケジューラ起動パラメタの定義]ウィンドウで疑似正常を無効にしている場合は、“疑似正常”状態は“正常終了”状態として表示されることに注意してください。

Start Time

前回のジョブネットが起動した時刻です。

End Time

前回のジョブネットが終了した時刻です。

Completion Code

前回のジョブネットの終了コードです。

注意事項【UNIX版】

- UTF-8コードの環境では、グループ名およびジョブネット名に日本語を使用している場合、出力項目の位置がずれて出力されることがあります。

使用例

```
jobschprint -y -sys 2
```

出力例

コマンドの出力結果“Group List”の例を以下に示します。

```
  Group List              date: 1998/10/25   time: 17:28

User Name: suzuki

Project Name: 総務部門

Group   job Net  Schedule  Status  Start Time        End Time        Completion Code
kanri   kyuuyo   Yes       Normal  1998/10/25 15:23  1998/10/25 15:40  0
        kokyaku  Yes       Normal  1998/10/25 15:40  1998/10/25 16:26  0
        backup   Yes       Normal  1998/10/25 16:26  1998/10/25 16:37  0

syuuhou hanbai   No        Wait
        eigyou   No        Wait
```

5.19 jobschprint -bコマンド

記述形式

```
jobschprint -b [-sys サブシステム番号]  (注)
```

注)

“[-sys サブシステム番号]”はSystemwalker Operation Manager EEのオプションです。

機能説明

登録されている起動日雛形名の一覧を出力します。

オプション

-b

登録されている起動日雛形名の一覧を出力します。

EE GEE **-sys サブシステム番号**

複数サブシステム運用をしているシステムで、操作対象のサブシステムを指定するためのオプションです。操作対象とするサブシステム番号を、0から9の範囲で指定します。このオプションを省略した場合、サブシステム番号0が操作対象となります。

復帰値

復帰値	意味
0	正常終了
0以外	異常終了。エラーの内容はエラーメッセージを確認してください。

コマンド格納場所

Windows	Systemwalker Operation Managerインストール先ディレクトリ¥MpWalker.JM¥bin
Solaris HP-UX AIX Linux	/usr/bin

出力項目の説明

出力項目の説明を以下に示します。

Schedule Model

起動日雛形名です。

注意事項【UNIX版】

- UTF-8コードの環境では、起動日雛形名に日本語を使用している場合、出力項目の位置がずれて出力されることがあります。

出力例

コマンドの出力結果“Schedule Model List”の例を以下に示します。

```
Schedule Model List            date: 1998/10/25   time: 17:30

Schedule Model: 毎日
Schedule Model: 平日
Schedule Model: 毎週月曜日
Schedule Model: 毎週金曜日
Schedule Model: 毎月10日
```

5.20 jobschprint -cコマンド

記述形式

```
jobschprint -c 起動日雛形名 [-sys サブシステム番号]  (注)
```

注)

“[-sys サブシステム番号]”はSystemwalker Operation Manager EEのオプションです。

機能説明

起動日雛形として登録されている起動日のうち、日付指定により設定した起動日を出力します。

オプション

-c

起動日雛形として登録されている起動日のうち、日付指定により設定した起動日を出力します。

起動日雛形名

出力する起動日雛形名を指定します。

EE GEE

-sys サブシステム番号

複数サブシステム運用をしているシステムで、操作対象のサブシステムを指定するためのオプションです。操作対象とするサブシステム番号を、0から9の範囲で指定します。このオプションを省略した場合、サブシステム番号0が操作対象となります。

復帰値

復帰値	意味
0	正常終了
0以外	異常終了。エラーの内容はエラーメッセージを確認してください。

コマンド格納場所

Windows	Systemwalker Operation Managerインストール先ディレクトリ¥MpWalker.JM¥bin
Solaris HP-UX AIX Linux	/usr/bin

出力項目の説明

出力項目の説明を以下に示します。

表の縦軸は1月から12月までを表し、横軸は日付を表します。“*”が付いている日は起動日雛形として登録したジョブネットの起動日です。

Schedule Model

起動日雛形名を表示します。

出力例

コマンドの出力結果“Schedule Model Matrix”の例を以下に示します。

```
  Schedule Model Matrix             date: 1998/10/25 time: 17:06

Schedule Model: 毎月10日

-------------------------------------------------------------------
    | 0                 10                  20                  30
 MON | 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
-------------------------------------------------------------------
 1   |                   *
 2   |                   *
```

```
 3   |                  *
 4   |                  *
 5   |                  *
 6   |                  *
 7   |                  *
 8   |                  *
 9   |                  *
10   |                  *
11   |                  *
12   |                  *
-----------------------------------------------------------------
```

5.21　jobschprint -dコマンド

記述形式

jobschprint -d 起動日雛形名 [-sys サブシステム番号]　(注)

注)

“[-sys サブシステム番号]”はSystemwalker Operation Manager EEのオプションです。

機能説明

起動日雛形として登録されている起動日のうち、曜日指定により設定した起動日を出力します。

オプション

-d

起動日雛形として登録されている起動日のうち、曜日指定により設定した起動日を出力します。

起動日雛形名

出力する起動日雛形名を指定します。

EE GEE **-sys サブシステム番号**

複数サブシステム運用をしているシステムで、操作対象のサブシステムを指定するためのオプションです。操作対象とするサブシステム番号を、0から9の範囲で指定します。このオプションを省略した場合、サブシステム番号0が操作対象となります。

復帰値

復帰値	意味
0	正常終了
0以外	異常終了。エラーの内容はエラーメッセージを確認してください。

コマンド格納場所

Windows	Systemwalker Operation Managerインストール先ディレクトリ¥MpWalker.JM¥bin
Solaris HP-UX AIX Linux	/usr/bin

出力項目の説明

出力項目の説明を以下に示します。

表の縦軸は1月から12月までを表し、横軸は曜日を表します。“*”が付いている曜日は起動日雛形として登録したジョブネットの起動日です。

Schedule Model

起動日雛形名を表示します。

出力例

コマンドの出力結果“Schedule Model Matrix”の例を以下に示します。

```
  Schedule Model Matrix         date: 1998/10/25    time: 17:08

Schedule Model: 毎週金曜日

----------------------------------------
MON | Sun  Mon  Tue  Wed  Thu  Fri  Sat
----------------------------------------
 1  |                           *
 2  |                           *
 3  |                           *
 4  |                           *
 5  |                           *
 6  |                           *
 7  |                           *
 8  |                           *
 9  |                           *
10  |                           *
11  |                           *
12  |                           *
----------------------------------------
```

5.22 jobschprint -hコマンド

記述形式

```
jobschprint -h カレンダ名 [-sys サブシステム番号]  (注)
```

注)

“[-sys サブシステム番号]”はSystemwalker Operation Manager EEのオプションです。

機能説明

登録されているカレンダの休日を出力します。今年から3年分のカレンダを出力します。

オプション

-h

登録されている休日(今年から3年分の休日を表示したカレンダ)を出力します。

カレンダ名

出力するカレンダ名を指定します。

EE GEE **-sys サブシステム番号**

複数サブシステム運用をしているシステムで、操作対象のサブシステムを指定するためのオプションです。操作対象とするサブシステム番号を、0から9の範囲で指定します。カレンダ情報はすべてのジョブスケジューラで共通に使用される情報のため、このオプションを指定しても省略した場合と同じ情報が出力されます。

復帰値

復帰値	意味
0	正常終了
0以外	異常終了。エラーの内容はエラーメッセージを確認してください。

コマンド格納場所

Windows	Systemwalker Operation Managerインストール先ディレクトリ¥MpWalker.JM¥bin
Solaris HP-UX AIX Linux	/usr/bin

出力項目の説明

出力項目の説明を以下に示します。なお、休日は“*”が表示されます。

Calendar Name

カレンダ名です。

Day Change Time

日変わり時刻です。

出力例

コマンドの出力結果“Job Net Holiday List”の例を以下に示します。

```
  Job Net Holiday List           date: 1998/10/25  time: 17:09

Calendar Name : 日曜日

Day Change Time: 00:00

1998/01

 Sun  Mon  Tue  Wed  Thu  Fri  Sat
                     1    2    3

  4    5    6    7    8    9   10
  *
 11   12   13   14   15   16   17
  *
 18   19   20   21   22   23   24
  *
 25   26   27   28   29   30   31
  *

1998/02

 Sun  Mon  Tue  Wed  Thu  Fri  Sat
```

```
 1    2    3    4    5    6    7
 *
 8    9   10   11   12   13   14
 *
```

5.23 jobschprint -uコマンド

記述形式

jobschprint -u [-long] [-sys サブシステム番号]　(注)

注)

“[-sys サブシステム番号]”はSystemwalker Operation Manager EEのオプションです。

機能説明

現在、誰がどこの端末からジョブスケジューラの画面を利用しているかを知りたい場合、本コマンドを起動すれば、利用者情報として“ユーザ名”および“クライアント名(IPアドレス)”を出力します。

利用者情報の出力には、複数サーバ監視クライアントの接続情報も含まれます。

オプション

-u

ジョブスケジューラの利用者情報(ユーザ名、クライアント名(IPアドレス))を出力します。

-long

出力項目“User Name”を最大値まで出力します。

EE GEE **-sys サブシステム番号**

複数サブシステム運用をしているシステムで、操作対象のサブシステムを指定するためのオプションです。操作対象とするジョブスケジューラに対応するサブシステム番号を、0から9の範囲で指定します。このオプションを省略した場合、サブシステム番号0が操作対象となります。

復帰値

復帰値	意味
0	正常終了
0以外	異常終了。エラーの内容はエラーメッセージを確認してください。

コマンド格納場所

Windows	Systemwalker Operation Managerインストール先ディレクトリ¥MpWalker.JM¥bin
Solaris HP-UX AIX Linux	/usr/bin

出力項目の説明

出力項目の説明を以下に示します。

User Name

ジョブスケジューラ利用者のユーザ名です。

【Windows版】

-longオプション指定の有無に関わらず、20バイト分すべて表示します。

【UNIX版】

先頭から12バイト分まで表示します。

-longオプションを指定した場合は、20バイト分すべて表示します。

Client Name

端末のクライアント名とIPアドレスです。クライアント名がない場合は“???”が表示されます。

注意事項

以下のような操作・状況において、現在接続していないクライアントの情報が出力される場合があります。

- Systemwalker Operation Managerクライアントを強制終了した。
- Systemwalker Operation Managerクライアントが動作するPCを強制終了(電源切断)した。
- ネットワークの問題により、クライアント／サーバ間の接続が切断された。

使用例

```
jobschprint -u -sys 2
```

出力例

コマンドの出力結果“Jobsch User List”の例を以下に示します。

```
  Jobsch User List           date: 1998/10/25  time: 17:12

User Name     Client Name
suzuki        king(123.123.123.123)
sato          jack(123.123.123.125)
yamada        ???(123.123.123.124)
```

5.24 jobschprint -vコマンド

記述形式

```
jobschprint -v [-sys サブシステム番号]  (注)
```

注)

“[-sys サブシステム番号]”はSystemwalker Operation Manager EEのオプションです。

機能説明

登録されているカレンダ名の一覧を出力します。

オプション

-v

登録されているカレンダ名の一覧を出力します。

EE GEE **-sys サブシステム番号**

複数サブシステム運用をしているシステムで、操作対象のサブシステムを指定するためのオプションです。操作対象とするサブシステム番号を、0から9の範囲で指定します。カレンダ情報はすべてのジョブスケジューラで共通に使用される情報のため、このオプションを指定しても省略した場合と同じ情報が出力されます。

復帰値

復帰値	意味
0	正常終了
0以外	異常終了。エラーの内容はエラーメッセージを確認してください。

コマンド格納場所

Windows	Systemwalker Operation Managerインストール先ディレクトリ¥MpWalker.JM¥bin
Solaris HP-UX AIX Linux	/usr/bin

出力項目の説明

出力項目の説明を以下に示します。

Calendar Name

カレンダ名です。

出力例

コマンドの出力結果“Holiday Calendar List”の例を以下に示します。

```
  Holiday Calendar List       date: 1998/10/25  time: 17:08

Calendar Name: 月曜日
Calendar Name: 火曜日
Calendar Name: 西営業所
Calendar Name: 南支店
```

5.25　jobschprint -o(-O)コマンド

記述形式

```
jobschprint {-o|-O} [-long] [-sys サブシステム番号]　(注)
```

注)

“[-sys サブシステム番号]”はSystemwalker Operation Manager EEのオプションです。

機能説明

登録されているプロジェクト名の一覧を出力します。

オプション

-o|-O

コマンドの実行ユーザが参照権以上の権限を持っているプロジェクトの一覧を出力します。

-long

出力項目“Project Name”を最大値まで出力します。

EE GEE **-sys サブシステム番号**

複数サブシステム運用をしているシステムで、操作対象のサブシステムを指定するためのオプションです。操作対象とするサブシステム番号を、0から9の範囲で指定します。このオプションを省略した場合、サブシステム番号0が操作対象となります。

復帰値

復帰値	意味
0	正常終了
0以外	異常終了。エラーの内容はエラーメッセージを確認してください。

コマンド格納場所

Windows	Systemwalker Operation Managerインストール先ディレクトリ¥MpWalker.JM¥bin
Solaris HP-UX AIX Linux	/usr/bin

出力項目の説明

出力項目の説明を以下に示します。

User Name

jobschprintコマンドの実行ユーザ名です。

Project Name

プロジェクト名です。

先頭から20バイト分まで表示します。-longオプションを指定した場合は、50バイト分すべて表示します。

Access

コマンド実行ユーザのプロジェクトに対するアクセス権を表示します。以下に示した4種類の状態から選択表示します。

all:

更新権
(または、登録権、操作権、参照権をすべて持つ場合)

change:

登録権

operate:

操作権

monitor:

参照権

Owner Name

プロジェクトの所有者名です。

注意事項【UNIX版】

- UTF-8コードの環境では、プロジェクト名に日本語を使用している場合、出力項目の位置がずれて出力されることがあります。

使用例

```
jobschprint -o -sys 2
```

出力例

コマンドの出力結果“Jobsch Project List”の例を以下に示します。

```
  Jobsch Project List         date: 2006/10/10  time: 09:22

User Name: suzuki

Project Name      Access     Owner Name
総務部門          all        ishikawa
営業部門          change     nakamura
企画部門          operate    hayashi
開発部門          monitor    sato
```

5.26　jobschprint -rhコマンド

記述形式

```
jobschprint -rh カレンダ名 [-sys サブシステム番号]　(注)
```

注)

“[-sys サブシステム番号]”はSystemwalker Operation Manager EEのオプションです。

機能説明

カレンダの休日情報を制御文の形式で出力します。

オプション

-rh

登録されているカレンダをjobschsetholコマンドで指定する休日情報制御文の記述形式で出力します。

カレンダ名

出力するカレンダ名を指定します。

EE GEE **-sys サブシステム番号**

複数サブシステム運用をしているシステムで、操作対象のサブシステムを指定するためのオプションです。操作対象とするサブシステム番号を、0から9の範囲で指定します。このオプションを省略した場合、サブシステム番号0が操作対象となります。

復帰値

復帰値	意味
0	正常終了

復帰値	意味
0以外	異常終了。エラーの内容はエラーメッセージを確認してください。

コマンド格納場所

Windows	Systemwalker Operation Managerインストール先ディレクトリ¥MpWalker.JM¥bin
Solaris HP-UX AIX Linux	/usr/bin

出力項目の説明

出力項目については、“休日情報制御文”を参照してください。

注意事項

「年間の休日設定」（[年間休日の設定]ウィンドウ）の情報は出力されません。

出力例

コマンドの出力結果の例を以下に示します。

```
HOL
calendarname 本社カレンダ
yearset 2010
monthset 11
dayset 1
yearset 2010
monthset 12
dayset 20
;
```

5.27　jobschprint -rspコマンド

記述形式

```
jobschprint -rsp 起動日雛形名 [-sys サブシステム番号]　(注)
```

注)

“[-sys サブシステム番号]”はSystemwalker Operation Manager EEのオプションです。

機能説明

起動日雛形情報を制御文の形式で出力します。

オプション

-rsp

登録されている起動日雛形をjobschsetcalコマンドで指定する起動日雛形制御文の記述形式で出力します。

起動日雛形名

出力する起動日雛形名を指定します。

EE GEE **-sys サブシステム番号**

複数サブシステム運用をしているシステムで、操作対象のサブシステムを指定するためのオプションです。操作対象とするサブシステム番号を、0から9の範囲で指定します。このオプションを省略した場合、サブシステム番号0が操作対象となります。

復帰値

復帰値	意味
0	正常終了
0以外	異常終了。エラーの内容はエラーメッセージを確認してください。

コマンド格納場所

Windows	Systemwalker Operation Managerインストール先ディレクトリ¥MpWalker.JM¥bin
Solaris HP-UX AIX Linux	/usr/bin

出力項目の説明

出力項目については、“起動日雛形制御文”を参照してください。

出力例

コマンドの出力結果の例を以下に示します。

```
CAL
execpatname サンプル
monthset 1
dayset 1,5,10,15,20
weekset MON,FRI
monthset 2
dayset 1,4,8,12,16
weekset MON,THU
monthset 5
dayset 1,10,20,30
weekset TUE,THU
monthset 11
dayset 1,11,22,30
weekset WED,THU
;
```

5.28 jobschprintcsv -nコマンド

記述形式

jobschprintcsv -n [-toplevel] [-title] [-sys サブシステム番号]　(注)

注)

“[-sys サブシステム番号]”はSystemwalker Operation Manager EEのオプションです。

機能説明

ジョブネットの現在の状態をCSV形式で、標準出力に出力します。

出力結果をファイルに保存すると、ジョブネットの現在の状態を表計算ソフトで管理することができます。(注)

注)

表計算ソフトに取り込む前に、出力結果を表計算ソフトがサポートする文字コードに変換しておく必要があります。

オプション

-n

コマンドの実行ユーザが参照できるジョブネットについて現在の状態を出力します。

-toplevel

子ジョブネット、およびリンクジョブネットを出力しません。階層化されていないジョブネットおよび階層化されているジョブネットのうち第1階層のものを出力します。

-title

指定した場合、1行目にタイトル行が出力されます。

EE GEE **-sys サブシステム番号**

複数サブシステム運用をしているシステムで、操作対象のサブシステムを指定するためのオプションです。操作対象とするサブシステム番号を、0から9の範囲で指定します。このオプションを省略した場合、サブシステム番号0が操作対象となります。

復帰値

復帰値	意味
0	正常終了
0以外	異常終了。エラーの内容はエラーメッセージを確認してください。

コマンド格納場所

Windows	Systemwalker Operation Managerインストール先ディレクトリ¥MpWalker.JM¥bin
Solaris HP-UX AIX Linux	/usr/bin

出力項目の説明

出力項目について説明します。

Host Name

本コマンドを実行したサーバのホスト名です。

EE GEE **Sub System**

操作対象のジョブスケジューラに対応するサブシステム番号です。

Project Name

ジョブネットが登録されているプロジェクト名です。

Job Net

ジョブスケジューラに登録されているジョブネット名です。

Job Net Comment

ジョブスケジューラに登録されているジョブネット名称です。

Status

ジョブネットの状態です。

以下に示した13種類の状態から選択表示されます。

Normal:

正常終了

Pseudo:

疑似正常

Abend:

異常終了

Forced:

強制終了

Warning:

警告

Execute:

実行中

Wait:

実行待ち

Pause:

停止中

Disable:

無効状態

Coped:

確認済

CarryOver:

持ち越し

StartDelay:

開始遅延

EndDelay:

終了遅延

[ジョブスケジューラ起動パラメタの定義]ウィンドウで疑似正常を無効にしている場合は、“疑似正常”状態は“正常終了”状態として表示されることに注意してください。

Scheduled Time

起動予定時刻です。ジョブネットの状態によっては表示されません。

Start Time

前回のジョブネット開始時刻です。ジョブネットの状態によっては表示されません。

End Time

前回のジョブネット終了時刻です。ジョブネットの状態によっては表示されません。

Completion Code

ジョブネットの終了コードです。ジョブネットの状態によっては表示されません。

注意事項

- マスタスケジュール管理機能を有効にしている場合は、stemprint -n(-N)コマンドでジョブネットの現在の状態を出力してください。stemprint -n(-N)コマンドを利用することで、状態を出力したい運用日やプロジェクトを絞り込むことができます。stemprint -n(-N)コマンドの詳細は、“6.12 stemprint -n(-N)コマンド”を参照してください。

出力例

コマンドの出力結果の例を以下に示します。(EE版またはGEE版の例)

```
Host Name,Sub System,Project Name,Job Net,Job Net
Comment,Status,Scheduled Time,Start Time,End Time,Completion Code
host1,3,営業,hanbai,販売管理,Abend,,2014/03/01 10:22,2014/10/25 10:38,10
host1,3,営業,kokyaku,顧客売上管理,Wait,2014/03/01 18:00,,,
host1,3,営業,eigyou,営業週報作成,Disable,2014/03/02,2015/02/26
23:00,2015/02/26 23:12,
host1,3,総務,suitou,出納データ管理,Execute,2014/03/05,2015/03/01 14:00,,
host1,3,総務,kyuuyo,給与・賞与計算,Normal,2014/03/20,2015/02/20
23:00,2015/02/20 23:50,0
```

上記のファイルを表計算ソフトで表示すると、以下のように表示されます。

Host Name	Sub System	Project Name	Job Net	Job Net Comment	Status	Scheduled Time
host1	3	営業	hanbai	販売管理	Abend	
host1	3	営業	kokyaku	顧客売上管理	Wait	2014/03/01 18:00
host1	3	営業	eigyou	営業週報作成	Disable	2014/03/02
host1	3	総務	suitou	出納データ管理	Execute	2014/03/05
host1	3	総務	kyuuyo	給与・賞与計算	Normal	2014/03/20

5.29 jobschprintcsv -gコマンド

記述形式

jobschprintcsv -g *YYYY MM DD* [-title] [-vparam] [-sys サブシステム番号]　(注)

注)

“[-sys サブシステム番号]”はSystemwalker Operation Manager EEのオプションです。

機能概要

ジョブネットの実行結果の履歴情報を、CSV形式で標準出力に出力します。

出力結果をファイルに保存すると、ジョブネットの実行結果の履歴情報を表計算ソフトで管理することができます。(注)

注)

表計算ソフトに取り込む前に、出力結果を表計算ソフトがサポートする文字コードに変換しておく必要があります。

オプション

-g

コマンドの実行ユーザが参照できるジョブネットについて履歴情報を出力します。

YYYY

西暦を4桁の数字で指定します。

MM

出力したい日の月を2桁以内の数字で指定します。

DD

出力したい日を2桁以内の数字で指定します。

-title

指定した場合、1行目にタイトル行が出力されます。

-vparam

可変パラメタを指定して起動されたジョブネットの場合、可変パラメタを表示します。

EE GEE

-sys サブシステム番号

複数サブシステム運用をしているシステムで、操作対象のサブシステムを指定するためのオプションです。操作対象とするサブシステム番号を、0から9の範囲で指定します。このオプションを省略した場合、サブシステム番号0が操作対象となります。

復帰値

復帰値	意味
0	正常終了
0以外	異常終了。エラーの内容はエラーメッセージを確認してください。

コマンド格納場所

Windows	Systemwalker Operation Managerインストール先ディレクトリ¥MpWalker.JM¥bin

Solaris HP-UX AIX Linux	/usr/bin

出力項目の説明

出力項目について説明します。

Host Name

本コマンドを実行したサーバのホスト名です。

EE GEE **Sub System**

操作対象のジョブスケジューラに対応するサブシステム番号です。

Project Name

ジョブネットが登録されているプロジェクト名です。

Job Net

ジョブスケジューラに登録されているジョブネット名です。

Job Net Comment

ジョブスケジューラに登録されているジョブネット名称です。

Status

ジョブネットの状態です。

以下に示した8種類の状態から選択表示されます。

Normal:

正常終了

Pseudo:

疑似正常

Abend:

異常終了(終了コード:正常終了コード上限値+1～255)

Forced:

強制終了(終了コード:256)

Execute:

実行中

E_Cancel:

実行拒否

Closed:

実行中断(終了コード:239)

Skipped:

スキップ

Start Time

ジョブネットが起動した時刻です。“西暦/月/日 時:分:秒”の形式で表示します。

End Time

ジョブネットが終了した時刻です。“西暦/月/日 時:分:秒”の形式で表示します。ジョブネットの状態によっては表示されません。

End Code

ジョブネットの終了コードです。ジョブネットの状態によっては表示されません。

Variable Parameter

可変パラメタを指定して起動されたジョブネットの可変パラメタです。-vparamオプションを指定した場合のみ表示されます。ジョブネットの状態によっては表示されません。

注意事項

- サイクリックに管理されているジョブスケジューラのログファイル(jobdb1.log/jobdb2.log/jobdb3.log)のうち、現在使用中のログファイルと1世代前のログファイルの履歴情報を出力します。
- マスタスケジュール管理機能を有効にしている場合は、stemprint -g(-G)コマンドで履歴情報を出力してください。本コマンドの場合、マスタスケジュール管理機能で指定した[スケジュール情報を保管する世代数]を超えて管理対象外となった運用日の履歴情報は出力できません。stemprint -g(-G)コマンドの詳細は、“6.11 stemprint -f(-F)/-g(-G)コマンド”を参照してください。

出力例

コマンドの出力結果の例を以下に示します。(EE版またはGEE版の例)

```
Host Name,Sub System,Project Name,Job Net,Job Net Comment,Status,Start Time,End Time,End Code, Variable Parameter
host1,3,総務,kyuuyo,給与・賞与計算,Normal,2014/03/01 10:00:01,2014/03/01 11:33:32,0,"200707"
host1,3,営業,backup,バックアップ業務,Normal,2014/03/01 10:30:01,2014/03/01 11:49:26,0,
host1,3,営業,uriage,売上データ管理,Normal,2014/03/01 12:48:42,2014/03/01 12:50:56,0,"sales01.dat"
host1,3,営業,kanri,顧客管理,Abend,2014/03/01 13:32:45,2014/03/01 13:42:37,2,
host1,3,総務,data,出納データ管理,Forced,2014/03/01 16:27:21,2014/03/01 16:31:21,256,
host1,3,総務,data,出納データ管理,Execute,2014/03/01 16:44:30,,,
```

上記のファイルを表計算ソフトで表示すると、以下のように表示されます。

Host Name	Sub System	Project Name	Job Net	Job Net Comment	Status	Start Time
host1	3	総務	kyuuyo	給与・賞与計算	Normal	2014/03/01 10:
host1	3	営業	backup	バックアップ業務	Normal	2014/03/01 10:3
host1	3	営業	uriage	売上データ管理	Normal	2014/03/01 12:48
host1	3	営業	kanri	顧客管理	Abend	2014/03/01 13:32:
host1	3	総務	data	出納データ管理	Forced	2014/03/01 16:27:2
host1	3	総務	data	出納データ管理	Execute	2014/03/01 16:44:

5.30 jobschprintcsv -lコマンド

記述形式

jobschprintcsv -l *YYYY MM* [-title] [-sys サブシステム番号]　(注)

注)

“[-sys サブシステム番号]”はSystemwalker Operation Manager EEのオプションです。

機能説明

ジョブネットの運用予定一覧を、CSV形式で標準出力に出力します。

出力結果をファイルに保存すると、ジョブネットの運用予定一覧を表計算ソフトで管理することができます。(注)

注)

表計算ソフトに取り込む前に、出力結果を表計算ソフトがサポートする文字コードに変換しておく必要があります。

オプション

-l

コマンドの実行ユーザが参照できるジョブネットについて、指定した年月の運用予定一覧を出力します。

複写起動されたジョブネットは出力されません。

YYYY

西暦を4桁の数字で指定します。指定できる年は、今年、来年および再来年です。年変わり直後であっても昨年は指定できません。

MM

出力したい日の月を2桁以内の数字で指定します。

-title

指定した場合、1行目にタイトル行が出力されます。

EE GEE **-sys サブシステム番号**

複数サブシステム運用をしているシステムで、操作対象のサブシステムを指定するためのオプションです。操作対象とするサブシステム番号を、0から9の範囲で指定します。このオプションを省略した場合、サブシステム番号0が操作対象となります。

復帰値

復帰値	意味
0	正常終了
0以外	異常終了。エラーの内容はエラーメッセージを確認してください。

コマンド格納場所

Windows	Systemwalker Operation Managerインストール先ディレクトリ¥MpWalker.JM¥bin
Solaris HP-UX AIX Linux	/usr/bin

出力項目の説明

出力項目について説明します。

数字は日付を表します。“*”が付いている日はジョブネットの起動日、半角スペースは非起動日で、それぞれ“,”(コンマ)で区切って出力します。なお、指定した年月の最初の1日は、区切りを表す半角スペースの後に、ジョブネットの起動日の場合は“*”、非起動日の場合は半角スペースを出力します。

Host Name

本コマンドを実行したサーバのホスト名です。

EE GEE **Sub System**

操作対象のジョブスケジューラに対応するサブシステム番号です。

Project Name

ジョブネットが登録されているプロジェクト名です。

Year/Month

-lオプションで指定された日付です。

Job Net

ジョブスケジューラに登録されているジョブネット名です。

Job Net Comment

ジョブスケジューラに登録されているジョブネット名称です。

出力例

コマンドの出力結果の例を以下に示します。(EE版またはGEE版の例)

```
Host Name,Sub System,Project Name,Year/Month,Job Net,Job Net Comment,1,2,3,4,5,6,7,8,9,
10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31
 host1,3,営業,2014/3,hanbai,販売管理,  ,  ,  ,*,  ,  ,  ,  ,  ,  ,  ,*,  ,  ,  ,  ,  ,  ,  ,*,  ,  ,  ,  ,
 host1,3,営業,2014/3,kokyaku,顧客売上管理,  ,  ,*,  ,  ,  ,  ,  ,  ,  ,*,  ,  ,  ,  ,  ,  ,  ,*,  ,  ,  ,  ,  ,
 host1,3,営業,2014/3,eigyou,営業週報作成, *,  ,  ,  ,  ,  ,*,  ,  ,  ,  ,  ,  ,*,  ,  ,  ,  ,  ,  ,  ,*,  ,  ,  ,  ,  ,*,  ,
```

上記のファイルを表計算ソフトで表示すると、以下のように表示されます。

Host Name	Sub System	Project Name	Year/Month	Job Net	Job Net Comment	1	2	3	4
host1	3	営業	2014/3	hanbai	販売管理				*
host1	3	営業	2014/3	kokyaku	顧客売上管理			*	
host1	3	営業	2014/3	eigyou	営業週報作成	*			

5.31 jobschprintcsv -fコマンド

記述形式

```
jobschprintcsv -f P [-title] [-vparam] [-sys サブシステム番号]  (注)
```

注)

“[-sys サブシステム番号]”はSystemwalker Operation Manager EEのオプションです。

機能概要

ジョブネットの実行結果の履歴情報を、CSV形式で標準出力に出力します。

出力結果をファイルに保存すると、ジョブネットの実行結果の履歴情報を表計算ソフトで管理することができます。(注)

注)

表計算ソフトに取り込む前に、出力結果を表計算ソフトがサポートする文字コードに変換しておく必要があります。

オプション

-f

コマンドの実行ユーザが参照できるジョブネットについて履歴情報を出力します。

-title

指定した場合、1行目にタイトル行が出力されます。

-vparam

可変パラメタを指定して起動されたジョブネットの場合、可変パラメタを表示します。

P

何日前の履歴情報を出力するかを0から5までの数字で指定します。

EE GEE **-sys サブシステム番号**

複数サブシステム運用をしているシステムで、操作対象のサブシステムを指定するためのオプションです。操作対象とするサブシステム番号を、0から9の範囲で指定します。このオプションを省略した場合、サブシステム番号0が操作対象となります。

復帰値

復帰値	意味
0	正常終了
0以外	異常終了。エラーの内容はエラーメッセージを確認してください。

コマンド格納場所

Windows	Systemwalker Operation Managerインストール先ディレクトリ¥MpWalker.JM¥bin
Solaris HP-UX AIX Linux	/usr/bin

出力項目の説明

出力項目について説明します。

Host Name

本コマンドを実行したサーバのホスト名です。

EE GEE **Sub System**

操作対象のジョブスケジューラに対応するサブシステム番号です。

Project Name

ジョブネットが登録されているプロジェクト名です。

Job Net

ジョブスケジューラに登録されているジョブネット名です。

Job Net Comment

ジョブスケジューラに登録されているジョブネット名称です。

Status

ジョブネットの状態です。

以下に示した8種類の状態から選択表示されます。

Normal:

正常終了

Pseudo:

疑似正常

Abend:

異常終了(終了コード:正常終了コード上限値+1～255)

Forced:

強制終了(終了コード:256)

Execute:

実行中

E_Cancel:

実行拒否

Closed:

実行中断(終了コード:239)

Skipped:

スキップ

Start Time

ジョブネットが起動した時刻です。“西暦/月/日 時:分:秒”の形式で表示します。

End Time

ジョブネットが終了した時刻です。“西暦/月/日 時:分:秒”の形式で表示します。ジョブネットの状態によっては表示されません。

End Code

ジョブネットの終了コードです。ジョブネットの状態によっては表示されません。

Variable Parameter

可変パラメタを指定して起動されたジョブネットの可変パラメタです。-vparamオプションを指定した場合のみ表示されます。ジョブネットの状態によっては表示されません。

注意事項

- マスタスケジュール管理機能を有効にしている場合は、stemprint -f(-F)コマンドで履歴情報を出力してください。本コマンドの場合、マスタスケジュール管理機能で指定した[スケジュール情報を保管する世代数]を超えて管理対象外となった運用日の履歴情報は出力できません。stemprint -f(-F)コマンドの詳細は、“6.11 stemprint -f(-F)/-g(-G)コマンド”を参照してください。

出力例

コマンドの出力結果の例を以下に示します。(EE版またはGEE版の例)

```
Host Name,Sub System,Project Name,Job Net,Job Net Comment,Status,Start Time,End Time,End Code, Variable Parameter
host1,3,総務,kyuuyo,給与・賞与計算,Normal,2001/03/01 10:00:01,2001/03/01 11:33:32,0,"200707"
host1,3,営業,backup,バックアップ業務,Normal,2001/03/01 10:30:01,2001/03/01 11:49:26,0,
host1,3,営業,uriage,売上データ管理,Normal,2001/03/01 12:48:42,2001/03/01 12:50:56,0,"sales01.dat"
host1,3,営業,kanri,顧客管理,Abend,2001/03/01 13:32:45,2001/03/01 13:42:37,2,
host1,3,総務,data,出納データ管理,Forced,2001/03/01 16:27:21,2001/03/01 16:31:21,256,
host1,3,総務,data,出納データ管理,Execute,2001/03/01 16:44:30,,,
```

上記のファイルを表計算ソフトで表示すると、以下のように表示されます。

Host Name	Sub System	Project Name	Job Net	Job Net Comment	Status	Start Time
host1	3	総務	kyuuyo	給与・賞与計算	Normal	2014/03/01 10:
host1	3	営業	backup	バックアップ業務	Normal	2014/03/01 10:3
host1	3	営業	uriage	売上データ管理	Normal	2014/03/01 12:48
host1	3	営業	kanri	顧客管理	Abend	2014/03/01 13:32:
host1	3	総務	data	出納データ管理	Forced	2014/03/01 16:27:2
host1	3	総務	data	出納データ管理	Execute	2014/03/01 16:44:

5.32 stemprintprojectコマンド

記述形式

```
stemprintproject プロジェクト名 [-sys サブシステム番号]（注）
```

注)

“[-sys サブシステム番号]”はSystemwalker Operation Manager EEのオプションです。

機能概要

プロジェクト情報を制御文の形式で出力します。

オプション

プロジェクト名

出力対象のプロジェクト名を指定します。

EE GEE

-sys サブシステム番号

複数サブシステム運用をしているシステムで、操作対象のサブシステムを指定するためのオプションです。出力対象のプロジェクトが存在するサブシステム番号を0から9の範囲で指定します。このオプションを省略した場合、サブシステム番号0が対象となります。

復帰値

復帰値	意味
0	正常終了
1以上	異常終了

コマンド格納場所

Windows	Systemwalker Operation Managerインストール先ディレクトリ¥MPWALKER.JM¥bin
Solaris Linux	/opt/FJSVstem/bin

実行に必要な権限

システム管理者(Administratorsグループに属するユーザ/スーパーユーザ)権限が必要です。

注意事項

プロジェクトを別の環境へ移行する場合は、プロジェクト情報制御文に出力されるユーザーIDやグループ名が移行先のシステムに存在する必要があります。

使用例

```
stemprintproject PRJ001 -sys 2
```

出力例

```
PRJ
projectname PRJ001
username root
manage_user stem04
operate_user guest,stem01
monitor_user testuser,stem02
;
```

EE GEE

第6章 マスタスケジュール管理コマンド

本章では、Systemwalker Operation Manager マスタスケジュール管理機能で使用するコマンドについて説明します。

注意

WindowsのUAC機能を有効にしている環境でコマンドを実行する場合、OSにログインしたユーザの権限によって権限の昇格を求めるダイアログが表示され、対処が必要となります。

詳細は、“Systemwalker Operation Manager 運用ガイド”の、“Windowsでコマンドを実行する場合の注意事項【Windows版】”を参照してください。

6.1 コマンド一覧

マスタスケジュール管理機能で使用するコマンドには以下のものがあります。

stemenvset 環境構築コマンド

マスタスケジュール管理の環境を、構築・解除します。

stemManager スケジュール管理コマンド

日ごとスケジュール管理を実行・管理します。

stemSetPrjMaster プロジェクトマスタ設定コマンド

登録されているプロジェクトを日ごとスケジュール管理のマスタ定義として設定します。

stemCopyProject 運用変更事前設定コマンド

運用変更をするためのプロジェクトを作成します。(プロジェクトの複写)

stemMoveProject 運用変更リリース日変更コマンド

運用変更用に作成されたプロジェクトのリリース日を変更します。(プロジェクトの移動)

stemBackProject 運用変更取消コマンド

運用変更されたマスタ情報をリリース日前の状態に戻します。

jobschmsgeventex メッセージ事象発生コマンド

運用日を意識したメッセージ事象を発生させます。

stemprintnet 退避情報確認コマンド

退避したスケジュール情報を表示します

stemConfig マスタスケジュール管理環境設定コマンド【UNIX版】

クラスタ環境下で、管理サーバおよびスケジュールサーバのIPアドレスを変更します。

stemprint -f(-F)/-g(-G)コマンド

マスタスケジュール管理機能を有効にしている場合の、ジョブネットの履歴情報を出力します。

stemprint -n(-N)コマンド

マスタスケジュール管理機能を有効にしている場合の、ジョブネットの現在の状態を出力します。

stemschnet ジョブネット投入コマンド

すでに作成済みの運用日管理プロジェクトへマスタ情報からジョブネットを個別に投入(抽出・登録)します。

stemsetdbpath クラスタ用DBディレクトリ変更コマンド【Windows版】

マスタスケジュール管理のデータベースディレクトリを変更します。

stemregservice クラスタ用サービス登録コマンド【Windows版】

マスタスケジュール管理機能のクラスタ用サービスの登録/削除を行います。クラスタ環境構築において、マスタスケジュール管理機能を使用する場合に使用します。

ポイント

拡張ユーザ管理機能が有効な場合は、以下のコマンドを実行したOSユーザがシステム管理者かどうかでアクセス権の認証が変わります。

a. 実行したOSユーザがシステム管理者の場合は、Operation Managerの管理者の権限(すべてのプロジェクトに対して更新権あり)で実行されます。

b. 実行したOSユーザがプロジェクトの所有者の場合は、所有者となっているプロジェクトに対して更新権を持っているものとして実行されます。

c. システム管理者でない場合、かつ、プロジェクトの所有者でない場合は、プロジェクトにアクセス権を持つすべてのOperation Managerユーザに対応するOSユーザの中に、コマンドを実行したOSユーザが含まれ、権限があれば、コマンドは実行されます。コマンドを実行したOSユーザが複数のOperation Managerユーザに対応づけられている場合、OSユーザは、対応づけられた複数のOperation Managerユーザのアクセス権の中で一番強い権限(更新権＞登録権・操作権＞参照権)を持ちます。なお、OSユーザに対応づけられたOperation Managerユーザに、登録権と操作権が設定されている場合、OSユーザは、両方の権限を持ちます。

- jobschmsgeventexコマンド
- stemprintコマンド

6.2 stemenvset 環境構築コマンド

記述形式

【Windows版】

```
stemenvset.exe [-set | -u]
```

【UNIX版】

```
stemenvset.sh [-set | -u]
```

機能説明

Systemwalker Operation Manager マスタスケジュール管理環境の構築・解除を行います。

オプション

下記オプション省略時は、マスタスケジュール管理機能の有効/無効を表示します。

-set

Systemwalker Operation Manager マスタスケジュール管理の環境を構築します。本オプションでのコマンド実行後、マスタスケジュール管理機能が有効となります。

-u

マスタスケジュール管理環境を削除します。複数のサブシステムが存在する場合は、すべてのサブシステムのマスタスケジュール管理環境を削除します。

復帰値

復帰値	意味
0	正常終了
0以外	異常終了

コマンド格納場所

Windows	Systemwalker Operation Managerインストールディレクトリ¥mpwalker.jm¥bin
Solaris HP-UX AIX Linux	/opt/FJSVstem/bin

実行に必要な権限

システム管理者(Administratorsグループに属するユーザ/スーパーユーザ)の権限が必要です。

注意事項

- スケジュール情報の分散中は、解除することはできません。
- マスタスケジュール管理機能を解除したとき、スケジュール状況の情報はすべて破棄されます。

実行結果／出力形式

正常終了 : なし

異常終了 : エラー内容は標準出力に出力されます。

使用例

本コマンドの使用例については、“Systemwalker Operation Manager マスタスケジュール管理 ユーザーズガイド”を参照してください。

6.3 stemManager スケジュール管理コマンド

記述形式

【Windows版】

```
stemManager [[-d yyyymmdd] [-force｜-del｜-chg]｜-n] [-sys サブシステム番号]
```

【UNIX版】

```
stemManager [[-d yyyymmdd] [[-force] [-h schedule_server]｜ -del ｜ -chg] ｜ -n] [-sys サブシステム番号]
```

機能説明

マスタ情報からスケジュール情報を抽出し、適用します。また、スケジュールの取り消し、再スケジュール、未来のスケジュール、運用変更を行います。

オプション

-d *yyyymmdd*

yyyymmdd で指定された日付のスケジュールを行います。このオプションは、以下の場合に指定します。

・ 自動起動でスケジュールに失敗した場合のリカバリ

・ 事前に未来のスケジュールを登録する場合

・ スケジュール実行がされずに過去となった運用日のスケジュールを登録する場合

-n

ジョブスケジューラのジョブとして登録し、毎回、翌日スケジュールを作成するためのオプションです。

-dおよび-nオプションを省略した場合は、当日のスケジュールを実行します。

-force

すでに抽出・適用されたスケジュール情報を新たに抽出したスケジュール情報で置き換える場合に指定します(再スケジュールです)。このオプションは、以下の場合に指定します。

・ 未来の日付でスケジュールした後に、マスタ情報が変更された場合

・ 当日のスケジュール状況が「なし」のときに、マスタ情報が変更された場合

すでにスケジュール情報が作成されている当日および過去の日付に対しては実行できません。

-h schedule_server【UNIX版】

指定したスケジュールサーバに対し、スケジュールのリカバリまたは再スケジュールを行います。新規にスケジュールするとき、スケジュールを取り消すときは指定できません。

-del

登録されたスケジュールを取り消します。

-chg

運用変更のみを行います。

-sys サブシステム番号

スケジュールを行いたいサブシステム番号を0から9の範囲で指定します。

省略時はサブシステム0が対象になります。ただし、サブシステム運用しているジョブスケジューラのジョブとして起動された場合は、起動元のサブシステム番号が設定されます。

全体のオプションを省略した場合は、当日のスケジュールを作成します。

復帰値

復帰値	意味
0	正常終了
0以外	異常終了

コマンド格納場所

Windows	Systemwalker Operation Managerインストールディレクトリ¥mpwalker.jm¥bin
Solaris HP-UX AIX Linux	/opt/FJSVstem/bin

実行に必要な権限

システム管理者(Administratorsグループに属するユーザ/スーパーユーザ)の権限が必要です。

注意事項

- 対象のスケジュールがすでに動作している場合は、そのスケジュールに対して作成、変更、削除はできません。
- スケジュールの実行が行える過去の日付は、スケジュール状況管理日数までです。
- 過去スケジュールで登録されたジョブネットに対し、以下の注意事項があります。
  - ジョブネットの状態にかかわらずプロジェクト数の上限値(1000)を超えた場合は、退避対象となります。退避後は、Systemwalker Operation Managerクライアントでジョブネットを操作することができません。登録されたジョブネットは、速やかに実行、終了させてください。
  - 持ち越し制御の対象外になります。
  - 時刻によるスケジュールが行われないため、時刻による起動条件が設定されている場合は、手動で起動する必要があります。
  - メッセージ事象による連携は可能ですが、jobschmsgeventexコマンドの-d、-dx、-sまたは-nオプションを使用して運用日を意識したメッセージ事象の発生を行う必要があります。

    (運用日を意識しないメッセージ事象発生では意図しないメッセージにより、ジョブネットが起動する場合があります。)
  - 未実行のままでも完了済として扱われ、[マスタスケジュール管理環境設定]ダイアログボックスで[ジョブネットが未完了の場合は削除しない]が設定されていても、保管する世代数を超えて管理対象外となった時点で削除されます。
- 旧バージョンのスケジュールサーバへスケジュール情報の分散を行った場合、利用できない機能があります。詳細は、“Systemwalker Operation Manager マスタスケジュール管理 ユーザーズガイド”の“管理サーバ・スケジュールサーバ接続のサポート範囲”を参照してください。

実行結果／出力形式

正常終了 : なし

異常終了 : エラー内容は標準出力およびイベントログ/SYSLOGに出力されます。-force、-del指定時は、標準出力のみに出力されます。

6.4 stemSetPrjMaster プロジェクトマスタ設定コマンド

記述形式

【Windows版】

```
stemSetPrjMaster.exe {set | del} -p プロジェクト名 [-sys サブシステム番号]
```

【UNIX版】

```
stemSetPrjMaster.sh {set | del} -p プロジェクト名 [-sys サブシステム番号]
```

機能説明

Systemwalker Operation Managerに登録したプロジェクトを日ごとスケジュール管理のマスタ情報として扱うように設定します。本コマンドで指定されたプロジェクトは、マスタ定義として、日ごとスケジュール管理の抽出・適用の対象となります。

本コマンドで指定されたプロジェクトは、マスタ定義として扱われるため、マスタDB上でのスケジュールはされません。

オプション

set

指定したプロジェクトに対し、マスタ設定します。マスタ設定されたプロジェクトのスケジュールは抑止されます。

del

指定したプロジェクトに対し、マスタ設定を解除します。

-p プロジェクト名

マスタ設定するプロジェクト、またはマスタ設定を解除するプロジェクトを指定します。プロジェクト名は45バイト以内で指定します。

-sys サブシステム番号

サブシステム番号を0から9の範囲で指定します。

指定したサブシステム内のプロジェクトのみがマスタ設定の対象になります。このオプションを省略した場合は、サブシステム0が対象となります。

復帰値

復帰値	意味
0	正常終了
0以外	異常終了

コマンド格納場所

Windows	Systemwalker Operation Managerインストールディレクトリ¥mpwalker.jm¥bin
Solaris HP-UX AIX Linux	/opt/FJSVstem/bin

実行に必要な権限

システム管理者(Administratorsグループに属するユーザ/スーパーユーザ)の権限が必要です。

注意事項

- プロジェクト名の最後5バイトに“+”、“=”と4桁の数字がついている場合、マスタ設定はできません。
- マスタに設定したプロジェクトは、以下のどちらかの時点からマスタDB上でのスケジュールはされなくなります。
  - 設定したあと、日変わり時刻を迎えた
  - 設定したあと、プロジェクトの属するサブシステムのジョブスケジューラデーモン/サービスを再起動した
- マスタの設定を解除したプロジェクトは、以下のどちらかの時点からスケジュールされるようになります。
  - 解除したあと、日変わり時刻を迎えた
  - 解除したあと、プロジェクトの属するサブシステムのジョブスケジューラデーモン/サービスを再起動した

  なお、クライアントのマスタ情報フォルダに格納されるプロジェクトは、マスタとして設定されており、スケジュールがされなくなったプロジェクトです。

実行結果／出力形式

正常終了 : なし

異常終了 : エラー内容は標準出力に出力されます。

使用例

【UNIX版】

“営業PRJ”をマスタ設定する例

```
# stemSetPrjMaster.sh set -p 営業PRJ
```

6.5　stemCopyProject 運用変更事前設定コマンド

記述形式

```
stemCopyProject プロジェクト名 -d リリース日 [-sys サブシステム番号]
```

機能説明

マスタ情報を事前に変更しておきたい場合、マスタ情報内の事前登録したいマスタプロジェクトを同じサブシステムに複写します。本コマンドにより、同じジョブネットやジョブの定義情報を含む運用変更用プロジェクトが作成されるため、そのプロジェクトに対し、ユーザが変更を行います。リリース日に運用変更用プロジェクトの内容がマスタ情報に置き換わります。

オプション

プロジェクト名

運用変更を行うマスタプロジェクトを指定します。プロジェクト名は45バイト以内で指定します。

-d リリース日

リリース日をmmdd形式で指定します。指定可能な日付は、任意です。指定日付のスケジュール情報の分散時に運用変更されます。

-sys サブシステム番号

運用変更したいプロジェクトが存在するサブシステム番号を0から9の範囲で指定します。省略時はサブシステム0が対象となります。

復帰値

復帰値	意味
0	正常終了
0以外	異常終了

コマンド格納場所

Windows	Systemwalker Operation Managerインストールディレクトリ¥mpwalker.jm¥bin
Solaris HP-UX AIX Linux	/opt/FJSVstem/bin

実行に必要な権限

システム管理者(Administratorsグループに属するユーザ/スーパーユーザ)の権限が必要です。

注意事項

- 運用変更事前設定ができるプロジェクトは、マスタプロジェクトのみです。
- 運用変更の失敗しているプロジェクトに対して、運用変更事前設定を行うことはできません。

実行結果／出力形式

正常終了 : なし

異常終了 : エラー内容は標準出力に出力されます。

6.6 stemMoveProject 運用変更リリース日変更コマンド

記述形式

```
stemMoveProject プロジェクト名 -fd 変更前リリース日 -d 変更後リリース日 [-sys サブシステム番号]
```

機能説明

運用変更用プロジェクトのリリース日を変更します。

オプション

プロジェクト名

運用変更を行うマスタプロジェクトを指定します。

-fd 変更前リリース日

変更前のリリース日をmmdd形式で指定します。

-d 変更後リリース日

変更後のリリース日をmmdd形式で指定します。

-sys サブシステム番号

リリース日を変更したいプロジェクトが存在するサブシステム番号を0から9の範囲で指定します。省略時はサブシステム0が対象となります。

復帰値

復帰値	意味
0	正常終了
0以外	異常終了

コマンド格納場所

Windows	Systemwalker Operation Managerインストールディレクトリ¥mpwalker.jm¥bin
Solaris HP-UX	/opt/FJSVstem/bin

AIX Linux	

実行に必要な権限

システム管理者(Administratorsグループに属するユーザ/スーパーユーザ)の権限が必要です。

注意事項

運用変更の失敗している運用日に対して、運用変更用プロジェクトのリリース日変更はできません。

実行結果／出力形式

正常終了 : なし

異常終了 : エラー内容は標準出力に出力されます。

6.7 stemBackProject 運用変更取消コマンド

記述形式

```
stemBackProject プロジェクト名 -d 対象リリース日 [-sys サブシステム番号]
```

機能説明

運用変更されたマスタ情報をリリース日前の定義情報に戻します。

オプション

プロジェクト名

運用変更を取り消すマスタプロジェクトを指定します。

-d 対象リリース日

運用変更を行ったリリース日をmmdd形式で指定します。

-sys サブシステム番号

運用変更を取り消したいプロジェクトが存在するサブシステム番号を0から9の範囲で指定します。省略時はサブシステム0が対象となります。

復帰値

復帰値	意味
0	正常終了
0以外	異常終了

コマンド格納場所

Windows	Systemwalker Operation Managerインストールディレクトリ¥mpwalker.jm¥bin
Solaris HP-UX AIX Linux	/opt/FJSVstem/bin

実行に必要な権限

システム管理者(Administratorsグループに属するユーザ/スーパーユーザ)の権限が必要です。

注意事項

- サブシステム内で運用変更により異常が発生しているプロジェクトがある場合は、運用変更の取り消しを行うことはできません。

実行結果／出力形式

正常終了 : なし

異常終了 : エラー内容は標準出力に出力されます。

6.8 jobschmsgeventex メッセージ事象発生コマンド

記述形式

【Windows版】

jobschmsgeventex *メッセージ事象名*[:プロジェクト名[/ジョブネット名]] [-h ホスト名|-f ファイル名] [-p *param* [-c *suffix*]] [-sys {all | サブシステム番号}] [-d *mmdd* | -dx| -n| -s]

【UNIX版】

jobschmsgeventex *メッセージ事象名*[:プロジェクト名[/ジョブネット名]] [-h ホスト名] [-p *param* [-c *suffix*]] [-sys {all | サブシステム番号}] [-d *mmdd* | -dx| -n| -s]

機能説明

従来のメッセージ事象発生コマンド(jobschmsgevent)の仕様に、運用日を意識したメッセージ事象の発生を可能とします。

なお、発生させたメッセージ事象のカウントやクリアのされ方については、“Systemwalker Operation Manager 運用ガイド”の“メッセージ事象とジョブネットの動作”を参照してください。

オプション

メッセージ事象名

発生させるメッセージ事象名を12バイト以内で指定します。

プロジェクト名

特定のプロジェクト内のジョブネットに対してのみメッセージ事象を発生させる場合に、プロジェクト名を指定します。

-dx、-d、-nまたは-sオプションと併用した場合は、-dx、-d、-nまたは-sで取得した運用日の指定したプロジェクトに通知されます。運用日のプロジェクト名を指定すると、-dx、-d、-nまたは-sで取得した運用日は無視され、指定したプロジェクトに通知されます。

ジョブネット名

特定のジョブネットに対してのみメッセージ事象を発生させる場合に、対象となるジョブネットを指定します。

-h ホスト名

メッセージ事象を発生させるホスト名を指定します。ホスト名は64バイト以内の英数字で指定します。

-f ファイル名【Windows版】

発生させるサーバのホスト名一覧を記述したファイル名を指定します。

-p *param*

メッセージ事象により起動されるジョブネットの各ジョブに渡す可変パラメタを指定します。本オプションを指定すると、メッセージ事象を受信したジョブネットの起動時に、ジョブ定義中に記述された変数“@.VPARAM@”を、指定した可変パラメタに置き換えます。

可変パラメタは200バイト以内の文字列で指定します。半角カナおよびユーザ定義文字を含むことはできません。

可変パラメタは省略可能です。省略した場合、ジョブ定義中に記述された変数“@.VPARAM@”は

空文字列に置き換えられてジョブが起動されます。

本オプションで指定した可変パラメタを受信できるのは、単一のメッセージ事象を待ち合わせるジョブネットです。

第1階層のジョブネットに渡された可変パラメタは、階層化されたジョブネットの2階層目以降のジョブネット(子ジョブネット)に引き継がれます。

2階層目以降のジョブネット/ジョブに直接可変パラメタは渡せません。ただし、jobschchkmsgevtコマンドの使用時は、ジョブネット変数(CM_VPARAM)から受け取ることができます。ジョブネット変数については、“Systemwalker Operation Manager 運用ガイド”の“ジョブネット変数をジョブ間で利用する”を参照してください。

本オプションを複数指定することはできません。

-c *suffix*

複写起動するジョブネット名につけるサフィックスを、30バイト以内の文字列で指定します。半角英数字およびハイフン(-)が使用できます。先頭文字は半角英数字のみ使用可能です。必ず-pオプションとともに指定し、-pオプションよりも後ろに指定します。

可変パラメタを指定して起動される対象のジョブネットが、指定したサフィックスつきのジョブネット名に複写され、複写されたジョブネットが起動されます。階層化されたジョブネットの場合、2階層目以降のジョブネットも複写されます。これにより、パラメタが異なる複数のジョブネットを並行して実行したり、パラメタが異なるジョブネットの実行結果を上書きすることなく保持できるようになります。

複写されるジョブネットのジョブネット名およびジョブネット名称は、“複写元のジョブネット名（またはジョブネット名称）＋アンダーバー(_)＋*suffix*で指定した文字列”となります。

複写は、ジョブネットが起動されるタイミング（階層化されたジョブネットの場合は、親ジョブネットが起動されるタイミング）で行われます。また、複写されたジョブネットは、次回、同じ名前のジョブネットが複写されるタイミングで上書きされます。

なお、グループ内ジョブネットに対しては、本オプションは無効となります。

-sys {all | サブシステム番号}

メッセージ事象を発生させるサブシステム番号をallまたは0から9の範囲で指定します。all指定時はすべてのサブシステムに発生します。省略時は、サブシステム0に発生します。ただし、ジョブスケジューラのジョブとして起動された場合は、起動元のサブシステムと同一のサブシステム番号に発生します。

-dx

本コマンドをジョブネットのジョブに定義する場合に指定します。運用日管理プロジェクトに登録されていないジョブ、または運用日管理プロジェクトに登録されているネットワークジョブや分散実行ジョブからは、本オプションを指定して実行することはできません。

本指定をした場合、コマンドが登録されているジョブネットの運用日と同一の運用日のスケジュール情報に対し、メッセージ事象を発生させます。-sオプションとは異なり、[起動日のみ有効]が設定されている未適用ジョブネットに対してメッセージは有効となりません。

他ホストでメッセージ事象を発生させる場合は、-hオプションを併用してローカルジョブとして実行してください。

-d *mmdd*

指定された運用日のスケジュール情報に対し、メッセージ事象を発生させます。[起動日のみ有効]が設定されている未適用ジョブネットに対してメッセージは有効となりません。

-n

本コマンドをジョブネットのジョブに定義する場合に指定します。運用日管理プロジェクトに登録されていないジョブ、または運用日管理プロジェクトに登録されているネットワークジョブや分散実行ジョブからは、本オプションを指定して実行することはできません。

本指定をした場合、コマンドが登録されているジョブネットの運用日の翌運用日のスケジュール情報に対し、メッセージ事象を発生させます。[起動日のみ有効]が設定されている未適用ジョブネットに対してもメッセージは有効となります。

他ホストでメッセージ事象を発生させる場合は、-hオプションを併用してローカルジョブとして実行してください。

-s

本コマンドをジョブネットのジョブに定義する場合に指定します。運用日管理プロジェクトに登録されていないジョブ、または運用日管理プロジェクトに登録されているネットワークジョブや分散実行ジョブからは、本オプションを指定して実行することはできません。

本指定をした場合、コマンドが登録されているジョブネットの運用日と同一の運用日のスケジュール情報に対し、メッセージ事象を発生させます。-dxオプションと異なり、[起動日のみ有効]が設定されている未適用ジョブネットに対してもメッセージは有効となります。

他ホストでメッセージ事象を発生させる場合は、-hオプションを併用してローカルジョブとして実行してください。

復帰値

復帰値	意味
0	正常終了
0以外	異常終了

コマンド格納場所

Windows	Systemwalker Operation Managerインストールディレクトリ¥mpwalker.jm¥bin
Solaris HP-UX AIX Linux	/opt/FJSVstem/bin

実行に必要な権限

一般ユーザの権限で実行できます。

注意事項

- “4.20 jobschmsgeventコマンド”の注意事項を参照してください。
- Systemwalker Operation Manager V13.1.0以前のサーバへは、-nまたは-sオプションによるメッセージ事象の通知を行うことはできません。以下のエラーメッセージを出力して、メッセージ事象の発生処理が異常となります。

  MpStem: ERROR: 10376: 指定したホストではサポートされていないオプションが指定されています。(指定ホスト名)

実行結果／出力形式

正常終了 : なし

異常終了 : エラー内容は標準出力に出力されます。

6.9 stemprintnet 退避情報確認コマンド

記述形式

```
stemprintnet [-f file_name] [-title] [-sys サブシステム番号]
```

機能説明

スケジュール登録時に、プロジェクト上限値オーバーで退避したスケジュールの情報をcsv形式で出力します。

オプション

-f *file_name*

出力ファイル名をフルパスで指定します。省略時は標準出力へ出力します。

-title

タイトル行の出力を行う場合に指定します。

-sys サブシステム番号

スケジュール情報を取得するサブシステム番号を0から9の範囲で指定します。省略した場合は0となります。

復帰値

復帰値	意味
0	正常終了
0以外	異常終了

コマンド格納場所

Windows	Systemwalker Operation Managerインストールディレクトリ¥mpwalker.jm¥bin
Solaris HP-UX AIX Linux	/opt/FJSVstem/bin

実行に必要な権限

システム管理者(Administratorsグループに属するユーザ/スーパーユーザ)の権限が必要です。

実行結果／出力形式

正常終了 :

-title指定時、“ホスト名,サブシステム番号,プロジェクト名,ジョブネット名,ジョブネット名称,状態,起動予定日時,開始日時,終了日時,終了コード”の形式で出力します。

```
Host Name,Sub System,Project Name,Job Net,Job Net
Comment,Status,Scheduled Time,Start Time,End Time,Completion
Code
```

```
host1,9,over+1018,test007,test007,Pseudo,,2004/10/18
13:27:16,2004/10/18 13:27:16,2
host1,9,over+1018,test008,test008,Abend,,2004/10/18
13:26:21,2004/10/18 13:26:21,2
host1,9,action+1018,test010,test010,Disable,,,,
```

異常終了 : エラー内容は標準出力に出力されます。

出力項目の説明

出力項目について説明します。

Host Name

stemprintnetコマンドを実行したサーバのホスト名です。

Sub System

操作対象のサブシステム番号です。

Project Name

プロジェクト名です。

Job Net

ジョブネット名です。

Job Net Comment

ジョブネット名称です。

Status

ジョブネットの状態です。

以下に示した13種類の状態から選択表示されます。

Normal:

正常終了(終了コード:0)

Pseudo:

疑似正常(終了コード:1～正常終了コード上限値)

Abend:

異常終了(終了コード:正常終了コード上限値+1～255)

Forced:

強制終了(終了コード:256)

Warning:

警告

Execute:

実行中

Wait:

実行待ち

Pause:

停止中

Disable:

無効状態

Coped:

確認済

CarryOver:

持ち越し

StartDelay:

開始遅延

EndDelay:

終了遅延

[ジョブスケジューラ起動パラメタの定義]ウィンドウで疑似正常を無効にしている場合は、“疑似正常”状態は“正常終了”状態として表示されることに注意してください。

Scheduled Time

起動予定時刻です。“西暦/月/日 時:分:秒”の形式で表示します。

Start Time

前回のジョブネット開始時刻です。“西暦/月/日 時:分:秒”の形式で表示します。

End Time

前回のジョブネット終了時刻です。“西暦/月/日 時:分:秒”の形式で表示します。

Completion Code

ジョブネットの終了コードです。

6.10 stemConfig マスタスケジュール管理環境設定コマンド【UNIX版】

記述形式

```
stemConfig set {-m | -h schedule_server} {-ip ip_address}
[-sys {all| サブシステム番号}]
```

機能説明

クラスタ環境での論理IPアドレスの設定で、管理サーバおよびスケジュールサーバのIPアドレスを変更するときに使用します。コマンド実行時は、マスタスケジュール管理環境が構築されており、対象とするサブシステムは日ごとスケジュール管理になっている必要があります。

オプション

set

マスタスケジュール管理環境設定情報にIPアドレスを設定する場合に指定します。

-m

管理サーバのIPアドレスを設定するときに指定します。全スケジュールサーバ上でも設定を変更する必要があります。

-h *schedule_server*

スケジュールサーバのIPアドレスを変更するときに指定します。IPアドレスを変更するホスト名を指定します。ホスト名には、[マスタスケジュール管理環境設定]ダイアログボックス－[サブシステム]の[スケジュールサーバの追加]ダイアログボックスで設定したホスト名を指定します。

-ip *ip_address*

設定するIPアドレスを指定します。省略不可です。

-sys {all | サブシステム番号}

IPアドレスを設定するサブシステム番号をallまたは0から9の範囲で指定します。all指定時は、日ごとスケジュール管理を行っているすべてのサブシステムが対象となります。省略時は、サブシステム0が対象となります。

復帰値

復帰値	意味
0	正常終了
0以外	異常終了

コマンド格納場所

Solaris HP-UX AIX Linux	/opt/FJSVstem/bin

実行に必要な権限

システム管理者(スーパーユーザ)の権限が必要です。

実行結果／出力形式

正常終了 : なし

異常終了 : エラー内容は標準出力に出力されます。

6.11 stemprint -f(-F)/-g(-G)コマンド

記述形式

```
stemprint {{-f|-F} P | {-g|-G} YYYY MM DD } {[-long]| [ -csv [ -title ] ]}
        [-vparam] [ -sys サブシステム番号 ]
```

機能説明

ジョブネットの履歴情報を出力します。

オプション

{-f|-F} *P*

コマンドの実行ユーザが参照できるジョブネットの履歴情報を日単位で出力します。

運用日管理プロジェクトにアクセス権限がない場合や、運用日管理プロジェクトが削除済みの場合、管理サーバではマスタプロジェクトのアクセス権限に従って履歴情報を出力します。スケジュールサーバでは、マスタスケジュール管理環境設定の[スケジュール情報を保管する世代数]で設定した世代数分のみの履歴情報を出力します。

何日前の履歴情報を出力するかを0から5までの数字で指定します。

{-g|-G} *YYYY MM DD*

コマンドの実行ユーザが参照できるジョブネットについて履歴情報を出力します。

運用日管理プロジェクトにアクセス権限がない場合や、運用日管理プロジェクトが削除済みの場合、管理サーバではマスタプロジェクトのアクセス権限に従って履歴情報を出力します。スケジュールサーバでは、マスタスケジュール管理環境設定の[スケジュール情報を保管する世代数]で設定した世代数分のみの履歴情報を出力します。

履歴情報の出力対象日を空白で区切った年月日の形式で指定します。

-long

出力項目“Job Net”および“Job Net Comment”を最大値まで出力します。

-csv

ジョブネットの履歴情報を、CSV形式で出力します。

出力結果をファイルに保存し、ジョブネットの履歴情報を表計算ソフトで管理することができます。(注)

注)

表計算ソフトに取り込む前に、出力結果を表計算ソフトがサポートする文字コードに変換しておく必要があります。

-title

1行目にタイトル行を出力します。-csvオプションが指定されている場合のみ指定可能です。

-vparam

可変パラメタを指定して起動されたジョブネットの場合、可変パラメタを表示します。

-sys サブシステム番号

複数サブシステム運用をしているシステムで、操作対象のサブシステムを指定するためのオプションです。操作対象とするサブシステム番号を、0から9の範囲で指定します。このオプションを省略した場合、サブシステム番号0が操作対象となります。

復帰値

復帰値	意味
0	正常終了
0以外	異常終了

コマンド格納場所

Windows	Systemwalker Operation Managerインストールディレクトリ¥mpwalker.jm¥bin
Solaris HP-UX AIX Linux	/opt/FJSVstem/bin

実行に必要な権限

一般ユーザの権限で実行できます。

注意事項

- サイクリックに管理されているジョブスケジューラのログファイル(jobdb1.log/jobdb2.log/jobdb3.log)のうち、現在使用中のログファイルと1世代前のログファイルの履歴情報を出力します。
- 日単位に出力される情報は、対象となる日付に開始情報が存在するジョブネットの履歴情報です。
- UTF-8コードの環境では、ジョブネット名およびジョブネット名称に日本語を使用している場合、出力項目の位置がずれて出力されることがあります。【UNIX版】

使用例

```
stemprint -f 0 -vparam -sys 2
```

実行結果／出力形式

正常終了 : -csvオプションの指定がない場合

```
  History Information of Job Net  date: 2007/08/13  time: 16:45

User Name: suzuki

Day Change Time: 00:00

Project Name: 総務部門

Date Job Net  Job Net Comment      Status   Start Time          End Time            End Code  Variable Parameter
0813 kyuuyo   給与・賞与計算         Normal   2007/08/13 10:00:01 2007/08/13 11:33:32 0         ”200707”
0813 backup   バックアップ業務       Normal   2007/08/13 10:30:01 2007/08/13 11:49:26 0
0813 uriage   売上データ管理         Normal   2007/08/13 12:48:42 2007/08/13 12:50:56 0         ”sales01.dat”
0813 kanri    顧客管理               Abend    2007/08/13 13:32:45 2007/08/13 13:42:37 2
0813 data     出納データ管理         Forced   2007/08/13 16:27:21 2007/08/13 16:31:21 256
0813 data     出納データ管理         Execute  2007/08/13 16:44:30
```

正常終了 : -csvオプションの指定がある場合

```
Host Name,Sub System,Project Name,Date,Job Net,Job Net Comment,Status,Start Time,End Time,End Code,Variable
Parameter
host1,3,総務部門,0813,kyuuyo,給与・賞与計算,Normal,2007/08/13 10:00:01,2007/08/13 11:33:32,0,”200707”
host1,3,総務部門,0813,backup,バックアップ業務,Normal,2007/08/13 10:30:01,2007/08/13 11:49:26,0,
host1,3,総務部門,0813,uriage,売上データ管理,Normal,2007/08/13 12:48:42,2007/08/13 12:50:56,0,”sales01.dat”
host1,3,総務部門,0813,kanri,顧客管理,Abend,2007/08/13 13:32:45,2007/08/13 13:42:37,2,
host1,3,総務部門,0813,data,出納データ管理,Forced,2007/08/13 16:27:21,2007/08/13 16:31:21,256,
host1,3,総務部門,0813,data,出納データ管理,Execute,2007/08/13 16:44:30,,,
```

異常終了 : エラー内容は標準出力に出力されます。

出力項目の説明

出力項目について説明します。

User Name

stemprintコマンドの実行ユーザの名前です。-csvオプション指定時は表示されません。

Day Change Time

日変わり時刻です。-csvオプション指定時は表示されません。

Host Name

stemprintコマンドを実行したサーバのホスト名です。-csvオプション指定時のみ表示されます。

Sub System

操作対象のサブシステム番号です。-csvオプション指定時のみ表示されます。

Project Name

マスタプロジェクト名です。

Date

マスタスケジュール管理機能の運用日です。運用日管理プロジェクト以外のプロジェクトの場合は、空欄になります。

Job Net

ジョブネット名です。

先頭から8バイト分まで表示します。-longオプションを指定した場合は、50バイト分すべて表示します。

Job Net Comment

ジョブネット名称です。

先頭から24バイト分まで表示します。-longオプションを指定した場合は、100バイト分すべて表示します。

Status

ジョブネットの状態です。

以下に示した8種類の状態から選択表示されます。

Normal:

正常終了(終了コード:0)

Pseudo:

疑似正常(終了コード:1～正常終了コード上限値)

Abend:

異常終了(終了コード:正常終了コード上限値+1～255)

Forced:

強制終了(終了コード:256)

Execute:

実行中

E_Cancel:

実行拒否

Closed:

実行中断(終了コード:239)

Skipped:

スキップ

Start Time

ジョブネットが起動した時刻です。“西暦/月/日 時:分:秒”の形式で表示します。

End Time

ジョブネットが終了した時刻です。“西暦/月/日 時:分:秒”の形式で表示します。

End Code

ジョブネットの終了コードです。

Variable Parameter

可変パラメタを指定して起動されたジョブネットの可変パラメタです。-vparamオプションを指定した場合のみ指定されます。

6.12 stemprint -n(-N)コマンド

記述形式

```
stemprint {-n|-N} [-d MMDD |-p プロジェクト名] [-toplevel]
        {[-long]|[ -csv [-title]]} [ -sys サブシステム番号 ]
```

機能説明

ジョブネットの現在の状態を出力します。

オプション

-n|-N

コマンドの実行ユーザが参照できるジョブネットについて現在の状態を出力します。

-dオプションまたは-pオプションの指定がない場合は、当日の運用日管理プロジェクトに登録されているジョブネットの現在の状態を出力します。

-d *MMDD*

指定した運用日の、運用日管理プロジェクトすべてに登録されているジョブネットの状態を出力します。

運用日を月日の形式で指定します。

-p プロジェクト名

指定したプロジェクトに登録されているジョブネットの現在の状態を出力します。

-toplevel

子ジョブネット、およびリンクジョブネットを出力しません。階層化されていないジョブネットおよび階層化されているジョブネットのうち第1階層のものを出力します。

-long

出力項目“Job Net”および“Job Net Comment”を最大値まで出力します。

-csv

ジョブネットの現在の状態を、CSV形式で出力します。

出力結果をファイルに保存し、ジョブネットの現在の状態を表計算ソフトで管理することができます。(注)

注)

表計算ソフトに取り込む前に、出力結果を表計算ソフトがサポートする文字コードに変換しておく必要があります。

-title

1行目にタイトル行を出力します。-csvオプションが指定されている場合のみ指定可能です。

-sys サブシステム番号

複数サブシステム運用をしているシステムで、操作対象のサブシステムを指定するためのオプションです。操作対象とするサブシステム番号を、0から9の範囲で指定します。このオプションを省略した場合、サブシステム番号0が操作対象となります。

復帰値

復帰値	意味
0	正常終了
0以外	異常終了

コマンド格納場所

Windows	Systemwalker Operation Managerインストールディレクトリ¥mpwalker.jm¥bin
Solaris HP-UX AIX Linux	/opt/FJSVstem/bin

実行に必要な権限

一般ユーザの権限で実行できます。

注意事項

UTF-8コードの環境では、ジョブネット名およびジョブネット名称に日本語を使用している場合、出力項目の位置がずれて出力されることがあります。【UNIX版】

実行結果／出力形式

正常終了 : -csvオプションの指定がない場合

```
  Job Net Status List            date: 2004/08/13    time: 16:29

User Name: suzuki

Project Name: 営業+0813

Job Net Job Net Comment Status  Scheduled Time   Start Time       End Time         Completion Code
hanbai  販売管理        Abend                   2004/08/13 10:22 2008/08/13 10:38 10
kokyaku 顧客売上管理    Wait    2004/08/13 18:00
eigyou  営業週報作成    Disable                 2004/08/13 13:00 2004/08/13 13:12


Project Name: 総務+0813

Job Net Job Net Comment Status  Scheduled Time   Start Time        End Time        Completion Code
suitou  出納データ管理  Execute                 2004/08/13 16:00
kyuuyo  給与・賞与計算  Normal                  2004/08/13 12:00  2004/08/13 12:50 0
```

正常終了 : -csvオプションの指定がある場合

```
Host Name,Sub System,Project Name,Job Net,Job Net Comment,Status,Scheduled Time,Start Time,End
Time,Completion Code
host1,3,営業+0813,hanbai,販売管理,Abend,,2004/08/13 10:22,2004/08/13 10:38,10
host1,3,営業+0813,kokyaku,顧客売上管理,Wait,2004/08/13 18:00,,,
host1,3,営業+0813,eigyou,営業週報作成,Disable,,2004/08/13 13:00,2004/08/13 13:12,
```

```
host1,3,総務+0813,suitou,出納データ管理,Execute,,2004/08/13 16:00,,
host1,3,総務+0813,kyuuyo,給与・賞与計算,Normal,,2004/08/13 12:00,2004/08/13 12:50,0
```

異常終了 : エラー内容は標準出力に出力されます。

出力項目の説明

出力項目について説明します。

User Name

stemprintコマンドの実行ユーザの名前です。-csvオプション指定時は表示されません。

Host Name

stemprintコマンドを実行したサーバのホスト名です。-csvオプション指定時のみ表示されます。

Sub System

操作対象のサブシステム番号です。-csvオプション指定時のみ表示されます。

Project Name

プロジェクト名です。

Job Net

ジョブネット名です。

先頭から8バイト分まで表示します。-longオプションを指定した場合は、50バイト分すべて表示します。

Job Net Comment

ジョブネット名称です。

先頭から24バイト分まで表示します。-longオプションを指定した場合は、100バイト分すべて表示します。

Status

ジョブネットの状態です。

以下に示した13種類の状態から選択表示されます。

Normal:

正常終了(終了コード:0)

Pseudo:

疑似正常(終了コード:1～正常終了コード上限値)

Abend:

異常終了(終了コード:正常終了コード上限値+1～255)

Forced:

強制終了(終了コード:256)

Warning:

警告

Execute:

実行中

Wait:

実行待ち

Pause:

停止中

Disable:

無効状態

Coped:

確認済

CarryOver:

持ち越し

StartDelay:

開始遅延

EndDelay:

終了遅延

[ジョブスケジューラ起動パラメタの定義]ウィンドウで疑似正常を無効にしている場合は、“疑似正常”状態は“正常終了”状態として表示されることに注意してください。

Scheduled Time

起動予定時刻です。“西暦/月/日 時:分”の形式で表示します。

Start Time

前回のジョブネット開始時刻です。“西暦/月/日 時:分”の形式で表示します。

End Time

前回のジョブネット終了時刻です。“西暦/月/日 時:分”の形式で表示します。

Completion Code

ジョブネットの終了コードです。

6.13 stemschnet ジョブネット投入コマンド

記述形式

```
stemschnet -ent プロジェクト名/ジョブネット名 -d yyyymmdd
        [-correct] [ -sys サブシステム番号 ]
```

機能説明

すでに作成済みの運用日管理プロジェクトへマスタ情報からジョブネットを個別に投入(抽出・登録)します。スケジュールサーバが登録済みでデフォルトホスト名に指定がある場合は、指定スケジュールサーバにジョブネットを投入します。

オプション

-ent

指定されたジョブネットを運用日管理プロジェクトに投入します。

プロジェクト名

マスタ情報のプロジェクト名を指定します。

指定可能なプロジェクトは、マスタプロジェクトまたは運用変更用プロジェクトです。プロジェクト名は50バイト以内で指定します。

以下の場合は登録できません。

- 投入する運用日の運用日管理プロジェクトが存在しない、かつ登録されている運用日管理プロジェクト数が256の場合

ジョブネット名

マスタ情報のジョブネット名を指定します。ジョブネット名は50バイト以内で指定します。

以下のジョブネットは指定できません。

- 階層(子)ジョブネット
- リンクジョブネット
- マスタリンクジョブネット
- グループ内ジョブネット

投入する運用日の起動日が設定されているジョブネットを指定します。

なお、投入する運用日の運用日管理プロジェクトに、すでに同名のジョブネットが存在する場合は投入できません。

-d *yyyymmdd*

ジョブネットの投入先運用日をyyyymmdd形式で指定します。

指定可能な運用日は当日または未来です。未来の場合は、30日先までの運用日を指定できます。

スケジュール情報の分散が実施済みの運用日のみ指定できます。

-correct

ジョブネット投入時にマスタ情報に設定されている起動時刻を経過している場合、起動時刻を現在時刻の5分後に時刻補正しジョブネットを投入します。起動時刻が、現在時刻から5分以内の場合も同様に時刻補正を行います。起動条件が「時刻起動(単一起動)」以外のジョブネットは、本オプションは無視されます。

-sys サブシステム番号

複数サブシステム運用をしているシステムで、操作対象のサブシステムを指定するためのオペランドです。操作対象とするサブシステム番号を、0から9の範囲で指定します。このオペランドを省略した場合、サブシステム番号0が操作対象となります。

復帰値

復帰値	意味
0	正常終了
0以外	異常終了

コマンド格納場所

Windows	Systemwalker Operation Managerインストールディレクトリ¥mpwalker.jm¥bin
Solaris HP-UX AIX Linux	/opt/FJSVstem/bin

実行に必要な権限

システム管理者(Administratorsグループに属するユーザ/スーパーユーザ)の権限が必要です。

注意事項

- マスタ情報のジョブネットおよびジョブが無効/停止操作されていても、ジョブネットおよびジョブの状態は引き継がれません。ジョブネット投入後のジョブネットおよびジョブの状態は“実行待ち”になります。

  ジョブネット投入時にマスタ情報の状態を保持させたい場合は、以下の手順を実施してください。

  ジョブネットの場合:

  1. jobschprint -nコマンドで登録されているジョブネットについて、現在の状態を出力します。
  2. 1.で無効/停止操作されていた場合、jobschcontrolコマンドでジョブネットを無効/停止状態にします。

  ジョブの場合:

  1. jobschprint -pコマンドで登録されているジョブについて、現在の状態を出力します。
  2. 1.で無効/停止操作されていた場合、jobschctljobコマンドでジョブを無効/停止状態にします。(注)

     注)jobschctljobコマンドで制御を行うジョブには、“ジョブ名”が必ず指定されている必要があります。

- Systemwalker Operation Manager V13.1.0以前のスケジュールサーバへ“-correct”オプションを指定してジョブネットを投入することはできません。ジョブネット投入コマンドが異常終了します。

使用例

マスタプロジェクト”PRJ”に登録されているジョブネット“NET001”を運用日管理プロジェクト”PRJ+0523”に投入する例

```
# stemschnet -ent PRJ/NET001 -d 20050523
```

実行結果/出力形式

正常終了：なし

異常終了：エラー内容は標準出力に出力されます。

【UNIX版】

スケジュールサーバにジョブネットを投入する場合は、正常終了、異常終了にかかわらず、以下の形式でスケジュールサーバ名が標準出力に出力されます。

異常終了の場合は、スケジュールサーバ名の出力の後に、エラー内容が出力されます。

```
Schedule Server : XXXX
```

XXXX ： スケジュールサーバ名

6.14 stemsetdbpath クラスタ用DBディレクトリ変更コマンド【Windows版】

記述形式

```
stemsetdbpath {-p | dbpath [-n] [-r] [-sys サブシステム番号]}
```

機能説明

マスタスケジュール管理機能のデータベースディレクトリを変更します。クラスタ環境において、マスタスケジュール管理機能のデータベースディレクトリを共有ディスク上に再配置する場合に使用します。

運用系で実行する場合は、“-n”オプションを指定せず、マスタスケジュール管理機能のデータベースディレクトリの登録情報の変更と複写をします。待機系で実行する場合は、“-n”オプションを指定し、登録情報のみを変更します。

オプション

-p

現在設定されているデータベースディレクトリを表示します。

dbpath

移動先データベースディレクトリをフルパスで指定します。指定されたディレクトリにstemDB*n*(*n*:サブシステム番号)を付加し、マスタスケジュール管理機能のデータベースディレクトリを移動します。

-n

データベースディレクトリの登録情報のみを変更する場合に指定します。省略した場合、移動先ディレクトリにデータベースを移動します。

-r

データベースディレクトリの変更前に、データベースを退避します。変更前のディレクトリと同一ディレクトリにstemDB*n*.local(*n*：サブシステム番号)を作成し、データベースを退避します。省略した場合、データベースは退避されません。

-sys サブシステム番号

データベースディレクトリを変更するサブシステム番号を0から9の範囲で指定します。省略時は、サブシステム0が対象となります。

復帰値

復帰値	意味
0	正常終了
0以外	異常終了

コマンド格納場所

Windows	Systemwalker Operation Managerインストールディレクトリ¥mpwalker.jm¥bin

実行に必要な権限/実行環境

システム管理者(Administratorsグループに属するユーザ)の権限が必要です。

注意事項

ジョブスケジューラサービスを停止した環境で実行してください。

使用例

- 1:1運用待機のクラスタ環境構築時、運用系でデータベースディレクトリを共有ディスクに配置する場合

```
stemsetdbpath.exe y:¥disk -sys 0
```

サブシステム0のデータベースディレクトリ情報を、「y:¥disk¥stemDB0」に変更し、データベースを移動します。

・ 1:1運用待機のクラスタ環境構築時、待機系でデータベースディレクトリを共有ディスクに配置する場合

```
stemsetdbpath.exe y:¥disk -n -sys 0
```

サブシステム0のデータベースディレクトリ情報を、「y:¥disk¥stemDB0」に変更します。データベースは移動しません。

・ 2ノード相互待機のクラスタ環境構築時、運用系でサブシステム1のデータベースディレクトリを共有ディスクに配置する場合

```
stemsetdbpath.exe y:¥disk -sys 1
```

サブシステム1のデータベースディレクトリ情報を「y:¥disk¥stemDB1」に変更し、データベースを移動します。

・ 現在のデータベースディレクトリを表示する場合

```
stemsetdbpath.exe -p
```

現在設定されているデータベースディレクトリを表示します。

実行結果/出力形式

正常終了 :

・ データベースディレクトリ変更時

MpStem: INFO: stemsetdbpath.exe exited normally.

・ データベースディレクトリ表示時

```
No. path
-----------------------------------------------------
 1 z:¥omgr1¥mpstemsv¥stemDB1
 2 y:¥omgr2¥mpstemsv¥stemDB2
```

異常終了 : エラー内容が標準出力に表示されます。

6.15 stemregservice クラスタ用サービス登録コマンド【Windows版】

記述形式

```
stemregservice {-reg | -del} [-sys サブシステム番号]
```

機能説明

マスタスケジュール管理機能のクラスタ用サービス(“Systemwalker MpStem*n*”サービス)の登録/削除を行います。クラスタ環境構築において、マスタスケジュール管理機能を使用する場合に使用します。

オプション

-reg

マスタスケジュール管理機能のクラスタ用サービスを登録します。

-del

マスタスケジュール管理機能のクラスタ用サービスを削除します。

-sys サブシステム番号

サービスの登録/削除を行うサブシステム番号を0から9の範囲で指定します。省略時は、サブシステム0が対象となります。

復帰値

復帰値	意味
0	正常終了
0以外	異常終了

コマンド格納場所

Windows	Systemwalker Operation Managerインストールディレクトリ¥mpwalker.jm¥bin

実行に必要な権限/実行環境

Administratorsグループに所属したユーザで、“オペレーティングシステムの一部として機能”の権利を有するユーザが必要です。

使用例

- サブシステム0のマスタスケジュール管理機能のクラスタ用サービスを作成する場合

```
stemregservice.exe -reg -sys 0
```

- サブシステム0のマスタスケジュール管理機能のクラスタ用サービスを削除する場合

```
stemregservice.exe -del -sys 0
```

実行結果/出力形式

正常終了 : MpStem: INFO: stemregservice.exe exited normally.

異常終了 : エラー内容が標準出力に表示されます。

第7章 ジョブ実行制御コマンド

本章では、Systemwalker Operation Managerが提供するジョブ実行制御コマンドについて説明します。なお、コマンドは、Systemwalker Operation Managerのサーバ機能がインストールされているサーバで起動してください。

注意

WindowsのUAC機能を有効にしている環境でコマンドを実行する場合、OSにログインしたユーザの権限によって権限の昇格を求めるダイアログが表示され、対処が必要となります。

詳細は、“Systemwalker Operation Manager 運用ガイド”の、“Windowsでコマンドを実行する場合の注意事項【Windows版】”を参照してください。

7.1 コマンド一覧

ジョブ実行制御で使用するコマンドには以下のものがあります。

システム管理者権限で使用するコマンド

ジョブ実行制御が提供するシステム管理者権限で使用するコマンドについて説明します。システム管理者権限で使用するコマンドを以下に示します。

mjstart 運用開始コマンド

ジョブ実行制御の運用を開始します。

mjstop 運用停止コマンド

ジョブ実行制御の運用を終了します。

mjsview ログファイル出力コマンド

ジョブ実行の履歴情報とジョブ実行制御サービスの稼働情報を表示します。

qalter キューの属性変更コマンド

キューの属性を変更します。

qjalter ジョブの属性変更コマンド

ジョブの属性を変更します。

qstart キューの起動コマンド

停止しているキューを起動します。

qstop キューの停止コマンド

動作中のキューを停止します。

mjdjadd 分散実行の構成ホスト追加コマンド

運用中に分散実行先ホストを追加します。

mjdjdel 分散実行の構成ホスト削除コマンド

運用中に分散実行先ホストを削除します。

mjdjalt 分散実行の設定変更コマンド

運用中に分散実行先のジョブの多重度を変更します。

mjsmdset 起動モード指定コマンド

ジョブ実行制御の起動モードを指定します。

一般ユーザの権限で使用するコマンド

ジョブ実行制御が提供する一般ユーザの権限で使用するコマンドについて説明します。一般ユーザの権限で使用するコマンドを以下に示します。

qdel ジョブの削除コマンド

ジョブを削除します。

qhold ジョブの保留(ホールド)コマンド

ジョブを保留します。

qjstat ジョブ情報の表示コマンド

ジョブの情報(ジョブ名/ジョブ番号/キュー名/ジョブの状態など)を表示します。

qmove ジョブのキュー間移動コマンド

ジョブを他のキューへ移動します。

qrls ジョブの保留解除(リリース)コマンド

ジョブの保留解除を行います。

qstat キューの状態表示コマンド

キューの状態を表示します。

qsub ジョブの投入コマンド

ジョブを投入します。

mjdjstat 分散実行の状態表示コマンド

分散実行の状態を表示します。

mjrmtjob リモートマシン上のコマンド/シェルスクリプト実行コマンド

リモートマシン上に配置されたコマンドまたはスクリプトを実行します。

mjkeygen キーペア作成コマンド【Windows版】

mjrmtjobコマンドで使用するキーペア(公開鍵/秘密鍵)を作成/設定・削除します。

注意事項

他サーバに実行依頼したネットワークジョブの操作は、実行サーバ上ではなく、ジョブの投入元サーバで行ってください。実行サーバ側で操作した場合の動作は保証されません。

7.2 mjstart 運用開始コマンド

記述形式

```
mjstart [-sys サブシステム番号]  (注)
```

注)

“[-sys サブシステム番号]”はSystemwalker Operation Manager EEのオプションです。

機能説明

ジョブ実行制御は以下の2段階の処理により機能が開始します。

a. ジョブ実行制御サービスの起動

b. ジョブ実行制御の運用の開始

Systemwalker Operation Managerサーバの起動とともに、a.が自動起動され、さらにb.が自動起動されます。

mjstartコマンドは、b.のジョブ実行制御の運用を開始するコマンドです。

オプション

EE GEE **-sys サブシステム番号**

複数サブシステム運用をしているシステムで、操作対象のサブシステムを指定するためのオプションです。操作対象とするサブシステム番号を、0から9の範囲で指定します。このオプションを省略した場合、サブシステム番号0が操作対象となります。

復帰値

復帰値	意 味
0	コマンド処理が正常に終了しました。
0以外	コマンド処理が正常に終了しませんでした。エラーメッセージが出力されます。

コマンド格納場所

Windows	Systemwalker Operation Managerインストールディレクトリ¥MpWalker.JM¥bin
Solaris HP-UX AIX Linux	/usr/lib/mjes

実行に必要な権限

システム管理者(Administratorsグループに属するユーザ/スーパーユーザ)権限が必要です。

7.3 mjstop 運用停止コマンド

記述形式

```
mjstop [-sys サブシステム番号] [-c | -w] [-i]
                    (注)
```

注)

“[-sys サブシステム番号]”はSystemwalker Operation Manager EEのオプションです。

機能説明

ジョブ実行制御は以下の2段階の処理により機能が終了します。

a. ジョブ実行制御の運用の停止

b. ジョブ実行制御サービス/デーモンの停止

Systemwalker Operation Managerサーバの停止とともに、a.が自動終了し、さらにb.が自動終了します。

mjstopコマンドは、a.のジョブ実行制御の運用を停止するコマンドです。

また、次回のジョブ実行制御の運用を開始するとき、またはサービス/デーモンを起動するときの起動モードの宣言を行います。

オプションを省略した場合は以下になります。

- 起動モードは変更されません。【Windows版】
- -wオプションが指定されたものとみなします。【UNIX版】

起動モードについては、“Systemwalker Operation Manager 運用ガイド”を参照してください。

オプション

EE GEE

-sys サブシステム番号

複数サブシステム運用をしているシステムで、操作対象のサブシステムを指定するためのオプションです。操作対象とするサブシステム番号を、0から9の範囲で指定します。このオプションを省略した場合、サブシステム番号0が操作対象となります。

-c

次回の起動モードを初期化モードとします。

-w

次回の起動モードを継続モードとします。

-i

本コマンドを実行後、サービス/デーモンの停止時に、ワークユニットを停止せず、サービスの停止のみ行います。

復帰値

復帰値	意 味
0	コマンド処理が正常に終了しました。
0以外	コマンド処理が正常に終了しませんでした。エラーメッセージが出力されます。

コマンド格納場所

Windows	Systemwalker Operation Managerインストールディレクトリ¥MpWalker.JM¥bin
Solaris HP-UX AIX Linux	/usr/lib/mjes

実行に必要な権限

システム管理者(Administratorsグループに属するユーザ/スーパーユーザ)権限が必要です。

注意事項

- -iオプションを指定してワークユニットを停止しなかった場合は、再起動時に以下の点に注意してください。

  ー ワークユニットが停止していないため、Systemwalker Operation Managerからのワークユニットの再起動に失敗します。ワークユニットを事前に停止しておいてください。

  なお、クラスタ運用時の場合も、フェールオーバ後、同様にワークユニットの再起動に失敗します。この場合も事前にワークユニットを停止しておいてください。

- ジョブスケジューラデーモンが起動中の状態で本コマンドを実行しないでください。ジョブスケジューラデーモンが起動中の状態で本コマンドを実行した場合ジョブネットが実行中のままとなる現象が発生する可能性があります。本コマンドを実行する場合はジョブスケジューラデーモンを停止してから実行してください。【UNIX版】

7.4 mjsview ログファイル出力コマンド

記述形式

```
mjsview [-sys サブシステム番号] [-c | -csv] [ログファイル名]
                    (注)
```

注)

“ [-sys サブシステム番号]”はSystemwalker Operation Manager EEのオプションです。

機能説明

ジョブ実行制御が出力するログファイルは、バイナリ属性を含んでおり、直接参照することができません。よって、このコマンドを使ってログファイルの内容を編集し、標準出力に形式化して出力します。

ログファイルを出力するための設定方法は、“Systemwalker Operation Manager 導入ガイド”を参照してください。

なお、[運用情報の定義]ウィンドウの“ジョブ実行履歴情報を保存する”(logパラメタ)を指定していない場合、ログファイルは以下に出力されます。

【Windows版】

```
ジョブ実行制御スプールパス¥hist¥日付.log
```

日付：yyyymmdd形式

【UNIX版】

```
/var/spool/mjes/hist/日付.log
```

日付：yyyymmdd形式

オプション

EE GEE

-sys サブシステム番号

複数サブシステム運用をしているシステムで、操作対象のサブシステムを指定するためのオプションです。操作対象とするサブシステム番号を、0から9の範囲で指定します。このオプションを省略した場合、サブシステム番号0が操作対象となります。

-c

“-c”を指定すると、コマンド発行時点までのログファイルの内容を形式化して出力した後、ログファイルを継続して監視し、ログを動的に形式化して出力し続けます。ログファイルが存在しない場合は、コマンドは終了します。

“-c”を省略すると、コマンド発行時点までのログファイルの内容を形式化して出力した後、コマンドは終了します。

-csv

“-csv”を指定すると、コマンド発行時点までのログファイルの内容をcsv形式で形式化して出力した後、コマンドは終了します。

ログファイル名

ログファイル名を指定します。

ログファイル名を省略すると、ジョブ実行履歴情報を最後に出力したログファイルが処理の対象となります。特定の日のログファイルの内容などを処理したい場合は、ログファイル名をフルパスで指定してください。

復帰値

復帰値	意 味
0	コマンド処理が正常に終了しました。

復帰値	意 味
0以外	コマンド処理が正常に終了しませんでした。エラーメッセージが出力されます。

コマンド格納場所

Windows	Systemwalker Operation Managerインストールディレクトリ¥MpWalker.JM¥bin
Solaris HP-UX AIX Linux	/usr/lib/mjes

実行に必要な権限

システム管理者(Administratorsグループに属するユーザ/スーパーユーザ)権限が必要です。

注意事項

- ジョブをキューイング中にSystemwalker Operation Managerを再起動したり、サーバがダウンした場合には、ログファイルに情報が格納されない場合があります。
- Systemwalker Operation Managerを再起動した場合、リカバリ対象のジョブについては、再度ジョブを開始するため、同一のログが出力される場合があります。
- 以下の条件をすべて満たしている場合、ジョブのpurgeログが一度にログファイルへ出力されます。
  - ネットワークジョブが実行される実行先サーバである。
  - 実行先サーバにおいて、ジョブが終了した時点で投入元サーバとの回線接続に失敗した(投入元サーバのシステムダウンや回線が切断された時)。
  - 回線接続に失敗した後、接続に成功した(投入元サーバが再起動された時や回線接続が復旧した時)。

参考

ログファイルの切り替え契機について

ログファイルは、日付が変わったときに自動的に切り替わるものではありません。

前日からジョブ実行制御のサービス/デーモンが起動しており、日付が変わってからジョブが一度も実行されていない場合は、本日分のログファイルが出力されていない状態であるため、ログファイルは切り替わりません。日付が変わったあとにジョブが実行されたタイミングで、ログファイルが切り替わります。

表示例

mjsview ログファイル出力コマンドの表示例を以下に示します。

```
2015/05/27 08:38:52 MJS901I MpMjes service start.
2015/05/27 15:10:22 MJS007I job jobschsleep(13.server1) accept
2015/05/27 15:10:22 MJS001I job jobschsleep(13.server1) start at queue1
2015/05/27 15:10:22 MJS003I job jobschsleep(13.server1) step mjsrjob start on server2:1
2015/05/27 15:10:26 MJS004I job jobschsleep(13.server1) step mjsrjob stop RC(0000)
2015/05/27 15:10:26 MJS002I job jobschsleep(13.server1) stop RC(0000)
2015/05/27 15:10:26 MJS006I job jobschsleep(13.server1) purge
2015/05/27 15:10:32 MJS007I job jobschsleep(14.server1) accept
2015/05/27 15:10:33 MJS001I job jobschsleep(14.server1) start at queue1
2015/05/27 15:10:41 MJS005I job jobschsleep(14.server1) cancel
```

```
2015/05/27 15:10:42 MJS002I job jobschsleep(14.server1) stop EC(40000222)
2015/05/27 15:10:42 MJS006I job jobschsleep(14.server1) purge
```

備考．ジョブ実行制御サービスの起動
正常終了ジョブ(ジョブ番号=13)
強制終了ジョブ(ジョブ番号=14)

EC：ECの文字列で表示される内部コードは16進数表示となります。
RC：RCの文字列で表示される内部コードは10進数表示となります。

“-csv”オプション指定時の表示例を以下に示します。

```
2015/05/27,08:38:52,MJS901I,MpMjes service start.
2015/05/27,15:10:22,MJS007I,job jobschsleep(13.server1) accept
2015/05/27,15:10:22,MJS001I,job jobschsleep(13.server1) start at queue1
2015/05/27,15:10:22,MJS003I,job jobschsleep(13.server1) step mjsrjob start on
server2:1
2015/05/27,15:10:26,MJS004I,job jobschsleep(13.server1) step mjsrjob stop,RC(0000)
2015/05/27,15:10:26,MJS002I,job jobschsleep(13.server1) stop,RC(0000)
2015/05/27,15:10:26,MJS006I,job jobschsleep(13.server1) purge
2015/05/27,15:10:32,MJS007I,job jobschsleep(14.server1) accept
2015/05/27,15:10:33,MJS001I,job jobschsleep(14.server1) start at queue1
2015/05/27,15:10:41,MJS005I,job jobschsleep(14.server1) cancel
2015/05/27,15:10:42,MJS002I,job jobschsleep(14.server1) stop,EC(40000222)
2015/05/27,15:10:42,MJS006I,job jobschsleep(14.server1) purge
```

備考．ジョブ実行制御サービスの起動
正常終了ジョブ(ジョブ番号=13)
強制終了ジョブ(ジョブ番号=14)

EC：ECの文字列で表示される内部コードは16進数表示となります。
RC：RCの文字列で表示される内部コードは10進数表示となります。

実行結果/出力形式

以下に出力形式を示します。

```
1) yyyy/mm/dd hh:mm:ss MJS001I job jobname(jobno.hostname) start at queuename
2) yyyy/mm/dd hh:mm:ss MJS002I job jobname(jobno.hostname) stop RC(rc)|EC(ec)
3) yyyy/mm/dd hh:mm:ss MJS003I job jobname(jobno.hostname) step stepname start
   yyyy/mm/dd hh:mm:ss MJS003I job jobname(jobno.hostname) step stepname start
on host-name:n
4) yyyy/mm/dd hh:mm:ss MJS004I job jobname(jobno.hostname) step stepname stop
RC(rc)|EC(ec)
5) yyyy/mm/dd hh:mm:ss MJS005I job jobname(jobno.hostname) cancel
6) yyyy/mm/dd hh:mm:ss MJS006I job jobname(jobno.hostname) purge
7) yyyy/mm/dd hh:mm:ss MJS007I job jobname(jobno.hostname) accept
   yyyy/mm/dd hh:mm:ss MJS007I job jobname(jobno.hostname) accept from host-
name:n
8) yyyy/mm/dd hh:mm:ss MJS901I MpMjes service start.
9) yyyy/mm/dd hh:mm:ss MJS902I MpMjes service stop.
```

以下に各ログの意味を説明します。
1）ジョブが開始したことを示します。
2）ジョブが終了したことを示します。
3）ジョブステップが開始したことを示します。（注1）
4）ジョブステップが終了したことを示します。
5）強制終了要求を受け付けたことを示します。
6）ジョブをスプールから削除したことを示します。
7）ジョブを受け付けたことを示します。（注2）

8）ジョブ実行制御サービス/デーモンが開始したことを示します。
9）ジョブ実行制御サービス/デーモンが終了したことを示します。

注1）

依頼元サーバにおいて、ネットワークジョブ/分散実行ジョブを他のリモートサーバへ依頼した場合、「on host-name」が追加されて出力されます。任意のサブシステムへ依頼するネットワークジョブの場合、「:*n*」も追加されます。

注2）

実行サーバにおいて、ネットワークジョブ/分散実行ジョブを他のリモートサーバから受け付けた場合、「from host-name:*n*」が追加されて出力されます。

以下に可変部分について出力される内容を説明します。

jobname:

ジョブ名が出力されます。

jobno:

ジョブ実行制御が割り当てたジョブ番号が出力されます。

hostname:

ジョブを投入したホスト名が出力されます。

queuename:

ジョブが開始したキュー名が出力されます。

stepname:

ジョブステップ名が出力されます。
RC(rc)|EC(ec): RC(rc)またはEC(ec)のどちらかが出力されます。
強制終了した場合やネットワーク異常などにより、ジョブ処理が中断した場合は、EC(ec)が出力され、それ以外は、RC(rc)が出力されます。
rc : ジョブの終了コードが10進数で出力されます。
ec : 異常を示す終了コードが16進数で出力されます。ジョブを強制終了した場合、ecは40000222と出力されます。

host-name:*n*:

ジョブステップが開始したことを示すログの場合、ネットワークジョブ/分散実行ジョブの実行先サーバのホスト名が出力されます。任意のサブシステムへ依頼するネットワークジョブの場合、「:*n*」も追加されます。
「:*n*」の*n*は、実行先サーバのサブシステム番号です。
ジョブを受け付けたことを示すログの場合、ネットワークジョブ/分散実行ジョブの依頼元サーバのホスト名が出力されます。「:*n*」の*n*は、依頼元サーバのサブシステム番号です。
サブシステム運用をしていない場合、*n*は0になります。

7.5 qalter キューの属性変更コマンド

記述形式

qalter [-sys サブシステム番号] *queue-name parameter*
(注)

注)

“[-sys サブシステム番号]”はSystemwalker Operation Manager EEのオプションです。

機能説明

qalterは、キューの属性を変更するコマンドです。

以下の属性を変更することができます。

- 投入可能ジョブ数
- ジョブの多重度
- ジョブの経過時間制限値
- ジョブの実行優先順位
- ジョブの実行優先度

オプション

EE GEE

-sys サブシステム番号

複数サブシステム運用をしているシステムで、操作対象のサブシステムを指定するためのオプションです。操作対象とするサブシステム番号を、0から9の範囲で指定します。このオプションを省略した場合、サブシステム番号0が操作対象となります。

オペランド

queue-name

属性を変更するキューのキュー名を指定します。

parameter

変更する属性とその値を指定します。

本オペランドは、同時に複数指定することができます。

指定可能な属性とその値の範囲は以下のとおりです。

maxjob

投入可能ジョブ数を表します。範囲は1～999です。

maxexec

多重度(同時実行可能数)を表します。範囲は1～999です。

elapse

経過時間制限値を表します。範囲は1～99999999です。

qprty

実行優先順位を表します。範囲は0～63です。

dprty

実行優先度を表します。範囲は、Windows版の場合は0～4、UNIX版の場合は-20～19です。

指定する形式は、ジョブ実行制御の初期化パラメタと同じ形式です。

投入可能ジョブ数を5とする場合の指定を以下に示します。

```
maxjob=5
```

復帰値

復帰値	意 味
0	コマンド処理が正常に終了しました。
0以外	コマンド処理が正常に終了しませんでした。エラーメッセージが出力されます。

コマンド格納場所

Windows	Systemwalker Operation Managerインストールディレクトリ¥MpWalker.JM¥bin
Solaris HP-UX AIX Linux	/usr/lib/mjes

実行に必要な権限

システム管理者(Administratorsグループに属するユーザ/スーパーユーザ)権限が必要です。

注意事項

- 変更した属性の内容は、ジョブ実行制御を初期化モードで起動するまでの間、保持されます。ジョブ実行制御を初期化モードで起動するとキューの属性は、初期化ファイルに定義した内容に変更されます。

7.6 qjalter ジョブの属性変更コマンド

記述形式

```
qjalter [-sys サブシステム番号] -dp dispatch-priority job
qjalter [-sys サブシステム番号] -p priority job
qjalter [-sys サブシステム番号] -dp dispatch-priority -p priority job
                    (注)
```

注)

“[-sys サブシステム番号]”はSystemwalker Operation Manager EEのオプションです。

機能説明

qjalterは、ジョブの属性を変更するコマンドです。

以下の属性を変更することができます。

- ジョブの実行優先度
- ジョブの実行優先順位

オプション

EE GEE

-sys サブシステム番号

複数サブシステム運用をしているシステムで、操作対象のサブシステムを指定するためのオプションです。操作対象とするサブシステム番号を、0から9の範囲で指定します。このオプションを省略した場合、サブシステム番号0が操作対象となります。

-dp *dispatch-priority*

【Windows版】

ジョブから起動されるプログラムのスレッドの優先順位を0～4の整数で指定します。値が大きいほどスレッドの優先順位が高くなります。本オプションは、run状態、またはoph状態のジョブに使用することはできません。

【UNIX版】

ジョブから起動されるプロセスの実行優先度を-20～19の整数で指定します。実行優先度は実行時のCPU割当て順位を表し、19が最も低く、-20が最も高くなります。本オプションは、run状態、またはoph状態のジョブに使用することはできません。

-p *priority*

ジョブのキュー内の優先順位を0～63の整数で指定します。優先順位は0が最も低く、63が最も高くなります。本オプションは、run状態、またはoph状態のジョブに使用することはできません。

オペランド

job

属性を変更するジョブを指定します。

ジョブは以下の形式で指定します。

job-name(jobno.hostname)

job-nameは、ジョブ投入時に指定するジョブ名です。

jobnoは、ジョブ実行制御が割り当てるジョブ受付けの番号です。

hostnameは、ジョブを投入したホスト名です。hostnameが省略された場合、自ホスト名が指定されたものとみなします。

jobno.hostnameは、同一ジョブ名のジョブが複数個存在しない場合は、省略可能です。

ジョブ名にワイルドカードは指定できません。

ジョブを以下のように指定した場合は、正常となります。

- job-name
- job-name(jobno)
- job-name(jobno.hostname)
- jobno
- jobno.hostname

以下のように指定した場合は、エラーとなります。

- job-name(hostname)
- job-name(.hostname)
- (hostname)
- hostname

復帰値

復帰値	意 味
0	コマンド処理が正常に終了しました。
0以外	コマンド処理が正常に終了しませんでした。エラーメッセージが出力されます。

コマンド格納場所

Windows	Systemwalker Operation Managerインストールディレクトリ¥MpWalker.JM¥bin
Solaris HP-UX AIX Linux	/usr/lib/mjes

実行に必要な権限

システム管理者(Administratorsグループに属するユーザ/スーパーユーザ)権限が必要です。

7.7 qstart キューの起動コマンド

記述形式

```
qstart [-sys サブシステム番号] queue-name . . .
                    (注)
```

注)

“[-sys サブシステム番号]”はSystemwalker Operation Manager EEのオプションです。

機能説明

qstartは、停止しているキューを起動するコマンドです。

オプション

EE GEE

-sys サブシステム番号

複数サブシステム運用をしているシステムで、操作対象のサブシステムを指定するためのオプションです。操作対象とするサブシステム番号を、0から9の範囲で指定します。このオプションを省略した場合、サブシステム番号0が操作対象となります。

オペランド

queue-name

起動しようとするキューのキュー名を指定します。

復帰値

復帰値	意 味
0	コマンド処理が正常に終了しました。
0以外	コマンド処理が正常に終了しませんでした。エラーメッセージが出力されます。

コマンド格納場所

Windows	Systemwalker Operation Managerインストールディレクトリ¥MpWalker.JM¥bin
Solaris HP-UX AIX Linux	/usr/lib/mjes

実行に必要な権限

システム管理者(Administratorsグループに属するユーザ/スーパーユーザ)権限が必要です。

7.8 qstop キューの停止コマンド

記述形式

qstop [-sys サブシステム番号] *queue-name* . . .
(注)

注)

“[-sys サブシステム番号]”はSystemwalker Operation Manager EEのオプションです。

機能説明

qstopは、動作中のキューを停止するコマンドです。

オプション

EE GEE

-sys サブシステム番号

複数サブシステム運用をしているシステムで、操作対象のサブシステムを指定するためのオプションです。操作対象とするサブシステム番号を、0から9の範囲で指定します。このオプションを省略した場合、サブシステム番号0が操作対象となります。

オペランド

queue-name

停止しようとするキューのキュー名を指定します。

復帰値

復帰値	意 味
0	コマンド処理が正常に終了しました。
0以外	コマンド処理が正常に終了しませんでした。エラーメッセージが出力されます。

コマンド格納場所

Windows	Systemwalker Operation Managerインストールディレクトリ¥MpWalker.JM¥bin
Solaris HP-UX AIX Linux	/usr/lib/mjes

実行に必要な権限

システム管理者(Administratorsグループに属するユーザ/スーパーユーザ)権限が必要です。

注意事項

- ジョブが実行中の場合は、ジョブを強制終了するのではなく、ジョブの終了を待ち合わせます。

7.9 mjdjadd 分散実行の構成ホスト追加コマンド

記述形式

mjdjadd [-sys サブシステム番号] *hostgroup-name host-name [parameter]*
(注)

注)

“[-sys サブシステム番号]”はSystemwalker Operation Manager EEのオプションです。

機能説明

運用中に分散実行のホストグループに構成ホストを追加するコマンドです。

オプション

EE GEE

-sys サブシステム番号

複数サブシステム運用をしているシステムで、操作対象のサブシステムを指定するためのオプションです。操作対象とするサブシステム番号を、0から9の範囲で指定します。このオプションを省略した場合、サブシステム番号0が操作対象となります。

オペランド

hostgroup-name

構成ホストを追加したいホストグループ名を指定します。

host-name

追加するホスト名を指定します。すでに登録されているホスト名を追加することはできません。1つのホストグループ内に設定できるホストは100個までです。ホスト名に自ホストを指定しないでください。
実行サブシステム名を使用することはできません。

parameter

追加するホストの属性とその値を指定します。省略することもできます。指定可能な属性とその値の範囲、および省略値は以下のとおりです。

maxexec

ホストに設定するジョブの多重度(同時実行可能数)を表します。範囲は1～999です。省略値は10です。

ジョブの多重度を5とする場合の指定を以下に示します。

```
maxexec=5
```

復帰値

復帰値	意 味
0	コマンド処理が正常に終了しました。
0以外	コマンド処理が正常に終了しませんでした。エラーメッセージが出力されます。

コマンド格納場所

Windows	Systemwalker Operation Managerインストールディレクトリ¥MpWalker.JM¥bin
Solaris HP-UX AIX Linux	/usr/lib/mjes

実行に必要な権限

システム管理者(Administratorsグループに属するユーザ/スーパーユーザ)権限が必要です。

注意事項

- Systemwalker Operation Managerサーバ(分散元)でのみ実行できます。
- 分散実行の有効/無効は、運用中に変更することはできません。
- 定義の変更は以下の場合のみ有効です。
  - 分散実行キューとして動作している場合([キューの作成/編集]ウィンドウで、[分散実行機能を有効にする]をチェックしている場合)
  - ジョブ実行制御サービス/デーモンが起動中の場合
- 変更した内容は、ジョブ実行制御を初期化モードで起動するまでの間、保持されます。ジョブ実行制御を初期化モードで起動すると、分散実行の設定は、初期化ファイルに定義した内容に変更されます。
- 構成ホストに自ホストを追加する場合は、[運用情報の定義]ウィンドウから実施し、本コマンドを使用しないでください。分散実行ジョブが終了コード0x40000999や1などで異常終了する場合があります。

使用例

ホストグループ“djgroup”にホスト“djhost2”を追加する場合の例を以下に示します。

```
C:¥>mjdjadd djgroup djhost2 maxexec=5
hostgroup = djgroup
 hostname = djhost0
  maxexec = 20   state = ACT    run = 2
 hostname = djhost1
  maxexec = 7    state = ACT    run = 0
 hostname = djhost2
  maxexec = 5    state = ACT    run = 0
```

7.10　mjdjdel 分散実行の構成ホスト削除コマンド

記述形式

```
mjdjdel [-sys サブシステム番号] hostgroup-name host-name
                                (注)
```

注)

“[-sys サブシステム番号]”はSystemwalker Operation Manager EEのオプションです。

機能説明

運用中に分散実行のホストグループから構成ホストを削除するコマンドです。

オプション

EE GEE

-sys サブシステム番号

複数サブシステム運用をしているシステムで、操作対象のサブシステムを指定するためのオプションです。操作対象とするサブシステム番号を、0から9の範囲で指定します。このオプションを省略した場合、サブシステム番号0が操作対象となります。

オペランド

hostgroup-name

削除したいホストが所属するホストグループ名を指定します。

host-name

削除するホスト名を指定します。ホスト名に自ホストを指定しないでください。

復帰値

復帰値	意 味
0	コマンド処理が正常に終了しました。
0以外	コマンド処理が正常に終了しませんでした。エラーメッセージが出力されます。

コマンド格納場所

Windows	Systemwalker Operation Managerインストールディレクトリ¥MpWalker.JM¥bin
Solaris HP-UX AIX Linux	/usr/lib/mjes

実行に必要な権限

システム管理者(Administratorsグループに属するユーザ/スーパーユーザ)権限が必要です。

注意事項

- Systemwalker Operation Managerサーバ(分散元)でのみ実行できます。
- 分散実行の有効/無効は、運用中に変更することはできません。
- 定義の変更は以下の場合のみ有効です。
  - 分散実行キューとして動作している場合([キューの作成/編集]ウィンドウで、[分散実行機能を有効にする]をチェックしている場合)
  - ジョブ実行制御サービス/デーモンが起動中の場合
- 変更した内容は、ジョブ実行制御を初期化モードで起動するまでの間、保持されます。ジョブ実行制御を初期化モードで起動すると、分散実行の設定は、初期化ファイルに定義した内容に変更されます。
- ホストグループには最低1つのホストが設定されている必要があります。そのため、ホストグループに設定されているホストが1つしかない場合には、そのホストを本コマンドで削除することはできません。
- 実行中のジョブが残っている状態でホストを削除した場合、ホストは定義上削除され、それ以降、削除したホストにジョブは投入されません。ただし、実行中のジョブはそのまま継続されます。
- 構成ホストに定義した自ホストを削除する場合は、[運用情報の定義]ウィンドウから実施し、本コマンドを使用しないでください。分散実行ジョブが終了コード0x40000999や1などで異常終了する場合があります。

使用例

ホストグループ“djgroup”からホスト“djhost1”を削除する例を、以下に示します。

```
C:¥>mjdjdel djgroup djhost1
hostgroup = djgroup
  hostname = djhost0
    maxexec = 20  state = ACT    run = 2
  hostname = djhost2
    maxexec = 5   state = ACT    run = 0
```

7.11　mjdjalt 分散実行の設定変更コマンド

記述形式

```
mjdjalt [-sys サブシステム番号] hostgroup-name host-name parameter
                                (注)
```

注)

“[-sys サブシステム番号]”はSystemwalker Operation Manager EEのオプションです。

機能説明

運用中に分散実行の設定を変更するコマンドです。

オプション

EE GEE

-sys サブシステム番号

複数サブシステム運用をしているシステムで、操作対象のサブシステムを指定するためのオプションです。操作対象とするサブシステム番号を、0から9の範囲で指定します。このオプションを省略した場合、サブシステム番号0が操作対象となります。

オペランド

hostgroup-name

設定を変更したいホストグループ名を指定します。

host-name

設定を変更したい分散先ホスト名を指定します。

parameter

変更するホストの属性とその値を指定します。指定可能な属性とその値の範囲は以下のとおりです。

maxexec

ホストに設定するジョブの多重度(同時実行可能数)を表します。範囲は1～999です。

変更した多重度が、実行中のジョブ数より小さい値の場合、一時的に実行中のジョブ数が多重度を超えた状態となりますが、実行中のジョブが終了するにつれて、変更した多重度で制限されるようになります。

ジョブの多重度を3とする場合の指定を以下に示します。

```
maxexec=3
```

復帰値

復帰値	意 味
0	コマンド処理が正常に終了しました。
0以外	コマンド処理が正常に終了しませんでした。エラーメッセージが出力されます。

コマンド格納場所

Windows	Systemwalker Operation Managerインストールディレクトリ¥MpWalker.JM¥bin
Solaris HP-UX	/usr/lib/mjes

AIX Linux	

実行に必要な権限

システム管理者(Administratorsグループに属するユーザ/スーパーユーザ)権限が必要です。

注意事項

- Systemwalker Operation Managerサーバ(分散元)でのみ実行できます。
- 分散実行の有効/無効は、運用中に変更することはできません。
- 定義の変更は以下の場合のみ有効です。
  - 分散実行キューとして動作している場合([キューの作成/編集]ウィンドウで、[分散実行機能を有効にする]をチェックしている場合)
  - ジョブ実行制御サービス/デーモンが起動中の場合
- 変更した内容は、ジョブ実行制御を初期化モードで起動するまでの間、保持されます。ジョブ実行制御を初期化モードで起動すると、分散実行の設定は、初期化ファイルに定義した内容に変更されます。

使用例

ホストグループ“djgroup”のホスト“djhost1”に設定されたジョブの多重度を3に変更する例を、以下に示します。

```
C:¥>mjdjalt djgroup djhost1 maxexec=3
hostgroup = djgroup
  hostname = djhost0
    maxexec = 20  state = ACT    run = 2
  hostname = djhost1
    maxexec = 3   state = ACT    run = 0
  hostname = djhost2
    maxexec = 5   state = ACT    run = 0
```

7.12 qdel ジョブの削除コマンド

記述形式

```
qdel [-sys サブシステム番号] [-u user-name] job . . .
                (注)
```

注)

“[-sys サブシステム番号]”はSystemwalker Operation Manager EEのオプションです。

機能説明

qdelは、指定されたジョブを削除するコマンドです。

ジョブの削除は以下のユーザが可能です。

- ジョブを投入したユーザ
- システム管理者

オプション

EE GEE **-sys サブシステム番号**

複数サブシステム運用をしているシステムで、操作対象のサブシステムを指定するためのオプションです。操作対象とするサブシステム番号を、0から9の範囲で指定します。このオプションを省略した場合、サブシステム番号0が操作対象となります。

-u *user-name*

システム管理者が他者のジョブを削除する場合、ジョブ所有者のユーザ名を指定します。

なお、ユーザ名に空白を含む場合は、ユーザ名全体をダブルクォーテーション(")で囲ってください。【Windows版】

Operation Managerユーザを指定することはできません。【UNIX版】

オペランド

job

削除するジョブを指定します。

ジョブは以下の形式で指定します。

なお、各指定項目についての説明および注意事項については、“7.6 qjalter ジョブの属性変更コマンド”を参照してください。

job-name(jobno.hostname)

復帰値

復帰値	意 味
0	コマンド処理が正常に終了しました。
0以外	コマンド処理が正常に終了しませんでした。エラーメッセージが出力されます。

コマンド格納場所

Windows	Systemwalker Operation Managerインストールディレクトリ¥MpWalker.JM¥bin
Solaris	/opt/FJSVMJS/usr/bin
HP-UX	/opt/FHPMJS/usr/bin
AIX	/opt/FAIXMJS/usr/bin
Linux	/opt/FJSVMJS/usr/bin

注意事項

- ジョブ名“job-name(jobno.hostname)”を省略することはできません。

7.13 qhold ジョブの保留(ホールド)コマンド

記述形式

qhold [-sys サブシステム番号] [-u *user-name*] *job* . . . (注)

注)

“[-sys サブシステム番号]”はSystemwalker Operation Manager EEのオプションです。

機能説明

qholdは、指定されたジョブを保留(ホールド)するコマンドです。

ジョブのホールドは以下のユーザが可能です。

- ジョブを投入したユーザ
- システム管理者

ホールド状態のジョブを解除するためには、qrlsコマンドを使用します。

オプション

EE GEE

-sys サブシステム番号

複数サブシステム運用をしているシステムで、操作対象のサブシステムを指定するためのオプションです。操作対象とするサブシステム番号を、0から9の範囲で指定します。このオプションを省略した場合、サブシステム番号0が操作対象となります。

-u *user-name*

システム管理者が他者のジョブをホールドする場合、ジョブ所有者のユーザ名を指定します。ジョブの所有者がホールドする場合、このオプションを省略することができます。

Operation Managerユーザを指定することはできません。【UNIX版】

システム管理者によってホールドされたジョブは、たとえジョブの所有者であっても状態を解除することはできません。

なお、ユーザ名に空白を含む場合は、ユーザ名全体をダブルクォーテーション(")で囲ってください。【Windows版】

オペランド

job

ホールドするジョブを指定します。

ジョブは以下の形式で指定します。

なお、各指定項目についての説明および注意事項については、“7.6 qjalter ジョブの属性変更コマンド”を参照してください。

job-name(jobno.hostname)

復帰値

復帰値	意 味
0	コマンド処理が正常に終了しました。
0以外	コマンド処理が正常に終了しませんでした。エラーメッセージが出力されます。

コマンド格納場所

Windows	Systemwalker Operation Managerインストールディレクトリ¥MpWalker.JM¥bin
Solaris	/opt/FJSVMJS/usr/bin
HP-UX	/opt/FHPMJS/usr/bin
AIX	/opt/FAIXMJS/usr/bin

Linux	/opt/FJSVMJS/usr/bin

注意事項

ジョブ名“job-name(jobno.hostname)”を省略することはできません。

que状態のジョブのみ、ホールドすることができます。この時、ジョブの状態はhld状態になります。

その他の状態のジョブをホールドすることはできません。

7.14 qjstat ジョブ情報の表示コマンド

記述形式

qjstat [-sys サブシステム番号] [-u *user-name*] [-l] [-m] [-x] [*job* . . .]
(注)

注)

“[-sys サブシステム番号]”はSystemwalker Operation Manager EEのオプションです。

機能説明

qjstatは、以下のジョブ情報を表示するコマンドです。

- ジョブ名
- ジョブ番号
- キュー名
- ジョブの状態

本コマンドを投入したユーザがシステム管理者である場合は、以下の情報も表示します。

- クライアントホスト名
- クライアントユーザ名

表示の対象となるジョブは、コマンド実行者の権限によって以下のように異なります。

- システム管理者以外の場合

  投入したジョブのみ表示します。
- システム管理者が-uオプションを省略した場合

  すべてのジョブを表示します。

オプションを省略した場合、本コマンド実行者が投入したジョブの状態を表示します。

また、該当するジョブが存在しない場合は、何も表示しません。

オプション

EE GEE **-sys サブシステム番号**

複数サブシステム運用をしているシステムで、操作対象のサブシステムを指定するためのオプションです。操作対象とするサブシステム番号を、0から9の範囲で指定します。このオプションを省略した場合、サブシステム番号0が操作対象となります。

-m

コマンド実行者がシステム管理者の場合、下記のジョブ情報を表示します。

- ジョブプロセス番号

  ローカルジョブを実行したとき生成されたプロセスのプロセス番号です。ネットワークジョブ、ワークユニットジョブ、JCL、分散実行(実行先がリモートホストの場合)のプロセス番号は表示されません。

- 依頼ホスト名

  ネットワークジョブ、JCL、分散実行の依頼ホスト名です。実行サーバが二重化されている場合は、実行依頼されているホスト名です。
  実行サブシステム名を指定したジョブの場合、実行サブシステム名に対応するホスト名が表示されます。

-u *user-name*

システム管理者が特定のユーザのジョブの情報を表示する場合、ジョブ所有者のユーザ名を指定します。

なお、ユーザ名に空白を含む場合は、ユーザ名全体をダブルクォーテーション(")で囲ってください。【Windows版】

Operation Managerユーザを指定することはできません。【UNIX版】

-l

ジョブの詳細情報を表示します。

- ジョブの実行優先順位
- ジョブの実行優先度
- ジョブの実行が開始されてからの経過時間(実行中ジョブのみ)

本オプションの使用者がシステム管理者である場合は、以下の情報も表示します。

- ジョブの投入時刻
- ジョブの実行開始時刻(実行中ジョブのみ)

-x

実行中(run状態)ジョブの情報を表示します。

オペランド

job

情報を表示するジョブを指定します。

ジョブは以下の形式で指定します。

なお、各指定項目についての説明および注意事項については、“7.6 qjalter ジョブの属性変更コマンド”を参照してください。

job-name(jobno.hostname)

ジョブの状態

ジョブは、投入されてから実行が完了するまでに、各種状態に移行します。

以下にそれぞれの状態の意味を示します。

hld

　状態がホールドされ、ジョブが他の状態に移行できないことを示します。

que

　ジョブがrun状態に移行するために実行の準備、順番を待っていることを示します。

run

　ジョブが実行中であることを示します。

oph

出力が保留されていることを示します。

復帰値

復帰値	意 味
0	コマンド処理が正常に終了しました。
0以外	コマンド処理が正常に終了しませんでした。エラーメッセージが出力されます。

コマンド格納場所

Windows	Systemwalker Operation Managerインストールディレクトリ¥MpWalker.JM¥bin
Solaris	/opt/FJSVMJS/usr/bin
HP-UX	/opt/FHPMJS/usr/bin
AIX	/opt/FAIXMJS/usr/bin
Linux	/opt/FJSVMJS/usr/bin

表示例

システム管理者が、“-l”および“-m”オプションを指定した場合の表示例を以下に示します。

```
(1)  jobname = updatejob
(2)    jobno = 00300
(8)    clienthost = client1          clientuser = user1                                  (9)
(3)    queuename = queue1@host1
(4)    status    = run
(5)    qprty = 31                    input date = 2014.11.18 input time = 20:03:56     (10)
(6)    dprty =  2 elapsed =     1231 start date = 2014.11.18 start time = 20:03:57     (11)
(12)   jobprocid =
(13)   exechost  =
                                  ↑(7)
```

一般ユーザが、“-l”オプションを指定した場合の表示例を以下に示します。

```
jobname              jobno  queuename            status qprty dprty  elapsed
updatejob            00300  queue1@host1            run    31     2     1231
(1)                  (2)    (3)                     (4)    (5)    (6)      (7)
```

表示項目を以下に示します。

(1) ジョブ名(一般ユーザが実行した場合、ジョブ名の表示は20バイトまで)
(2) ジョブ番号
(3) キュー名
(4) ジョブの状態［run:実行中,que:実行待ち,hld:実行保留,oph:出力保留］
(5) ジョブの実行優先順位
(6) ジョブの実行優先度
(7) ジョブの実行が開始されてからの経過時間(実行中ジョブのみ)

(8) クライアントホスト名
(9) クライアントユーザ名
(10) ジョブの投入時刻
(11) ジョブの実行開始時刻(実行中ジョブのみ)
(12) ジョブプロセス番号
(13) 依頼ホスト名

注意事項(情報の表示順について)

- ジョブの状態をキーにしてソートします。同じ状態のジョブが複数存在している場合は、状態別にジョブ番号をキーにしてソートします。

7.15 qmove ジョブのキュー間移動コマンド

記述形式

```
qmove [-sys サブシステム番号] [-d] [-u user-name] [-z]job [queue]...
                    (注)
```

注)

"[-sys サブシステム番号]"はSystemwalker Operation Manager EEのオプションです。

機能説明

qmoveは、自サーバ上の指定されたジョブを、現在キューイングされているキューから指定されたキューに移動する(現在キューイングされているキューから削除し、キュー名で指定されたキューにキューイングしなおす)コマンドです。ジョブがque状態のとき、移動できます。

ジョブの移動は、以下のユーザが可能です。

- ジョブを投入したユーザ
- システム管理者

なお、以下の移動は本コマンドでは実行できません。

- 分散実行キューによって投入されたジョブの移動
- 分散実行キューへの移動

オプション

EE GEE **-sys サブシステム番号**

複数サブシステム運用をしているシステムで、操作対象のサブシステムを指定するためのオプションです。操作対象とするサブシステム番号を、0から9の範囲で指定します。このオプションを省略した場合、サブシステム番号0が操作対象となります。

-d

キュー名で指定されたキューに移動を試みるたびに移動結果を表示します。

-u *user-name*

システム管理者が他者のジョブの移動を行う場合、ジョブ所有者のユーザ名を指定します。

なお、ユーザ名に空白を含む場合は、ユーザ名全体をダブルクォーテーション(")で囲ってください。【Windows版】

Operation Managerユーザを指定することはできません。【UNIX版】

-z

ジョブの移動が成功しても、成功したことを知らせるメッセージを表示しません。

本オプションを省略した場合は、ジョブの移動が成功した時にジョブ名、移動先のキュー名を表示します。同時に-uオプションを指定した場合は、ジョブの所有者名も表示します。

移動に失敗した場合、または、移動時にエラーが発生した場合には、本オプションの指定の有無にかかわらずエラーメッセージを表示します。

オペランド

job

移動するジョブを指定します。

ジョブは以下の形式で指定します。

なお、各指定項目についての説明および注意事項については、“7.6 qjalter ジョブの属性変更コマンド”を参照してください。

job-name(jobno.hostname)

queue

queueには、移動先のキューを指定します。

キュー名にワイルドカードは指定できません。

本オペランドを省略した場合には、[運用情報の定義]ウィンドウの“ジョブ投入先のキューの省略値を指定する”(systemパラメタのdfltqueueオペランド)で指定された、デフォルトキューに対して移動を試みます。

キュー名は複数指定することができます。ジョブ実行制御は指定された順にキューに移動を試み、最初に見つけた移動可能なキューにジョブを移動します。ジョブの移動後、指定されたキューが残っていたとしても、それらのキューにはジョブを移動しません。

指定されたキューがすべて存在していない場合、または現在ジョブがキューイングされているキューが移動先として指定された場合、ジョブ実行制御は何も行いません。

復帰値

復帰値	意 味
0	コマンド処理が正常に終了しました。
0以外	コマンド処理が正常に終了しませんでした。エラーメッセージが出力されます。

コマンド格納場所

Windows	Systemwalker Operation Managerインストールディレクトリ¥MpWalker.JM¥bin
Solaris	/opt/FJSVMJS/usr/bin
HP-UX	/opt/FHPMJS/usr/bin
AIX	/opt/FAIXMJS/usr/bin
Linux	/opt/FJSVMJS/usr/bin

注意事項

- ジョブ名“job-name(jobno.hostname)”を省略することはできません。

記述例

システム管理者が“user1”のジョブ(ジョブ番号1)を“queue3”へ移動する場合(ただし、何らかの原因で、“queue3”への移動に失敗した場合は、“queue4”へ移動する)の記述例を以下に示します。

```
qmove -d -u user1 1 queue3 queue4
```

7.16　qrls ジョブの保留解除(リリース)コマンド

記述形式

```
qrls [-sys サブシステム番号] [-u user-name] job . . .
                (注)
```

注)

“[-sys サブシステム番号]”はSystemwalker Operation Manager EEのオプションです。

機能説明

qrlsは、指定されたジョブの保留解除(ホールド[hld状態]の解除)を行うコマンドです。

ジョブのリリースは以下のユーザが可能です。

- ジョブを投入したユーザ
- システム管理者

本コマンドでジョブをリリースすると、ジョブはホールドされる前の状態(que状態)になります。

オプション

EE GEE

-sys サブシステム番号

複数サブシステム運用をしているシステムで、操作対象のサブシステムを指定するためのオプションです。操作対象とするサブシステム番号を、0から9の範囲で指定します。このオプションを省略した場合、サブシステム番号0が操作対象となります。

-u *user-name*

システム管理者がジョブをリリースする場合、ジョブ投入者のユーザ名を指定します。

なお、ユーザ名に空白を含む場合は、ユーザ名全体をダブルクォーテーション(")で囲ってください。【Windows版】

Operation Managerユーザを指定することはできません。【UNIX版】

オペランド

job

リリースするジョブを指定します。

ジョブは以下の形式で指定します。

なお、各指定項目についての説明および注意事項については、“7.6 qjalter ジョブの属性変更コマンド”を参照してください。

```
job-name(jobno.hostname)
```

復帰値

復帰値	意 味
0	コマンド処理が正常に終了しました。
0以外	コマンド処理が正常に終了しませんでした。エラーメッセージが出力されます。

コマンド格納場所

Windows	Systemwalker Operation Managerインストールディレクトリ¥MpWalker.JM¥bin
Solaris	/opt/FJSVMJS/usr/bin
HP-UX	/opt/FHPMJS/usr/bin
AIX	/opt/FAIXMJS/usr/bin
Linux	/opt/FJSVMJS/usr/bin

注意事項

ジョブ名“job-name(jobno.hostname)”を省略することはできません。

システム管理者がホールドしたジョブは、たとえそのジョブの所有者であってもqrlsで解除できません。これはシステム管理者がジョブ実行制御運用のためにジョブをホールドしても、ジョブの所有者が簡単に解除できてしまうと、運用に支障が出るおそれがあるからです。

7.17 qstat キューの状態表示コマンド

記述形式

```
qstat [-a ][-sys サブシステム番号] [<queuename(s)>...]
                (注)
```

注)

“[-sys サブシステム番号]”はSystemwalker Operation Manager EEのオプションです。

機能説明

qstatは、以下のキュー情報を表示するコマンドです。

- キュー名
- キューの状態
- 実行優先順位
- 実行優先度
- 経過時間制限値
- 多重度
- 投入可能ジョブ数
- 実行中ジョブ数
- キューイング中ジョブ数
- 保留中ジョブ数
- デフォルトキュー名（-aオプション指定時）
- システム多重度（-aオプション指定時）

・ 優先順位の省略値（-aオプション指定時）

オプション

EE GEE

-sys サブシステム番号

複数サブシステム運用をしているシステムで、操作対象のサブシステムを指定するためのオプションです。操作対象とするサブシステム番号を、0から9の範囲で指定します。このオプションを省略した場合、サブシステム番号0が操作対象となります。

-a

下記のキュー情報も追加して表示します。

・ デフォルトキュー名

・ システム多重度

・ 優先順位の省略値

オペランド

<queuename(s)>

キューの情報を表示するキュー名を15バイト以内で指定します。

queuenameが指定された場合、指定されたキューに関する情報のみ表示します。

queuenameが省略された場合、全キューに関する情報を表示します。

キュー名にワイルドカードは指定できません。

キューの状態

キューには、以下の状態があります。

ACT

　キューは動作中です。ジョブの受付、ジョブの実行が可能です。

INACT

　キューは停止中です。ジョブの受付は可能ですが実行はできません。

復帰値

復帰値	意 味
0	コマンド処理が正常に終了しました。
0以外	コマンド処理が正常に終了しませんでした。エラーメッセージが出力されます。

コマンド格納場所

Windows	Systemwalker Operation Managerインストールディレクトリ¥MpWalker.JM¥bin
Solaris	/opt/FJSVMJS/usr/bin
HP-UX	/opt/FHPMJS/usr/bin
AIX	/opt/FAIXMJS/usr/bin
Linux	/opt/FJSVMJS/usr/bin

表示例

qstat キューの情報表示コマンドの表示例を以下に示します。

```
queuename      status  qprty  dprty  elapsed  maxexec  maxjobs  run  que  hold
queue1@host1       ACT        31     0        0       10       999    0    0    0
queue2@host1       ACT        31     0        0       10       999    0    0    0
queue3@host1       ACT        31     0        0       10       999    0    0    0
queue4@host1       ACT        31     0        0       10       999    0    0    0
queue5@host1       ACT        31     0        0       10       999    0    0    0
online1@host1      ACT        31     0        0       30       999    0    0    0
```

“-a”オプション指定時の表示例を以下に示します。併せて表示項目との対応を説明します。

```
      (11)                (12)           (13)

defqueue:queue1@host1   maxexec:30      defprty:31
queuename             status qprty dprty  elapsed maxexec maxjobs    run    que    hold
online1@host1         ACT       31    2        0      30      -1      0      0      0
queue1@host1          ACT       31    2        0     300      -1      0      0      0
queue2@host1          ACT       31    2        0      10      -1      0      0      0
queue3@host1          ACT       31    2        0      10      -1      0      0      0
queue4@host1          ACT       31    2        0      10      -1      0      0      0
queue5@host1          ACT       31    2        0      10      -1      0      0      0

  (1)                 (2)       (3)   (4)      (5)    (6)     (7)     (8)    (9)   (10)
```

表示項目を以下に示します。

(1) キュー名
(2) キューの状態［ACT:稼働,INACT:停止］
(3) 実行優先順位
(4) 実行優先度
(5) 経過時間制限値
(6) 多重度［-1:制限していない］
(7) 投入可能ジョブ数［-1:制限していない］
(8) 実行中ジョブ数
(9) キューイング中ジョブ数
(10) 保留中ジョブ数
(11) デフォルトキュー名［not specify:指定していない］
(12) システム多重度［-1:制限していない］
(13) 優先順位の省略値［-1:指定していない］

7.18 qsub ジョブの投入コマンド

記述形式

qsub [*options*] [*job_file*] [*parameters*]

機能説明

qsubはジョブを投入するコマンドです。

オプション (*options*)

qsubコマンドで指定可能なオプションを以下に示します。

EE GEE **-sys サブシステム番号**

複数サブシステム運用をしているシステムで、操作対象のサブシステムを指定するためのオプションです。操作対象とするサブシステム番号を、0から9の範囲で指定します。このオプションを省略した場合、サブシステム番号0が操作対象となります。

-C *comment*

ジョブにコメントを付加します。511バイト以内で指定します。コメントに空白文字を含む場合、コメントを半角のダブルクォーテーション(")で囲む必要があります。以下に例を示します。

```
-C "This is a comment."
```

本オプションの指定がない場合、ジョブ実行制御は環境変数JC_COMMENTを検索します。この環境変数が存在していれば、JC_COMMENTで記述されている文字列をコメントとみなします。

コメント内にダブルクォーテーションを記述する場合は、ダブルクォーテーションの直前に“¥”を指定することでダブルクォーテーションを文字として扱うことが可能です。なお、直前に“¥”を記述したダブルクォーテーションは、コメント全体を括るためのダブルクォーテーションとして扱われません。以下に例を示します。

```
-C "This is a ¥"client listing¥""
```

-ch *client-hostname*

ジョブの依頼元のホスト名を特定するためのクライアントホスト名を指定します。

クライアントホスト名の最大は、64バイトです。

JCLファイルを投入する場合、JCLのhost文の指定は無効になり、本オプションの指定が有効になります。

-cu *client-username*

ジョブの依頼元のユーザ名を特定するためのクライアントユーザ名を指定します。クライアントユーザ名の最大は64バイトです。

JCLファイルを投入する場合、JCLのuser文の指定は無効になり、本オプションの指定が有効になります。

このオプションは、システム管理者(Administratorsグループに属するユーザ/スーパーユーザ)のみ指定できます。

-dp *dispatch-priority*

【Windows版】

ジョブから起動されるプログラムのスレッドの優先順位を、0～4の整数で指定します。値が大きいほどスレッドの優先順位が高くなります。

JCLファイルを投入する場合、JCLのdprty文の指定は無効になり、本オプションの指定が有効になります。

本オプションが省略された場合、[キューの作成/編集]ウィンドウで指定する“ジョブ実行優先度を指定する”(queueパラメタのdfltdprtyオペランド)の定義値が設定されます。ただし、JCLファイルでdprty文が指定された場合、dprty文の指定が有効になります。

【UNIX版】

起動するプロセスの実行優先度を-20～19の整数で指定します。実行優先度は実行時のCPU割当て順位を表し、19が最も低く、-20が最も高くなります。

JCLファイルを投入する場合、JCLのdprty文の指定は無効になり、本オプションの指定が有効になります。

本オプションが省略された場合、[キューの作成/編集]ウィンドウで指定する“ジョブ実行優先度を指定する”(queueパラメタのdfltdprtyオペランド)の定義値が設定されます。ただし、JCLファイルでdprty文が指定された場合、dprty文の指定が有効になります。

-e [*drive-letter*:] [*path-name*] *stderr-file*【*Windows版*】

-e [*path-name*] *stderr-file*【*UNIX版*】

ジョブの実行で生じる標準エラー出力をpath-nameに指定したパス名にstderr-fileというファイル名で出力します。

path-nameの指定がない場合、パス名はジョブを投入した時のカレントワーキングディレクトリとみなします。

指定されたパスに出力できない場合、カレントワーキングディレクトリに出力します。

本オプションが省略された場合、標準エラー出力ファイル名は“ジョブ名.eジョブ番号”になります。ジョブ名がKEISAN、ジョブ番号が20であるジョブの標準エラー出力ファイル名は、“KEISAN.e20”になります。

本オプションは-eoオプションと同時に指定することはできません。

-env *環境変数設定ファイル名* [-jo *位置番号*]

qsubコマンドでジョブを投入する際、ジョブに与える任意の環境変数を指定できます。

ジョブ投入API(Mp_SubmitJob)では利用できません。

ジョブに渡す環境変数は、あらかじめ“7.18.1 ジョブ実行制御の環境変数設定ファイル”の形式で任意のファイル内に指定し、ジョブ投入時に環境変数設定ファイルの名前を指定します。

また、１つの環境変数設定ファイル内に、複数のパターンの環境変数を指定することも可能であり、ジョブ投入時に、何番目のパターンの環境変数を使用するのかを指定することができます。

- 1ジョブ当たりに指定できる環境変数の数は、32個までです。
- 各環境変数に指定できる環境変数名は、64バイトまでです。
- 各環境変数に指定できる環境変数値は、256バイトまでです。
- １つの環境変数設定ファイル内に指定できるパターン数は、255個までです。

環境変数設定ファイル名

ジョブに任意の環境変数を与える場合に指定します。

-envオプションを指定した場合、本オプションの直後に環境変数設定ファイル名を指定する必要があります。環境変数設定ファイル名を指定しない場合やファイル名に誤りがある場合、ジョブは投入されません。

環境変数設定ファイルは、ジョブに与える任意の環境変数名と環境変数値を格納したファイルです。記述形式は、“7.18.1 ジョブ実行制御の環境変数設定ファイル”を参照してください。

位置番号

-envオプションで指定した環境変数格納ファイル内の何個目のパターンを使用するのかを本オプションで指定します。本オプションが省略された場合、１個目のパターンが指定されたと見なされます。

-eo

標準エラー出力を標準出力ファイルに出力します。

-eオプションと同時に指定することはできません。

-ld *HHMM*

ジョブの打ち切り時刻を指定します。本オプションを指定したジョブは、打ち切り時刻を迎えると強制終了します。終了コードは“0x40000322”です。

“HH”は“00～23”(時)、“MM”は“00～59”(分)の表記です。

本オプションは-ltオプションおよび[登録－ジョブ]ウィンドウ－[詳細情報]シートの[実行予測時間]と同時に指定できません。

-j *job-name*

ジョブ名を指定します。job-nameの長さには制限はありませんが、有効となるのは先頭64バイトまでです。空白、半角カナ、ユーザ定義文字および以下の半角文字はジョブ名として使用できません。

/ ? * ; , ¥ " < > | : ( ) ' . $ `

1文字目に # を指定することはできません。

また、ネットワークジョブおよび分散実行ジョブの場合は、& を指定することはできません。

JCLファイルを投入する場合、JCLのjobstart文におけるジョブ名の指定は無効になり、本オプションの指定が有効になります。

本オプションが省略された場合、オペランドjob_fileに指定するジョブファイル名(ただし、パス名および、拡張子を取り除いたもの)をジョブ名とみなします。ただし、JCLファイルでjobstart文にジョブ名が指定された場合、jobstart文の指定が有効になります。

ジョブファイル名も指定しない場合、STDINをジョブ名とみなします。

job-nameの先頭に数字が指定された場合、ジョブ実行制御はジョブ名の先頭に“j”を付加します。

UTF-8コードの環境(ネットワークジョブの場合、投入サーバまたは実行サーバのいずれかがUTF-8コードの環境)では日本語コードを使用しないでください。

UTF-8コードの環境(ネットワークジョブの場合、投入サーバまたは実行サーバのいずれかがUTF-8コードの環境)で、オペランドjob_fileに指定するジョブファイル名に日本語を含む場合、以下に従ってください。

- job_fileがJCLの場合、jobstart文のジョブ名に日本語コードを含まない文字列を指定し、本オプションを省略してください。

  または、本オプションに日本語コードを含まない文字列を指定してください(このとき、jobstart文のジョブ名の指定は任意です)。

- job_fileがJCLでない場合、本オプションに日本語コードを含まない文字列を指定してください。

-lt *limittime*

ジョブ実行経過時間の制限値を指定します。

経過時間は秒単位で、1～99999999の範囲の値を指定します。

本オプションが省略された場合、[キューの作成/編集]ウィンドウで指定する“ジョブ経過時間を制限する”(queueパラメタのlimittimeオペランド)での定義値が設定されます。

本オプションは-ldオプションと同時に指定できません。

-no

本オプションを指定した場合、ジョブの結果ファイル(ジョブの標準出力ファイル/エラー出力ファイル)を生成しません。

本オプションを指定した場合、標準出力/標準エラー出力を指定ファイルに出力するような-oオプション、-eオプション、-eoオプションは、同時に指定できません。

-nr

ジョブの再実行を禁止します。

本オプションを付けて投入されたジョブは、そのジョブが実行中にシステムダウンが発生した場合、ジョブ実行制御の再起動時に再実行されません。

ただし、ネットワークジョブおよび分散実行ジョブは、実行サーバでシステムダウンが発生した場合、ジョブの状態や本オプションの有無にかかわらず、ジョブ実行制御の再起動時に再実行はされません。

-o [*drive-letter*:] [*path-name*] *stdout-file*【*Windows版*】

-o [*path-name*] *stdout-file*【*UNIX版*】

ジョブの実行で生じる標準出力をpath-nameに指定したパスにstdout-fileというファイル名で出力します。

path-nameの指定がない場合、パス名はジョブを投入した時のカレントワーキングディレクトリとみなします。

指定されたパスに出力できない場合、カレントワーキングディレクトリに出力します。

本オプションが省略された場合、標準出力ファイル名は“ジョブ名.oジョブ番号”になります。ジョブ名がKEISAN、ジョブ番号が20であるジョブの標準出力ファイル名は、“KEISAN.o20”になります。

-ofe

MJ_OUTPUTFILES環境変数で指定した出力ファイルが実行サーバに存在しなかった場合にジョブをエラー終了扱い(終了コードとして0x40000008を設定)とする場合に指定します。

-ofeオプションを指定しない場合、MJ_OUTPUTFILES環境変数で指定した出力ファイルが実行サーバ上に存在しなくても、ジョブの終了コードに0x40000008を設定しません。ジョブの終了コードは、実行サーバ上におけるジョブの終了コードが採用されます。

MJ_OUTPUTFILES環境変数については、“7.18.2 転送する入出力ファイルの指定方法”を参照してください。

-p *priority*

ジョブのキュー内のプライオリティを0～63の整数で指定します。

プライオリティは0が最も低く、63が最も高くなります。

本オプションで指定したプライオリティは、キュー上のジョブの実行順位を決めるためのもので、実行プライオリティを決定するものではありません。

ジョブをキューイングする場合、ジョブ実行制御は、キュー内に存在するジョブのキュー内プライオリティを調べます。新たに投入されたジョブは、キューイングされているプライオリティの低いジョブの前にキューイングされます。

また、新規ジョブと同じプライオリティのジョブが存在する場合、そのジョブの後ろに新規ジョブをキューイングします。

JCLファイルを投入する場合、JCLのqprty文の指定は無効になり、本オプションの指定が有効になります。

本オプションが省略された場合、[運用情報の定義]ウィンドウの“優先順位の省略値を指定する”(queueパラメタのdfltprtyオペランド)の定義値が設定されます。定義されていない場合には、-1が表示されます。ただし、JCLファイルでqprty文が指定された場合、qprty文の指定が有効になります。

-q *queue-name*

ジョブをキューイングするキュー名を指定します。

本オプションの指定がない場合、ジョブ実行制御は環境変数QSUB_QUEUEを検索します。この環境変数が存在していれば、QSUB_QUEUEで記述されている文字列をキュー名とみなします。

本オプションの指定がなく、環境変数QSUB_QUEUEも存在していない場合、システム管理者がデフォルトキューを設定していれば、そのキューにキューイングされます。

本オプションの指定がなく、環境変数QSUB_QUEUEもなく、デフォルトキューも設定されていない場合、“not specify”と表示され、ジョブはキューイングされません。

-R *resource-name*[,*resource-type*]

ジョブが使用するリソース名および排他属性を指定します。resource-nameは最大64バイトで、半角英数字、“_”（半角アンダーバー）、“-”（半角ハイフン）を利用することが可能です。resource-typeには以下のいずれかを指定します。

sh

リソースを他のジョブと共有します。

ex

リソースを占有します。

resource-nameとresource-typeの間の区切りには“,”を用います。

resource-typeが省略された場合はshが指定されたものとします。

リソース名 RDBを、排他属性を占有とする場合の指定を以下に示します。

```
-R RDB,ex
```

JCLファイルを投入する場合、JCLのresource文の指定は無効になり、本オプションの指定が有効になります。

-rh *hostname*

本オプションを指定したジョブは指定されたサーバ上で実行されます。ジョブの標準出力ファイルおよび標準エラー出力ファイルは、投入元サーバに作成されます。

実行サブシステム名を使用することもできます。実行サブシステム名については“20.8.1 実行サブシステム名定義ファイル”を参照してください。

なお、本オプションを指定するときは以下の点に注意してください。

- ジョブがJCLだった場合は本オプションは無視されます。
- 本オプションを指定して分散実行キューにジョブを投入した場合、投入が拒否されます。

ジョブが実行されるキューは以下の手順で決定されます。

1. 投入元ジョブの実行キュー(投入時に指定したキュー名、または省略値)と同一名のキュー名。
2. 同一のキュー名が存在しない場合は、ジョブを依頼する他サーバで[運用情報の定義]ウィンドウの“ジョブ投入先のキューの省略値を指定する”(systemパラメタのdfltqueueオペランド)に定義されたキュー名。
3. [運用情報の定義]ウィンドウで“ジョブ投入先のキューの省略値を指定する”(systemパラメタのdfltqueueオペランド)が指定されていない場合は、“not specify”と表示され、ジョブの投入に失敗します。

また、本オプションを指定して、ネットワークジョブとして他サーバへジョブを依頼する場合、以下のa.b.またはc.に該当する場合はネットワークジョブを実行することができません。

a. 以下の条件にすべて該当している場合、ネットワークジョブを実行できません。
   - ジョブの投入元サーバ、実行サーバ(本オプションで指定されたサーバ)ともにWindowsサーバのネットワークジョブである
   - 実行サーバ上で[運用情報の定義]ウィンドウの“ジョブを所有者の権限で実行する”(execasuserパラメタ)が指定されている
   - ジョブ投入者のユーザ名が“root”である
   - 上記のジョブ投入者(root)のユーザパスワードと、実行サーバ上の“Systemwalker MpMjes”サービスのログオンアカウントのユーザのパスワードが異なる

b. 以下の条件にすべて該当している場合、ネットワークジョブを実行できません。
   - ジョブの投入元サーバがUNIXサーバ、実行サーバがWindowsサーバのネットワークジョブである
   - 実行サーバ上で[運用情報の定義]ウィンドウの“ジョブを所有者の権限で実行する”(execasuserパラメタ)が指定されている

c. 分散実行キューにジョブを投入した場合、ネットワークジョブを実行できません。

なお、a.のジョブ投入者とは“root”というユーザ名であり、システム管理者とは限りません。

また、ジョブ投入者とは、以下を指しています。

- ジョブスケジューラから実行された場合のプロジェクトの所有者
- qsubコマンドの“-cu”オプションで指定したクライアントユーザ名(オプション省略時は“qsub”コマンドの投入ユーザ)
- ジョブ投入APIを使用しているアプリケーションを実行したユーザ

-rsys 実行先サブシステム番号

ネットワークジョブの実行先サーバのサブシステム番号を指定します。実行先サーバの、指定したサブシステム番号のサブシステムにジョブが投入されます。実行先サブシステム番号には0～9を指定します。

自ホストと異なるホストのサブシステムにネットワークジョブを依頼するときは、-rsysオプションと同時に、-rhオプションを指定してください。自ホストのサブシステムにネットワークジョブを依頼するときは、-rhオプションを指定する必要はありません。

例えば、依頼ホスト“Ahost”のサブシステム1に、ジョブファイル“job.bat”を渡してジョブを依頼するときは、以下のように指定します。

```
qsub -rh Ahost -jft -rsys 1 job.bat
```

本オプションを指定したジョブは、分散実行キュー、旧互換負荷分散キューに投入できません。投入しようとするとエラーになります。

-rhオプションに実行サブシステム名を指定した場合は、本オプションは無視されて、実行サブシステム名に対応するサブシステム番号にジョブが投入されます。

-jft

ジョブファイルを-rhオプションで指定したサーバまたは、分散実行サーバへ転送して実行します。本オプションを指定しない場合は、-rhオプションで指定したサーバ上、または分散実行サーバ上に存在するジョブファイルを実行します。

本オプションは以下の条件を満たした場合にのみ有効です。

- ジョブファイルがJCL以外である。
- -rhオプションを指定している。または、分散実行キューにジョブを投入するとき。

-x

ジョブ投入時に存在するすべての環境変数をジョブ実行環境で利用できるようにします。

本オプションは-rhオプションと同時に指定しないでください。

環境変数を実行サーバに引き継ぐには、本オプションの代わりに、-envオプションを使用し、必要な環境変数を設定してください。

-z

ジョブを投入した時、メッセージを表示しません。

本オプションを指定した場合、ジョブの投入が成功してもジョブの投入が成功したことを知らせるメッセージを表示しません。

ただし、ジョブの投入が失敗した場合、本オプションの指定の有無に関係なく、常にエラーメッセージを表示します。

本オプションを省略した場合、ジョブの投入が成功した時に以下に示す形式のメッセージを表示します。

```
Job job submitted to queue: queue-name
```

jobは、以下の形式で表示されます。

なお、各表示項目についての説明および注意事項については、“7.6 qjalter ジョブの属性変更コマンド”を参照してください。

```
job-name(jobno.hostname)
```

queue-nameは、ジョブがキューイングされたキュー名を示します。

-r *retrycount*

ネットワークジョブの実行依頼中に回線異常が発生した場合にリトライを行います。この時のリトライ回数を、1～20の範囲で指定します。

リトライの間隔のデフォルト値は、300秒です。[運用情報の定義]ウィンドウ－[ネットワーク]シートの[ネットワークジョブで、接続エラー時のリトライの動作を変更する] が有効の場合は、指定しているリトライの間隔で動作します。

なお、ジョブの実行依頼処理が終了した後、ジョブの実行完了を待っている間に回線異常が発生した場合、本オプションによるリトライは行われません。

オペランド

job_file

実行するジョブファイルを指定します。指定可能なファイルは以下のとおりです。

- バッチファイル/シェルスクリプト

  拡張子が「BAT」または「CMD」のファイルです。【Windows版】

- 実行プログラム

  拡張子が「EXE」または「COM」のファイルです。【Windows版】

- JCLファイル

  ジョブ実行制御が提供する業務手続言語(JCL)によって作成した、拡張子が「JCL」のファイルです。

- Systemwalkerスクリプトファイル

  Systemwalkerスクリプト言語によって作成した、拡張子が「SWT」のファイルです。

UTF-8コードの環境(ネットワークジョブの場合、投入サーバまたは実行サーバのいずれかがUTF-8コードの環境)で、指定するファイル名に日本語を含む場合は、必ず-jオプションを併用して日本語コードを含まないジョブ名になるようにしてください。

parameters

バッチファイル/シェルスクリプト、実行プログラム、SystemwalkerスクリプトまたはJCLに渡すパラメタを指定します。

半角空白を含む文字列をパラメタとして指定する場合は、1つのパラメタとする文字列全体をダブルクォーテーション(")で囲む必要があります。

ジョブファイルにjob.bat、そのファイルに渡すパラメタとして0822および1000を指定する場合は、以下のように記述します。

```
qsub job.bat 0822 1000
```

なお、パラメタに*?[]などのシェルによって特別な意味を持つ文字を使用する場合は、文字列全体を一対の引用符(''または"")で囲んでください。ジョブファイルに/usr/bin/ls、パラメタとして-l、/home/guest/*および/home/guest/work/???を指定する場合は、以下のように記述します。【UNIX版】

```
qsub /usr/bin/ls -l '/home/guest/*' '/home/guest/work/???'
```

復帰値

復帰値	意 味
0	コマンド処理が正常に終了しました。
0以外	コマンド処理が正常に終了しませんでした。エラーメッセージが出力されます。

コマンド格納場所

Windows	Systemwalker Operation Managerインストールディレクトリ¥MpWalker.JM¥bin
Solaris	/opt/FJSVMJS/usr/bin
HP-UX	/opt/FHPMJS/usr/bin
AIX	/opt/FAIXMJS/usr/bin
Linux	/opt/FJSVMJS/usr/bin

注意事項

- job_fileを省略した場合、標準入力ファイル(STDIN)からジョブとして実行されるコマンドを読み込みます。標準入力がコンソールである場合、コマンドの入力を終了するには以下を入力します。

  【Windows版】

  ```
  [Ctrl-Z] + [Enter]
  ```

  【UNIX版】

  ```
  [Ctrl-D]
  ```

- 標準入力ファイル(STDIN)からジョブとして実行されるコマンドを読み込んだときは以下の点に注意してください。

  ー UNIX版で実行した場合、-rhオプションで指定するサーバにはUNIX版のSystemwalker Operation Managerサーバ機能がインストールされている必要があります。

  ー Windows版で実行した場合、-rhオプションで指定するサーバにはWindows版のSystemwalker Operation Managerサーバ機能がインストールされている必要があります。

- job_fileは、qsubコマンドを投入した時に、ジョブ実行制御のスプールへ作業ファイルとしてコピーされる場合があります([運用情報の定義]ウィンドウの[旧互換]シートの[ジョブ登録時に指定したスクリプトファイルをスプールに複写し、実行する]を有効にした場合)。その場合は、ジョブ実行制御は、その作業ファイルを参照してジョブを実行するため、キューイングした後にjob_fileの内容を変更してもその変更内容は、ジョブの実行には反映されません。

- コマンド行に指定可能なオプションは、すべてバッチファイルの最初のコメントブロックに記述することができます。【Windows版】

  バッチファイルに記述されたオプションと同じオプションをコマンド行で指定した場合、コマンド行で指定されたオプションとその引数が優先されます。

- job_fileにジョブ実行制御のqsubコマンドやジョブ実行制御のqsubコマンドを呼び出しているバッチファイル/シェルスクリプトを指定しないでください。

- ユーザ名に半角空白を含むユーザから、qsubコマンドを実行しないでください。また、-cuオプションに半角空白を含むユーザ名を指定しないでください。

- 以下のディレクトリ名またはファイル名に、半角空白を使用しないでください。

  ー job_fileオペランドに指定するジョブファイル名

  ー -eオプションに指定するパス名およびファイル名

  ー -oオプションに指定するパス名およびファイル名

  なお、PATH環境変数からパスを読み込む場合も、パス名に半角空白を設定しないでください。

- 本コマンドをSystemwalker Operation Managerのジョブから実行した場合、Systemwalker Operation Manager/Systemwalker Centric Managerの動作は保証されません。詳細については、“Systemwalker Operation Manager 運用ガイド”の“ジョブとして実行してはいけないコマンド”を参照してください。

- [運用情報の定義]ウィンドウの[旧互換]シートの[ジョブ登録時に指定したスクリプトファイルをスプールに複写し、実行する]がチェックされている場合、ジョブとして投入できるバッチファイル/シェルスクリプトの一行の長さは2050バイトまでです。

バッチファイルについて【Windows版】

バッチファイル内にオプションを記述することで、qsubコマンド投入時にオプションを指定する必要がなくなります。

ただし、[運用情報の定義]ウィンドウー[旧互換]シートの[ジョブ登録時に指定したスクリプトファイルをスプールに複写し、実行する]がチェックされていない場合にはオプション識別は実行されません。

オプション識別時のアルゴリズム

1. バッチファイルを1行読み込む。ファイルが終わりなら8.へ。
2. 空白行の場合、または最初に現れた文字が“:”の場合、1.へ。
3. 最初に現れた単語が“rem”でなければ8.へ。
4. “rem”の次に現れた文字が“@$”でなければ1.へ。
5. “@$”の直後に続く文字が“-”でなければ8.へ。
6. オプションの解析を行う。
7. 次の“-”文字を検索する。“-”が出現したら6.へ。行の最後まで出現しなければ1.へ。
8. 終了。これ以降のオプションは無視する。

オプション記述例

```
rem バッチファイル 例
rem
rem ジョブをサブミットするキューをCLASSAにする
rem @$-q CLASSA
rem
rem これ以降のオプションは無視
rem @$
```

7.18.1 ジョブ実行制御の環境変数設定ファイル

ファイル名

任意

使用用途

qsubコマンドの-env 環境変数設定ファイル名で呼び出し、ジョブに与える任意の環境変数を指定します。

格納場所

任意

ファイル形式

１つの環境変数格納ファイル内に複数のパターンの環境変数を定義できます。

１つのパターンの区切りとして、先頭行の先頭カラムから[JOB*n*]の形式で記述が必要です。*n*は、1～255の範囲の値を記述します。

記述ミスやメンテナンス性を高めるために、[JOB1]から開始して、昇順に記述することを推奨します。

環境変数の定義は、環境変数名の定義行と環境変数値の定義行の2行が1対の構成となります。

記述例を以下に示します。

```
[JOB1]
ENVNAME1=DATA1                    －＋            －＋
ENVVALUE1=/export/200401/data001－＋2行で1対      ｜
ENVNAME2=DATA2                                    ｜1つめのパターン
ENVVALUE2=/export/200401/data002                  －＋
[JOB2]
ENVNAME1=DATA1
ENVVALUE1=/export/200402/data001
ENVNAME2=DATA2
ENVVALUE2=/export/200402/data002
```

パラメタ

[JOB*n*]

パターンの先頭行に必ず記述する必要があります。

- 先頭カラムから記述します。
- *n*は、数字で、1～255の範囲で指定します。記述する順番などに制約はありません。
- qsubコマンドの-joオプションで指定するパターン番号が見つからない場合、-joオプションを省略した場合で、[JOB1]が存在しない場合は、エラーとなります。
- 同じパターン番号が存在した場合、先頭から検索して最初に見つかったパターン番号の部分に記述された環境変数情報が採用されます。
- qsubコマンドの投入時には、環境変数設定ファイル内すべてのフォーマットがチェックされるわけではありません。指定されたパターン番号に該当する部分のみを読み込み、必要な環境変数情報を読み込みます。したがって、[JOB*n*]の*n*の値は1～255の範囲外であったり、数字以外であったり、または省略されていても、エラーにはなりません。

ENVNAME*m*=環境変数名

環境変数を設定したい環境変数名を指定します。

- ENVNAMEは固定で、*m*の部分を1～昇順に変化させて定義します。*m*の最大値は32です。

  *m*の部分を1～昇順に検索して、次の番号が見つからなくなった時点で検索を終了し、そこまでに記述されていた環境変数情報が有効となります。例えば、*m*の部分を1,2,3,5 の順番で記述した場合、1,2,3の情報は有効となりますが、5の情報は無視されます。
- 最初にENVNAME1が存在しない場合は、エラーとなります。
- 後述のENVVALUEの*m*と一致させることで環境変数名と環境変数値が結合されます。
- 環境変数名は、最大64バイトまで指定できます。
- 環境変数名に指定不可文字はありません。
- 環境変数名は、OSで設定される環境変数や他の「ENVNAMEm」の値と重複しないように指定します。

ENVVALUE*m*=環境変数値

ENVNAME*m* で指定した環境変数に設定した環境変数値を指定します。

- ENVVALUE は固定で、*m*の部分をENVNAME*m*と一致する数字を指定します。*m*の最大値は32です。
- ENVNAME*m*と一致するENVVALUE*m*が存在しない場合、エラーとなります。
- 環境変数値は、最大256バイトまで指定できます。
- 環境変数値に指定不可文字はありません。

使用例

環境変数 DATA1 に、環境変数値として/export/200401/data001 を設定する場合、以下のように記述します。

```
ENVNAME1=DATA1
ENVVALUE1=/export/200401/data001
```

7.18.2　転送する入出力ファイルの指定方法

入出力ファイルを転送するには、qsubコマンドを実行するユーザの環境変数に入力ファイルと出力ファイル名を指定します。入出力ファイルを転送するための環境変数の指定方法について説明します。

qsubを実行するユーザに以下の環境変数を指定することで入出力ファイルを転送します。

１ジョブ当たり、入力ファイル、出力ファイルともに32個までの転送指定が可能です。

指定する個々のファイルのファイル名長は、最大で64バイトです。

qsubコマンド起動時のカレントディレクトリ配下のファイルの送受信が可能です。転送ファイルのフルパス指定はできません。

ジョブ投入API(Mp_SubmitJob)では利用できません。

環境変数名

環境変数	意 味
MJ_INPUTFILES	実行サーバへ送信する入力ファイルを指定します
MJ_OUTPUTFILES	実行サーバから受信する出力ファイルを指定します

環境変数値の指定形式

環境変数には、転送の対象とするファイル名を指定します。フルパス指定はできません。

ファイル名は最大で32個まで指定でき、カンマで区切って指定します。

１つのファイル名長の最大は64バイトです。

OSのファイルシステムが許可していない文字以外に、2バイト文字、半角空白および以下の文字をファイル名に含むことができません。

? * , ¥ " < > | : ( ) & =

環境変数は以下のように指定してください。

```
ファイル名１,ファイル名２,ファイル名３,・・・
```

入力ファイル

実行サーバで実行するアプリケーションに必要な入力ファイルを準備します。

入力ファイルは、以下のディレクトリ配下に配置する必要があります。

```
qsubコマンド起動時のカレントディレクトリ
```

上記のディレクトリ配下に存在しないファイルを実行サーバに転送することはできません。

MJ_INPUTFILESに指定したファイルは、実行サーバ上の“ジョブ作業用ディレクトリ”配下に配置(転送)されます。

ジョブの実行が終了すると、実行サーバ上の“ジョブ作業用ディレクトリ”配下に配置(転送)した入力ファイルは削除されます。

出力ファイル

実行サーバで実行するアプリケーションが出力したファイルをスケジュールサーバに転送する場合、出力ファイルは“ジョブ作業用ディレクトリ”直下に出力するようアプリケーション側の考慮が必要です。

上記のディレクトリ配下に存在しないファイルを実行サーバからスケジュールサーバへ転送することはできません。

MJ_OUTPUTFILESに指定したファイルは、該当のジョブをqsubコマンドで起動した際のカレントディレクトリ配下に配置(転送)されます。すでにファイルが存在していた場合には上書きされます。

ジョブの実行が終了すると、実行サーバ上の“ジョブ作業用ディレクトリ”配下に出力された出力ファイルは削除されます。

MJ_OUTPUTFILES環境変数で指定された出力ファイルが実行サーバ上に存在しなかった場合の動作をqsubコマンドの-ofeオプションで選択することができます。-ofeオプションの詳細は、“7.18 qsub ジョブの投入コマンド”を参照してください。

ジョブ作業用ディレクトリ

ジョブプロセスに設定された以下の環境変数に設定されます。

JC_TRANSFILEDIR

入出力ファイルの転送を実施していない場合、上記の環境変数は設定されません。

7.19 mjdjstat 分散実行の状態表示コマンド

記述形式

mjdjstat [-sys サブシステム番号] [*hostgroup-name*...] (注)

注)

“[-sys サブシステム番号]”はSystemwalker Operation Manager EEのオプションです。

機能説明

mjdjstatは、以下の分散実行情報を表示するコマンドです。

- ホストグループ名
- ホスト名
- ホストの実行多重度
- ホストの状態
- ホストの実行中ジョブ数

オプション

EE GEE

-sys サブシステム番号

複数サブシステム運用をしているシステムで、操作対象のサブシステムを指定するためのオプションです。操作対象とするサブシステム番号を、0から9の範囲で指定します。このオプションを省略した場合、サブシステム番号0が操作対象となります。

オペランド

hostgroup-name

状態を表示するホストグループ名を指定します。複数指定する場合は、半角空白で区切ります。ワイルドカードは使用できません。

hostgroup-nameが指定された場合、指定されたホストグループに関する情報のみ表示します。hostgroup-nameが省略された場合、すべてのホストグループに関する情報を表示します。

ホストの状態

ホストには以下の状態があります。

ACT

ジョブを投入する分散実行先サーバの対象とみなされるホストの状態です。以下のいずれかの状態になっているホストが対象です。

- ジョブ実行制御の運用が開始されてから、実行先として一度もジョブが投入されていない
- 前回ジョブを投入した結果、投入できた
- 前回ジョブの投入を試みた結果、DOWN状態だったが、DOWNを検知してから約10分以上経過した

DOWN

ジョブを投入する分散実行先サーバの対象とみなされないホストの状態です。ジョブの投入を試みた場合に、分散実行先サーバがダウンしていることを検知した時は、最大10分間この状態になります。

NA

本コマンドを実行したサーバのジョブ実行制御が停止中のため、状態が取得できません。

復帰値

復帰値	意 味
0	コマンド処理が正常に終了しました。
0以外	コマンド処理が正常に終了しませんでした。エラーメッセージが出力されます。

コマンド格納場所

Windows	Systemwalker Operation Managerインストールディレクトリ¥MpWalker.JM¥bin
Solaris	/opt/FJSVMJS/usr/bin
HP-UX	/opt/FHPMJS/usr/bin
AIX	/opt/FAIXMJS/usr/bin
Linux	/opt/FJSVMJS/usr/bin

表示例

ホストグループ“djgroup1”と“djgroup2”の情報を表示する例を、以下に示します。

```
C:¥>mjdjstat djgroup1 djgroup2
hostgroup = djgroup1
  hostname = djhost11
    maxexec = 20  state = ACT    run = 2
  hostname = djhost12
    maxexec = 3   state = ACT    run = 0
  hostname = djhost13
    maxexec = 5   state = ACT    run = 0
hostgroup = djgroup2
  hostname = djhost21
    maxexec = 10  state = ACT    run = 1
  hostname = djhost22
    maxexec = 10  state = ACT    run = 1
```

注意事項

- Systemwalker Operation Managerサーバ(分散元)でのみ実行できます。
- 本コマンドを実行したサーバのジョブ実行制御サービス/デーモンが停止中でも、その時点での定義の状態は参照できます。ただし、ホストの状態は“NA”、ホストの実行中ジョブ数は“0”となります。

7.20 mjsmdset 起動モード指定コマンド

記述形式

```
mjsmdset [-sys サブシステム番号] [-c｜-w]
                (注)
```

注)

“[-sys サブシステム番号]”はSystemwalker Operation Manager EEのオプションです。

機能説明

次回のジョブ実行制御の運用を開始するとき、またはデーモン/サービスを起動するときの起動モードを指定できます。

オプションを省略した場合は、-wオプションが指定されたものとみなします。

起動モードについては、“Systemwalker Operation Manager 運用ガイド”を参照してください。

オプション

EE GEE

-sys サブシステム番号

複数サブシステム運用をしているシステムで、操作対象のサブシステムを指定するためのオプションです。操作対象とするサブシステム番号を、0から9の範囲で指定します。このオプションを省略した場合、サブシステム番号0が操作対象となります。

-c

次回のデーモン/サービス起動時に初期化モードで起動します。

-w

次回のデーモン/サービス起動時に継続モードで起動します。

復帰値

復帰値	意 味
0	コマンド処理が正常に終了しました。
0以外	コマンド処理が正常に終了しませんでした。エラーメッセージが出力されます。

コマンド格納場所

Windows	Systemwalker Operation Managerインストールディレクトリ¥MpWalker.JM¥bin
Solaris HP-UX AIX Linux	/usr/lib/mjes/bin

実行に必要な権限

システム管理者(Administratorsグループに属するユーザ/スーパーユーザ)権限が必要です。

7.21 mjrmtjob リモートマシン上のコマンド/シェルスクリプト実行コマンド

記述形式

【Windows版】

```
mjrmtjob.exe -h 実行ホスト名 [-p OS種別] [-l 文字コード] [-u ユーザ名] [-P ポート番号] -d カレントディレクトリ "実行コマンド パラメタ…"
```

【UNIX版】

```
mjrmtjob -h 実行ホスト名 [-p OS種別] [-l 文字コード] [-u ユーザ名] [-P ポート番号]
-d 'カレントディレクトリ' '実行コマンド パラメタ…'
```

機能説明

Systemwalker Operation Managerがインストールされていないリモートマシン上に配置されたコマンドまたはスクリプトを実行するコマンドです。

オプション

-h 実行ホスト名

リモートマシンのホスト名を指定します。64バイト以内で指定します。

-p OS種別

リモートマシンのOS名です。

Linux、SolarisまたはWindowsのいずれかを指定します。大文字小文字は区別しません。

省略した場合、mjrmtjobコマンドを実行したサーバのOS種別になります。

-l 文字コード

リモートマシンの文字コードを指定します。“ASCII”、“SJIS”、“EUC”、“UTF-8”のいずれかを指定します。OSにより指定できる文字コードが決まっています。

本指定に従って、コマンドおよびスクリプトのパラメタ、標準出力、標準エラー出力の文字コード変換を行います。省略した場合、または、指定可能な文字コード以外を指定した場合は、本コマンドを実行するサーバの文字コードで動作(文字コード変換しない)します。

以下にリモートマシンのOS別に指定可能な文字コードを示します。大文字・小文字は区別しません。

OS種別	指定可能文字コード
Windows	ASCII、または、SJIS
Solaris	ASCII、SJIS、EUC、または、UTF-8
Linux	ASCII、または、UTF-8

-u ユーザ名

リモートマシン上で、コマンドまたはスクリプトを実行するユーザ名を指定します。

64バイト以内で指定します。Active Directoryを利用している場合は、ローカルアカウントは指定できません。省略した場合、ジョブの実行ユーザになります。

-P ポート番号

リモートマシンに接続する際のSSHのポート番号を指定します。1～65535の間の数値を指定してください。省略した場合は“22”で動作します。

-d カレントディレクトリ【Windows版】

リモートマシン上のコマンドまたはスクリプトを実行するカレントディレクトリを指定します。64バイト以内のフルパスで指定します。

-d 'カレントディレクトリ'【UNIX版】

リモートマシン上のコマンドまたはスクリプトを実行するカレントディレクトリをシングルクォーテーション(')で囲んで指定します。64バイト以内のフルパスで指定します。

オペランド

"実行コマンド パラメタ…"【Windows版】

ジョブとしてリモートマシン上で実行するコマンドまたはスクリプトを指定します。続いてパラメタを指定します。
全体をダブルクォーテーション(")で囲んでください。

'実行コマンド パラメタ…'【UNIX版】

ジョブとしてリモートマシン上で実行するコマンドまたはスクリプトを指定します。続いてパラメタを指定します。
全体をシングルクォーテーション(')で囲んでください。

復帰値

復帰値	意 味
201～256	mjrmtjobコマンドが異常終了しました。または、ジョブスケジューラが返すジョブの終了コードです。 復帰値については、“22.1 ジョブスケジューラの終了コード”を参照してください。
上記以外	実行したコマンドの復帰値になります。

コマンド格納場所

Windows	Systemwalker Operation Managerインストールディレクトリ¥MpWalker.JM¥bin
Solaris Linux	/usr/bin

注意事項

- 本コマンドの実行には、リモートマシン用のライセンスが必要です。
- 本コマンドは、Systemwalker Operation Managerのジョブとして登録された場合にだけ実行することができます。コマンドプロンプトから直接実行するなど、Systemwalker Operation Managerのジョブ以外の方法で実行することはできません。
- [登録－ジョブ]ウィンドウで本コマンドを指定する際、[ディレクトリ]欄、[依頼ホスト名]欄、[実行ユーザ]欄などの指定は、すべてmjrmtjobコマンド自身に対する指定になります。リモートマシン上で実行するコマンドまたはスクリプトに対する指定は、mjrmtjobコマンドのパラメタとして[コマンド]欄、または[パラメタ]欄で指定してください。
- 本コマンドの各オプションに指定する引数に使用できる文字の範囲は、以下のとおりです。

リモートマシンのOS種別	指定可能文字
Windows	英数字、記号(¥.:-_) “¥”はホスト名、ユーザ名には指定不可
Solaris Linux	英数字、記号(/.-_) “/”はホスト名、ユーザ名には指定不可 “:”はホスト名にのみ指定可能

- リモートマシンがダウンした場合、以下のファイルが残存することがあります。リモートマシン起動時に手動で削除してください。

リモートマシンのOS種別	リモートマシン上のファイル(注)
Windows	カレントディレクトリ配下の以下のファイル

リモートマシンのOS種別	リモートマシン上のファイル(注)
	・ ホスト名_プロセスID.vbs ・ ホスト名_プロセスID.ucmd.bat ・ ホスト名_プロセスID.killpid.bat ・ ホスト名_プロセスID.stdlog ・ ホスト名_プロセスID.ret ・ ホスト名_プロセスID.errlog
Solaris Linux	カレントディレクトリ配下の以下のファイル ・ ホスト名_プロセスID.sh ・ ホスト名_プロセスID.ucmd.sh ・ ホスト名_プロセスID.killpid ・ ホスト名_プロセスID.stdlog ・ ホスト名_プロセスID.ret

(注)ファイル名先頭の“ホスト名”はmjrmtjobコマンドを実行するサーバのホスト名

- オペランドで指定された実行コマンドをリモートマシン上(Unixの場合はシェル、Windowsの場合はVBスクリプト)でSSH通信によりリモート実行します。このため、制御が戻ってこないなど、この実行で問題があるようなコマンドまたはシェルスクリプトは実行しないでください。この場合の動作は保証されません。以下は問題があるコマンドの一例です。

  - exitコマンド
  - デーモンを起動するコマンド
  - yesコマンド

  など

- 実行継続モードでの実行の継続はできません。
- 本コマンド実行時に以下のメッセージが前回履歴に出力される場合がありますが、SSHの初回接続時に出力されるものであり、問題はありません。

```
Warning: Permanently added ホスト名 (RSA) to the list of known hosts.
```

- リモートマシンのOSがSolaris/Linuxの場合はシェル、Windowsの場合はバッチファイルとして実行するため、パラメタに文字としてダブルクォーテーションを指定する場合(Windows版)、または、シングルクォーテーションを指定する場合(UNIX版)を除いて、オペランドの指定形式はその仕様に依存します。
  実行コマンドは、オペランド全体を囲むダブルクォーテーション(Windows版)、またはシングルクォーテーション(UNIX版)を除いた指定形式が、リモートマシン上で実行可能であるように指定してください。

使用例【Windows版】

- Linuxサーバ“HOSTB”で“/user/command/keisan”コマンドを、パラメタに“aaa bbb ccc”を指定して、ユーザ“UserA”で“/work”をカレントディレクトリとして実行する場合の例：

```
mjrmtjob.exe -h HOSTB -p Linux -l UTF-8 -u UserA -d /work ”/user/command/keisan aaa bbb ccc”
```

- 実行コマンドのパラメタに、文字としてダブルクォーテーション(")を指定したい場合、文字として使用したいダブルクォーテーション(")の前に、円記号(¥)を3個付加してください。
  実行コマンドのパラメタに“東京 A" ccc”を指定して実行する場合の例：

```
mjrmtjob.exe -h HOSTB -p Linux -l UTF-8 -u UserA -d /work ”/user/command/keisan 東京 A¥¥¥” ccc”
```

- 実行コマンドのパラメタにリモートマシンの環境変数を展開したものを指定する場合、以下のように環境変数を“%”で囲んで指定します。

実行コマンドのパラメタに環境変数XYZを指定して実行する場合の例：
（リモートマシンの環境変数XYZの値は“東京”とします）

```
mjrmtjob.exe -h HOSTC -p Windows -l SJIS -u UserA -d d:¥work ”d:¥user¥command¥keisan %XYZ% aaa”
```

リモートマシン上での実行時は、以下のようになります。

```
d:¥user¥command¥keisan 東京 aaa
```

- 実行コマンドのパラメタに“%”で囲まれた文字列を文字列として指定する場合、“%”を“%”でエスケープします。以下のように指定します。
実行コマンドのパラメタに“%XYZ%”の文字列を指定して実行する場合の例：

```
mjrmtjob.exe -h HOSTC -p Windows -l SJIS -u UserA -d d:¥work ”d:¥user¥command¥keisan %%XYZ%% aaa”
```

リモートマシン上での実行時は、以下のようになります。

```
d:¥user¥command¥keisan %XYZ% aaa
```

使用例【UNIX版】

- Linuxサーバ“HOSTB”で“/user/command/keisan”コマンドを、パラメタに“aaa bbb ccc”を指定して、ユーザ“UserA”で“/work”をカレントディレクトリとして実行する場合の例：

```
mjrmtjob -h HOSTB -p Linux -l UTF-8 -u UserA -d ’/work’ ’/user/command/keisan aaa bbb ccc’
```

- 実行コマンドのパラメタに、文字としてシングルクォーテーション(')を指定したい場合は、以下のようにしてください。

  1. 実行コマンドの前からパラメタに含まれるシングルクォーテーションの直前までをシングルクォーテーションで囲むように指定します。
  2. パラメタに含むシングルクォーテーションを“¥”でエスケープした上で、ダブルクォーテーションで囲んで指定します。なお、リモートマシンがWindowsの場合、エスケープは不要です。
  3. その後にパラメタが続く場合は、再びシングルクォーテーションで囲んで指定します。

  Linuxサーバ“HOSTB”で“/user/command/keisan”コマンドを、パラメタに“東京 A' ccc”を指定して、ユーザ“UserA”で“/work”をカレントディレクトリとして実行する場合の例：

```
mjrmtjob -h HOSTB -p Linux -l UTF-8 -u UserA -d ’/work’ ’/user/command/keisan 東京 A’”¥’ ”’ ccc’
```

7.22 mjkeygen キーペア作成コマンド【Windows版】

記述形式

```
[作成の場合]
mjkeygen create -k 公開鍵格納ディレクトリ [-u ユーザ名]
[削除の場合]
mjkeygen delete [-u ユーザ名]
[出力の場合]
mjkeygen getkey -k 公開鍵格納ディレクトリ [-u ユーザ名]
[確認の場合]
mjkeygen chkkey [-u ユーザ名]
```

機能説明

mjrmtjob(リモートマシン上のコマンド/シェルスクリプト実行コマンド)で使用するキーペア(公開鍵/秘密鍵)を作成/設定・削除します。

createおよびgetkeyの場合は、公開鍵を指定されたディレクトリに出力します。

なお、RSA以外の鍵(ECDSA)は、MJKEYGEN_KEYTYPE環境変数を設定することで、作成することができます。

オプション

create

本コマンドの実行ユーザ、または-uオプションで指定したユーザのキーペアを作成してシステムに設定し、公開鍵(ユーザ名.mjeskey.pub)を出力します。

-k 公開鍵格納ディレクトリ

公開鍵の格納ディレクトリを64バイト以内のフルパスで指定します。
半角英数字、記号(¥.:-_)が使用できます。

-u ユーザ名

キーペアの対象のユーザ名を指定します。任意のユーザを64バイト以内で指定します。半角英数字、記号(.:-_)が使用できます。
本コマンドをシステム管理者(Administratorsグループに属するユーザ)が実行する場合のみ指定可能です。
Active Directoryを利用している場合は、ローカルアカウントは指定できません。
また、ユーザプリンシパル名を指定することはできません。
本オプション省略時、キーペアの対象ユーザは、コマンドの実行ユーザになります。

delete

システムに設定されたキーペアを削除します。

getkey

システムにキーペアが設定されている場合に公開鍵(ユーザ名.mjeskey.pub)を出力します。公開鍵を誤って削除した場合などに利用します。

chkkey

システムにキーペアが設定されているか確認します。

環境変数

RSA以外の鍵(ECDSA)を作成する場合に指定します。

環境変数	値（鍵の種類）
MJKEYGEN_KEYTYPE	以下のいずれかを指定します。 ・ ecdsa256 ・ ecdsa384 ・ ecdsa521

復帰値

復帰値	意 味
0	コマンド処理が正常に終了しました。
0以外	コマンド処理が正常に終了しませんでした。エラーメッセージが出力されます。

コマンド格納場所

Windows	Systemwalker Operation Managerインストールディレクトリ¥MpWalker.JM¥bin

注意事項

- 本コマンドは、mjrmtjobコマンドでリモートマシン上に配置されたコマンドまたはスクリプトを実行する場合に、事前設定で使用するコマンドです。それ以外の用途には使用できません。
- キーペアが既に設定されている状態で、再度、本コマンドをcreateで実行した場合、エラーとなります。更新したい場合は、deleteで削除してからcreateを実行してください。
- Systemwalker Operation Manager環境設定クライアントの[運用情報の定義]ウィンドウにおいて、[ジョブを所有者の権限で実行する]を指定していないサブシステムについては、システム管理者でmjkeygenコマンドを実行して、ジョブ実行制御のログオンアカウント(デフォルトはSYSTEM)に対応するキーペアを作成して利用してください。

使用例

[一般ユーザが実行]

- キーペアを作成して設定する例

```
mjkeygen create -k D:¥temp
```

- キーペアを削除する例

```
mjkeygen delete
```

- 既に設定済みの公開鍵を取得する例

```
mjkeygen getkey -k D:¥temp
```

- キーペアが設定済みか確認する例

```
mjkeygen chkkey
```

[システム管理者が実行]

- システム管理者がユーザ:SYSTEMのキーペアを作成し設定する例（RSAの場合）

```
mjkeygen create -k D:¥temp -u SYSTEM
```

- システム管理者がユーザ:SYSTEMのキーペアを作成し設定する例（ECDSA256の場合）

```
SET MJKEYGEN_KEYTYPE=ecdsa256
mjkeygen create -k D:¥temp -u SYSTEM
```

- システム管理者がユーザ:SYSTEMのキーペアを削除する例

```
mjkeygen delete -u SYSTEM
```

- システム管理者が既に設定済みのユーザ:SYSTEMの公開鍵を取得する例

```
mjkeygen getkey -k D:¥temp -u SYSTEM
```

- システム管理者がユーザ:SYSTEMのキーペアが設定済みか確認する例

```
mjkeygen chkkey -u SYSTEM
```

出力例

- キーペアが設定済みの状態で、“mjkeygen chkkey”を実行した場合

```
exist
```

・ キーペアが未設定の状態で、“mjkeygen chkkey”を実行した場合

```
not exist
```

第8章 アクション管理コマンド【Windows版】

本章では、Systemwalker Operation Managerが提供するアクション管理コマンドについて説明します。なお、コマンドは、Systemwalker Operation Managerのサーバ機能がインストールされているサーバで起動してください。

注意

WindowsのUAC機能を有効にしている環境でコマンドを実行する場合、OSにログインしたユーザの権限によって権限の昇格を求めるダイアログが表示され、対処が必要となります。

詳細は、“Systemwalker Operation Manager 運用ガイド”の、“Windowsでコマンドを実行する場合の注意事項【Windows版】”を参照してください。

8.1 Mp_SpAct アクション停止コマンド

記述形式

```
Mp_SpAct -s
```

機能説明

音声通知(メッセージ読上げ、WAVファイル再生)を停止します。

終了コード

0	正常に停止しました。
4	音声通知中ではありません。
8	アクション管理サービスは起動されていません。
32	パラメタエラーです。
64	その他のエラーです。

実行に必要な権限/実行環境

管理者権限で起動したコマンドプロンプト上で本コマンドを実行してください。

8.2 Mp_SpSnd 音声停止コマンド

記述形式

```
Mp_SpSnd
```

機能説明

音声停止コマンドは、再生中の音声通知(メッセージ読上げ、WAVファイル再生)を停止します。なお、本コマンドは音声通知が実行されているクライアントで実行します。

復帰値

0	正常に停止しました。
4	音声通知中ではありません。
8	アクション実行サーバは起動されていません。

32	パラメタエラーです。
64	その他のエラーです。

コマンド格納場所

Systemwalker Operation Managerインストールディレクトリ¥MpWalker¥bin

実行に必要な権限/実行環境

- 一般ユーザ用コマンド
- 管理者権限で起動したコマンドプロンプト上で本コマンドを実行してください。

使用例

```
Mp_SpSnd
```

実行結果／出力形式

なし

第9章 業務連携コマンド【Windows版】

本章では、Systemwalker Operation Managerが提供する業務連携コマンドについて説明します。なお、コマンドは、Systemwalker Operation Managerのサーバ機能がインストールされているサーバで起動してください。

注意

WindowsのUAC機能を有効にしている環境でコマンドを実行する場合、OSにログインしたユーザの権限によって権限の昇格を求めるダイアログが表示され、対処が必要となります。

詳細は、“Systemwalker Operation Manager 運用ガイド”の、“Windowsでコマンドを実行する場合の注意事項【Windows版】”を参照してください。

9.1 サーバ業務連携コマンド

サーバ業務連携コマンドは、サーバの業務システムの構築を支援するためのコマンドです。

参考

サーバ業務連携コマンドは、バッチファイルを作成して登録することも可能です。

9.1.1 EXECASVR.EXE アプリケーション起動コマンド

記述形式

```
EXECASVR.EXE [/UN:UserName] [/LG:LoginGroup]
        [/RC:nn /RI:nnnn] /AE:ExecuteAppli
        [AppliParam]
```

機能説明

サーバにインストールされている任意のアプリケーションを起動します。起動アプリケーションは、バックグラウンドで動作し、画面には表示されません。また、起動したアプリケーションは、終了の待ち合わせは行いません。

パラメタ

/UN:UserName

指定したアプリケーションをどのユーザの権限で起動するかを指定します。

UserNameは、パスワード管理簿にあらかじめ設定したユーザ名を指定します。UserNameに空白文字を含む場合は、ダブルクォーテーション(")で囲んでください。

UserNameは、“ドメイン名¥ユーザ名”または“サーバ名¥ユーザ名”のいずれかの形式で指定してください。ドメイン名またはサーバ名は半角英数字15文字以内で、ユーザ名は半角英数字20文字以内で指定してください。

本パラメタを省略した場合は、システム権限でアプリケーションは起動されます。

/LG:LoginGroup

パスワード管理簿のログイン定義名を指定します。

LoginGroupは、256文字以内で指定し、空白文字を含む場合はダブルクォーテーション(")で囲んでください。

/UNパラメタを指定した場合は、本パラメタも指定してください。

/RC:*nn*

アプリケーションの起動でエラーが発生した場合のリトライ回数を指定します。

nnに0～99の整数を指定します。それ以外の値を設定すると、エラーになります。省略した場合はリトライを行いません。

/RI:*nnnn*

エラーが発生した場合のリトライ間隔を指定します。

nnnnに0～9999の整数を秒単位で指定します。それ以外の値を設定すると、エラーになります。省略値は0です。

本パラメタは、/RCパラメタで1以上を指定した場合に有効となります。

/AE:*ExecuteAppli*

起動アプリケーション名をローカルドライブパスで指定します。ExecuteAppliに、アプリケーション名を指定します。起動アプリケーション名に空白がある場合は、ダブルクォーテーション(")で囲んでください。ExecuteAppliに指定できる文字数はAppliParamを含めて511文字までです。

AppliParam

ExecuteAppliで指定したアプリケーションに対するパラメタを指定します。パラメタを複数指定する場合は、それぞれの間に空白文字を入れてください。各パラメタに空白文字を含む場合は、その両端をダブルクォーテーション(")で囲んでください。

復帰値

復帰値	意 味
0	正常終了
0以外	異常終了

コマンド格納場所

Systemwalker Operation Managerインストールディレクトリ¥MpWalker.JM¥bin

記述例

“c:¥Program files¥app¥App.exe”を“foo¥Usr1”の権限で起動します。サーバログオン時に使用するユーザのパスワードはパスワード管理簿のログイン定義名“grp1”から獲得します。

```
EXECASVR.EXE /UN:foo¥Usr1 /LG:grp1 /AE:"c:¥Program files¥app
¥App.exe"
```

処理結果

コマンドの処理が終了すると、終了時の情報を標準出力に出力します。

出力される内容を以下に示します。

```
[処理開始日時] YYYY-MM-DD HH:MM:SS
[起動アプリケーション] 起動アプリケーション名
[リトライ回数] リトライ回数
[発生したエラー] エラーメッセージ
[処理終了日時] YYYY-MM-DD HH:MM:SS
```

[発生したエラー]には、正常終了の場合は“なし”が出力されます。強制終了やエラーの場合はメッセージが出力されます。メッセージ内容については“Systemwalker Operation Manager メッセージ集”を参照してください。

アプリケーション起動を行う前にエラーが発生した場合は、以下の形式でメッセージが出力されます。

```
[処理開始日時]　YYYY-MM-DD HH:MM:SS
[コマンドエラー]　エラーメッセージ
[処理終了日時]　YYYY-MM-DD HH:MM:SS
```

メッセージ内容については“Systemwalker Operation Manager メッセージ集”を参照してください。

使用上の注意

- 本コマンドを使用する場合は、サーバ上で業務連携サービスを動作させておいてください。
- 事前にパスワード管理コマンドを使って、パスワード管理簿の設定を行ってください。詳細は“9.3.1 LGMANAGE.EXE パスワード管理コマンド”を参照してください。
- 本コマンドは、起動したアプリケーションの処理を監視しません。処理結果を知る必要がある場合は、アプリケーション自身に終了コードの通知機能を用意してください。
- 本コマンドにより起動したアプリケーションは、画面を表示しないため、画面表示を伴うアプリケーションは起動できません。また、ユーザのレジストリを必要とするアプリケーションも起動できません。
- /UN:UserNameおよび/LG:LoginGroupを指定する場合、あらかじめUserNameに指定するユーザに対して、[コントロールパネル]の[管理ツール]－[ローカル セキュリティ ポリシー]などを使用し、“サービスとしてログオン”の権限を与えてください。
- 本コマンドを強制終了した場合、起動したアプリケーションに対しての強制終了を行うことはできません。
- コマンドの強制終了は、アプリケーションの起動が完了した時点で行われます。このため、起動に時間がかかるアプリケーションの場合、コマンドの終了を待たされる場合があります。
- 本コマンドでは、キーボードからの入力を必要とするアプリケーションは起動できません。

9.1.2　FCOPYSVR.EXE ファイル転送コマンド

記述形式

```
FCOPYSVR.EXE /S:SourceName [/SU:SourceUserName]
        /D:DestName [/DU:DestUserName]
        [/LG:LoginGroup] [/A | /B]
        [/RC:nn /RI:nnnn]
```

機能説明

サーバ間でのファイルの転送を行うコマンドです。ローカルファイル、Windowsのディレクトリ共有を利用したファイルまたはFTPサーバ上のファイルを転送することができます。Systemwalker Operation Managerがインストールされていない転送先のサーバにも転送できます。

パラメタ

/S:SourceName

転送元のファイルのパス名をファイル名も含めて指定します。ファイル名には、ワイルドカードも指定できます。

SourceNameには、ファイルのパス名を次のいずれかの方法で、URL形式で指定してください。パス名部分は255文字以内で指定し、空白文字を含む場合はダブルクォーテーション(")で囲んでください。

ローカルドライブパス指定:

ドライブ名を含んだパスを指定します。

“file://”文字列に続いて、任意のファイルのパス名を指定します。

例 file://C:¥DATABASE¥DATABASE.DB

UNC形式指定(Windows共有方式):

“¥¥コンピュータ名¥共有名”の形式でパスを指定します。コンピュータ名はWindowsのコンピュータ名を半角英数字で15文字以内で指定してください。共有名は80バイト以内で指定してください。

“file://”文字列に続いて、任意のファイルのパス名を指定します。

例 file://¥¥foo¥SHARE¥DATABASE¥DATABASE.DB

FTPサーバ上のパス指定(FTP方式):

FTPサーバに対するパス名を指定します。

パス名は、フルパスまたはFTPでログイン時のカレントディレクトリからの相対パスで指定することができます。“ftp://”文字列に続いて、“サーバ名/任意のファイルのパス名”の形式で指定します。

例)フルパスで指定する場合：

ftp://foo.bar.com//Usr/DataBase/DataBase.DB

相対パスで指定する場合：

ftp://foo.bar.com/DataBase/DataBase.DB

/SU:*SourceUserName*

SourceNameで指定したファイルにアクセスするためのユーザ名を指定します。

SourceUserNameは、パスワード管理簿にあらかじめ設定したユーザ名を指定します。SourceUserNameに空白文字を含む場合は、ダブルクォーテーション(")で囲んでください。

SourceNameの指定がローカルドライブパスの場合は、本指定は無視されます。UNC形式指定またはFTPサーバ上のパスで指定した場合は、必ず本指定を行ってください。

UNC形式指定の場合は、SourceUserNameは“ドメイン名¥ユーザ名”または“サーバ名¥ユーザ名”のいずれかの形式で指定してください。この場合、ドメイン名またはサーバ名は半角英数字15文字以内で、ユーザ名は半角英数字20文字以内で指定してください。

なお、FTPサーバ上のパス指定の場合は、“ドメイン名¥ユーザ名”を指定できません。パスワード管理簿に“ユーザ名@ドメイン名”で登録したユーザまたは、ローカルアカウントを指定してください。

FTPサーバ上のパス指定の場合は、SourceUserNameが半角英数字20文字以内になるように指定してください。

/D:*DestName*

転送先のファイルのパス名をファイル名も含めて指定します。ファイル名には、ワイルドカードで指定もできます。

ファイルのパス名の指定方法は/S:SourceNameと同様です。/S:SourceNameを参照してください。

すでに存在するファイル名を指定した場合は、上書きされます。また、/Sに指定するファイルのパス名(SourceName)と同じパス名を指定しないでください。

/DU:*DestUserName*

DestNameで指定したファイルにアクセスするためのユーザ名を指定します。

DestUserNameは、パスワード管理簿にあらかじめ設定したユーザ名を指定します。DestUserNameに空白文字を含む場合は、ダブルクォーテーション(")で囲んでください。

DestNameの指定がローカルドライブパスの場合は、本指定は無視されます。UNC形式指定またはFTPサーバ上のパスの場合は、必ず本指定を行ってください。

UNC形式指定の場合は、DestUserNameは“ドメイン名¥ユーザ名”または“サーバ名¥ユーザ名”のいずれかの形式で指定する必要があります。この場合、ドメイン名またはサーバ名は半角英数字15文字以内で、ユーザ名は半角英数字20文字以内で指定してください。

なお、FTPサーバ上のパス指定の場合は、“ドメイン名¥ユーザ名”を指定できません。パスワード管理簿に“ユーザ名@ドメイン名”で登録したユーザまたは、ローカルアカウントを指定してください。

FTPサーバ上のパス指定の場合は、DestUserNameが半角英数字20文字以内になるように指定してください。

/LG:*LoginGroup*

パスワード管理簿のログイン定義名を指定します。LoginGroupに、256文字以内でログイン定義名を指定し、空白文字を含む場合はダブルクォーテーション(")で囲んでください。

/SU、/DUの指定をした場合は必ず本指定を行ってください。

/A

ASCIIテキストモードでファイルを転送します。

本指定は、FTPサーバ上のファイルを転送する場合のみ有効となります。

/B

バイナリモードでファイルを転送します。

本指定は、FTPサーバ上のファイルを転送する場合のみ有効となります。

/A、/Bともに指定されていない場合は、バイナリモードでファイルを転送します。

/RC:*nn*

ファイルの転送でエラーが発生した場合のリトライ回数を指定します。

nnに0～99の整数を指定します。それ以外の値を設定すると、エラーとなります。省略した場合はリトライを行いません。

転送元ファイルまたは転送先ファイルにワイルドカードを指定した場合、失敗したファイルのみリトライが行われます。

/RI:*nnnn*

エラーが発生した場合のリトライ間隔を指定します。

nnnnに0～9999の整数を秒単位で指定します。それ以外の値を設定した場合は、エラーとなります。省略値は0です。

本パラメタは、/RCパラメタで1以上を指定した場合に有効となります。

復帰値

復帰値	意 味
0	正常終了
0以外	異常終了

コマンド格納場所

Systemwalker Operation Managerインストールディレクトリ¥MpWalker.JM¥bin

記述例

サーバ上のローカルファイル“c:¥database¥master¥database.db”をローカルファイル“d:¥database¥database.db”へ転送します。転送に失敗した場合は、60秒間隔で最大3回リトライします。

```
FCOPYSVR.EXE /S:file://c:¥database¥master¥database.db
        /D:file://d:¥database¥database.db /RC:3 /RI:60
```

処理結果

コマンドの処理が終了すると、終了時の情報を標準出力に出力します。

出力される内容を以下に示します。

```
[処理開始日時]　YYYY-MM-DD HH:MM:SS
[サーバホスト／ユーザ名]　サーバホスト名／ユーザ名 (*)
[リトライ回数]　リトライ回数 (*)
[発生したエラー]　エラーメッセージ (*)
[ファイル転送]　転送元ファイル名 転送先ファイル名 (**)
[リトライ回数]　リトライ回数 (**)
[発生したエラー]　エラーメッセージ (**)
[処理終了日時]　YYYY-MM-DD HH:MM:SS
```

(*):

/SU、/DUを指定した場合に出力される情報です。

(**):

ワイルドカードが指定された場合、それぞれのファイルについての情報も出力されます。

[発生したエラー]には、正常終了の場合は“なし”が出力されます。強制終了やエラーになった場合はメッセージが出力されます。メッセージ内容については“Systemwalker Operation Manager メッセージ集”を参照してください。

ファイルの転送を行う前にエラーが発生した場合は、以下の形式でメッセージが出力されます。

```
[処理開始日時]　YYYY-MM-DD HH:MM:SS
[コマンドエラー]　エラーメッセージ
[処理終了日時]　YYYY-MM-DD HH:MM:SS
```

メッセージ内容については“Systemwalker Operation Manager メッセージ集”を参照してください。

使用上の注意

- 本コマンドを使用する場合は、サーバ上で業務連携サービスを動作させておいてください。
- 事前にパスワード管理コマンドを使って、パスワード管理簿の設定を行ってください。詳細は“9.3.1 LGMANAGE.EXE パスワード管理コマンド”を参照してください。
- UNC形式指定で接続する場合は、“Windows共有方式”を利用しています。そのため、業務連携コマンドまたはほかのアプリケーション(エクスプローラーなど)から、同一のサーバに対して同時に接続するときは、同じ資格情報(ユーザ名/パスワード)を使用してください。

  それぞれで異なる資格情報を使用して、コマンドを実行すると、接続エラーとなります。
- FTPサーバ上のパス指定でファイルを転送する場合、転送元および転送先の両方をFTPサーバ上のパス指定で転送することはできません。転送元と転送先に指定できる組み合わせは以下のとおりです。いずれかの組み合わせで指定してください。

	転送元ファイル	転送先ファイル
1	ローカルパス形式	ローカルパス形式
2	ローカルパス形式	UNC形式
3	ローカルパス形式	FTPサーバパス形式
4	UNC形式	ローカルパス形式
5	UNC形式	UNC形式
6	UNC形式	FTPサーバパス形式
7	FTPサーバパス形式	ローカルパス形式
8	FTPサーバパス形式	UNC形式

- FTPサーバ上のパス指定でファイル転送ができるFTPサーバおよびファイルは、“DIR”コマンドの結果が以下の形式(パーミッション、リンク数、所有者、グループ、サイズ、日付、時刻、ファイル名の順に英数字表記)で表示されるサーバおよびファイルに限ります。日本語を含むファイル名については、“DIR”コマンドの結果

でファイル名が正しく表示されるファイルを扱うことができます。ファイル名が文字化けして表示されるファイルは扱うことができません。

```
-rw-rw-r--  1 xxxxx  xxxxxx     605877 Dec 19 18:44 xxxxx.xxx
-rw-rw-r--  1 xxxxx  xxxxxx     560661 Dec 19 18:59 xxxxx.xxx
drwx------  4 xxxxx  xxxxx          96 Jan 20  1998 xxxxx
-rw-rw-r--  1 xxxxx  xxxxx        6042 Jan 25 13:38 xxxxxxxxxx
-rw-rw-r--  1 xxxxx  xxxxx        5433 Feb 3  17:11 xxxxx.xxx
-rw-rw-r--  1 xxxxx  xxxxx        2155 Feb 3  17:35 xxxxx.xxx
-rw-rw-r--  1 xxxxx  xxxxx        3090 Feb 1  20:13 xxxxx.xxx
```

FTPサーバ上のパス指定でファイル転送が可能かどうかの確認は、次の方法で行ってください。

Windowsの場合:

－ コマンドプロンプト上でftpコマンドを実行します。

UNIXの場合:

－ サーバコンソールやtelnet接続したコンソール上でftpコマンドを実行します。

以下に、Windows上で確認する場合の手順を示します。

```
C:¥ ftp XXXXX                          ←ftpコマンド(XXXXX:ftpサーバ名)
Connected to XXXXX.
220 XXXXX FTP server (UNIX(r) System V Release 4.0) ready.
Name (XXXXX:YYYYY): YYYYY              ←ftpサーバのユーザ名指定（YYYYY:ユーザ名）
331 Password required for YYYYY.
Password:                              ←パスワード入力
230 User YYYYY logged in.
ftp> dir                               ←dirコマンド入力
200 PORT command successful.
          ↓
dirの結果が出力されます。
```

日付・時刻表示が日本語の場合、OSやFTPサーバのマニュアルを参照し英語表示となるように設定してください。

出荷状態の日本語Solaris2.5以降に付属のFTPサーバでは、ディレクトリの日付・時刻が日本語で表示されます。上記の形式で表示するためには、/etc/default/initファイルに以下の記述を追加してください。

```
LC_TIME=C
```

- コマンドの強制終了は、1つのファイルの転送が完了した時点で行われます。このため、大きなファイルを転送中に強制終了を行うと、コマンドの終了を待たされる場合があります。
- 本コマンドの/sパラメタに指定したファイル名に、ワイルドカードを指定した場合は、次の注意が必要です。

  一度に処理するファイル数は、同時に実行するほかのサーバ業務連携コマンドで処理するファイル数を含めて、合計500ファイル以下としてください。
- パラメタに指定するファイル名にワイルドカードを使用する場合は、ファイル名全体をワイルドカード(アスタリスクのみ)とすることをお勧めします。ファイル名の一部をワイルドカードにする場合は、転送したファイルが想定するファイル名となるかどうかを確認のうえ利用してください。
- 1ファイルあたりのサイズは、2GB以下としてください。

9.1.3 FCTRLSVR.EXE ファイル操作コマンド

記述形式

```
FCTRLSVR.EXE {/DELETE | /RENAME | /MKDIR | /RMDIR} /
F:FilePathName
        [/FN:FileName] [/UN:UserName]
        [/LG:LoginGroup] [/RC:nn /RI:nnnn]
```

機能説明

サーバ上のファイルまたはディレクトリを操作するコマンドです。ファイル操作では、ファイルの削除およびファイル名を変更することができます。ディレクトリ操作では、ディレクトリを作成および削除することができます。本コマンドで操作可能なファイルは、ローカルファイル、Windowsのディレクトリ共有を利用したファイルおよびFTPサーバ上のファイルです。

Systemwalker Operation Managerがインストールされていない操作先サーバを操作することもできます。

パラメタ

/DELETE

ファイルを削除します。

/RENAME

ファイル名を変更します。

/MKDIR

ディレクトリを作成します。

/RMDIR

ディレクトリを削除します。

/F:*FilePathName*

対象のファイルまたはディレクトリのパス名を指定します。/DELETEまたは/RENAMEを指定した場合は、ファイル名には、ワイルドカードも指定できます。

FilePathNameは、URL形式のファイルのパス名を次のいずれかの方法で指定してください。パス名は255文字以内で指定し、空白文字を含む場合はダブルクォーテーション(")で囲んでください。

ローカルドライブパス指定:

ドライブ名を含んだパス名を指定します。

“file://”文字列に続いて、任意のファイルのパス名を指定します。

例 file://C:¥DATABASE¥DATABASE.DB

UNC形式指定(Windows共有方式):

“¥¥コンピュータ名¥共有名”の形式でパス名を指定します。コンピュータ名はWindowsのコンピュータ名を半角英数字15文字以内で指定してください。共有名は80バイト以内で指定してください。

“file://”文字列に続いて、任意のファイルのパス名を指定します。

例 file://¥¥foo¥SHARE¥DATABASE¥DATABASE.DB

FTPサーバ上のパス指定(FTP方式):

FTPサーバに対するパス名を指定します。

パス名は、フルパスまたはFTPでログイン時のカレントディレクトリからの相対パスで指定することができます。“ftp://”文字列に続いて、“サーバ名/任意のファイルのパス名”の形式で指定します。

例)フルパスで指定する場合：

ftp://foo.bar.com//Usr/DataBase/DataBase.DB

相対パスで指定する場合：

ftp://foo.bar.com/DataBase/DataBase.DB

/FN:*FileName*

変更先のファイル名を指定します。ワイルドカードも指定できます。FileNameに、255文字以内でファイル名を指定し、空白文字を含む場合はダブルクォーテーション(")で囲んでください。

本パラメタは、/RENAMEを指定した場合のみ有効です。

/UN:*UserName*

FilePathNameで指定したファイルまたはディレクトリにアクセスするユーザ名を指定します。

UserNameは、パスワード管理簿にあらかじめ設定したユーザ名を指定します。UserNameに空白文字を含む場合は、ダブルクォーテーション(")で囲んでください。

FilePathNameがローカルドライブパスで指定した場合は本指定は無視されます。

FTPサーバ上のパスまたはUNC形式で指定した場合は、必ず本指定を行ってください。

UNC形式指定の場合は、UserNameは“ドメイン名¥ユーザ名”または“サーバ名¥ユーザ名”のいずれかの形式で指定してください。この場合、ドメイン名またはサーバ名は半角英数字15文字以内で、ユーザ名は半角英数字20文字以内で指定してください。

なお、FTPサーバ上のパス指定の場合は、“ドメイン名¥ユーザ名”を指定できません。パスワード管理簿に“ユーザ名@ドメイン名”で登録したユーザまたは、ローカルアカウントを指定してください。

FTPサーバ上のパス指定の場合は、UserNameが半角英数字20文字以内になるように指定してください。

/LG:*LoginGroup*

パスワード管理簿のログイン定義名を指定します。LoginGroupは、256文字以内でログイン定義名を指定し、空白文字を含む場合はダブルクォーテーション(")で囲んでください。

/UNの指定をした場合は必ず本指定を行ってください。

/RC:*nn*

ファイルの操作でエラーが発生した場合のリトライ回数を指定します。

nnは0～99の整数を指定してください。それ以外の値を設定した場合は、エラーになります。省略した場合はリトライを行いません。

/RI:*nnnn*

エラーが発生した場合のリトライ間隔を指定します。

nnnnに0～9999の整数を秒単位で指定してください。それ以外の値を設定した場合は、エラーになります。省略値は0です。

本パラメタは、/RCパラメタで1以上を指定した場合に有効となります。

復帰値

復帰値	意 味
0	正常終了
0以外	異常終了

コマンド格納場所

Systemwalker Operation Managerインストールディレクトリ¥MpWalker.JM¥bin

記述例

サーバ上のローカルファイル“c:¥database¥master¥database.db”をファイル名“database_backup.db”へ変更します。ファイル名変更に失敗した場合は、60秒間隔で最大3回リトライします。

```
FCTRLSVR.EXE /RENAME /F:file://c:¥database¥master¥database.db
        /FN:database_backup.db /RC:3 /RI:60
```

処理結果

コマンドの処理が終了すると、終了時の情報を標準出力に出力します。

出力される内容を以下に示します。

```
[処理開始日時]　YYYY-MM-DD HH:MM:SS
[サーバホスト／ユーザ名]　サーバホスト名／ユーザ名 (*)
[リトライ回数]　リトライ回数 (*)
[発生したエラー]　エラーメッセージ (*)
[ファイル操作種別]　操作ファイル／ディレクトリ名 変更後ファイル名 (**)
(***)
[リトライ回数]　リトライ回数 (**)
[発生したエラー]　エラーメッセージ (**)
[処理終了日時]　YYYY-MM-DD HH:MM:SS
```

(*):

/UNを指定した場合に出力される情報です。

(**):

ワイルドカードが指定された場合、それぞれのファイルについての情報も出力されます。

(***):

変更後ファイル名は、/RENAMEが指定された場合のみ出力されます。

[発生したエラー]には、正常終了の場合は“なし”が出力されます。強制終了やエラーの場合はメッセージが出力されます。メッセージ内容については“Systemwalker Operation Manager メッセージ集”を参照してください。

“ファイル操作種別”は、“ファイル削除”、“ファイル名変更”、“ディレクトリ作成”、“ディレクトリ削除”のいずれかが出力されます。

ファイル操作を行う前にエラーが発生した場合は、以下の形式でメッセージが出力されます。

```
[処理開始日時]　YYYY-MM-DD HH:MM:SS
[コマンドエラー]　エラーメッセージ
[処理終了日時]　YYYY-MM-DD HH:MM:SS
```

メッセージ内容については“Systemwalker Operation Manager メッセージ集”を参照してください。

使用上の注意

- 本コマンドを使用する場合は、サーバ上で業務連携サービスを動作させておいてください。
- 事前にパスワード管理コマンドを使って、パスワード管理簿の設定を行ってください。詳細は“9.3.1 LGMANAGE.EXE パスワード管理コマンド”を参照してください。
- UNC形式指定で接続する場合は、“Windows共有方式”を利用しています。そのため、業務連携コマンドまたはほかのアプリケーション(エクスプローラーなど)から、同一のサーバに対して同時に接続するときは、同じ資格情報(ユーザ名/パスワード)を使用してください。

  それぞれで異なる資格情報を使用して、コマンドを実行すると、接続エラーとなります。
- FTPサーバ上のパス指定でファイルを操作できるFTPサーバおよびファイルは、“DIR”コマンドの結果が以下の形式(パーミッション、リンク数、所有者、グループ、サイズ、日付、時刻、ファイル名の順に英数字表記)で表示されるサーバおよびファイルに限ります。日本語を含むファイル名については、“DIR”コマンドの結果

でファイル名が正しく表示されるファイルを扱うことができます。ファイル名が文字化けして表示されるファイルは扱うことができません。

```
-rw-rw-r--  1 xxxxx  xxxxxx     605877 Dec 19 18:44 xxxxx.xxx
-rw-rw-r--  1 xxxxx  xxxxxx     560661 Dec 19 18:59 xxxxx.xxx
drwx------  4 xxxxx  xxxxx          96 Jan 20  1998 xxxxx
-rw-rw-r--  1 xxxxx  xxxxx        6042 Jan 25 13:38 xxxxxxxxxxx
-rw-rw-r--  1 xxxxx  xxxxx        5433 Feb 3  17:11 xxxxx.xxx
-rw-rw-r--  1 xxxxx  xxxxx        2155 Feb 3  17:35 xxxxx.xxx
-rw-rw-r--  1 xxxxx  xxxxx        3090 Feb 1  20:13 xxxxx.xxx
```

FTPサーバ上のパス指定が可能であるかどうかの確認方法は、“9.1.2 FCOPYSVR.EXE ファイル転送コマンド”を参照してください。

日付・時刻表示が日本語の場合、OSやFTPサーバのマニュアルを参照し英語表示となるように設定してください。

なお、出荷状態の日本語Solaris2.5以降に付属のFTPサーバでは、ディレクトリの日付・時刻が日本語で表示されます。上記の形式で表示するためには、/etc/default/initファイルに以下の記述を追加してください。

```
LC_TIME=C
```

- 本コマンドの/fパラメタに指定したファイル名に、ワイルドカードを指定した場合は、次の注意が必要です。

  一度に処理するファイル数は、同時に実行するほかのサーバ業務連携コマンドで処理するファイル数を含めて、合計500ファイル以下としてください。

- パラメタに指定するファイル名にワイルドカードを使用する場合は、ファイル名全体をワイルドカード(アスタリスクのみ)とすることをお勧めします。ファイル名の一部をワイルドカードにする場合は、操作したファイルが想定するファイル名となるかどうかを確認のうえ利用してください。

- 1ファイルあたりのサイズは、2GB以下としてください。

9.1.4 FPRSSVR.EXE ファイル圧縮/伸長コマンド

記述形式

```
FPRSSVR.EXE {/P | /E} /S:SourceName /D:DestName
        [/RC:nn /RI:nnnn]
```

機能説明

サーバ上のローカルファイルの圧縮および伸長を行います。

転送元でファイルを圧縮して、転送先でファイルを伸長するといった使い方ができ、ファイルの転送時間が短縮され、ネットワーク上の負荷を軽減できます。

パラメタ

/P

ファイルを圧縮します。

/E

ファイルを伸長します。

/S:SourceName

圧縮/伸長元ファイルのパス名をローカルパスで指定します。ローカルパスは、ドライブ名を含んだ絶対パスで、"C:¥DATABASE¥DATABASE.DB"のように指定します。FTPサーバ上のパス指定はできません。圧縮元ファイルの場合は、ファイル名には、ワイルドカードも指定できます。

SourceNameは、255文字以内で指定してください。SourceNameに空白文字を含む場合は、ダブルクォーテーション(")で囲んでください。

/D:*DestName*

圧縮/伸長先のファイルのパス名をローカルパスで指定します。FTPサーバ上のパス指定はできません。伸長先ファイルの場合は、ファイル名には、ワイルドカードで指定もできます。

DestNameは、255文字以内で指定してください。DestNameに空白文字を含む場合は、ダブルクォーテーション(")で囲んでください。

すでに存在するファイル名を指定した場合は、上書きされます。また、/Sに指定するファイルのパス名(SourceName)と同じパス名を指定しないでください。

/RC:*nn*

ファイルの圧縮/伸長でエラーが発生した場合のリトライ回数を指定します。nnは0～99の整数で指定してください。それ以外の値を設定すると、エラーになります。省略した場合は、リトライを行いません。

圧縮元ファイル、または伸長先ファイルにワイルドカードが指定された場合、失敗したファイルのみリトライを行います。

/RI:*nnnn*

エラーが発生した場合のリトライ間隔を指定します。nnnnには0～9999の整数を秒単位で指定してください。それ以外の値を設定すると、エラーになります。省略値は0です。

本パラメタは、/RCパラメタで1以上を指定した場合に有効となります。

復帰値

復帰値	意 味
0	正常終了
0以外	異常終了

コマンド格納場所

Systemwalker Operation Managerインストールディレクトリ¥MpWalker.JM¥bin

記述例

サーバ上のローカルファイル“d:¥database¥database.prs”を“c:¥database¥copy¥*.*”に伸長します。伸長に失敗した場合は、60秒間隔で最大3回リトライします。

```
FPRSSVR.EXE /E /S:d:¥database¥database.prs /D:c:¥database¥copy¥*.*
/RC:3 /RI:60
```

処理結果

コマンドの処理が終了すると、終了時の情報を標準出力に出力します。

出力される内容を以下に示します。

```
[処理開始日時]  YYYY-MM-DD HH:MM:SS
[ファイル圧縮／伸長]  圧縮／伸長元ファイル名 圧縮／伸長先ファイル名 (*)
[リトライ回数]  リトライ回数 (*)
[発生したエラー]  エラーメッセージ (*)
[処理終了日時]  YYYY-MM-DD HH:MM:SS
```

(*):

ワイルドカードが指定された場合、それぞれのファイルについての情報も出力されます。

[発生したエラー]には、正常終了の場合は“なし”が出力されます。強制終了やエラーの場合はメッセージが出力されます。メッセージ内容については“Systemwalker Operation Manager メッセージ集”を参照してください。

圧縮/伸長処理を行う前にエラーが発生した場合は、以下の形式でメッセージが出力されます。

```
[処理開始日時]　YYYY-MM-DD HH:MM:SS
[コマンドエラー]　エラーメッセージ
[処理終了日時]　YYYY-MM-DD HH:MM:SS
```

メッセージ内容については“Systemwalker Operation Manager メッセージ集”を参照してください。

使用上の注意

- 本コマンドを使用する場合は、サーバ上で業務連携サービスを動作させておいてください。
- UNIX版 Systemwalker Operation Managerが動作しているサーバ上でfprssvrコマンドの/wオプションをつけて圧縮されたファイルを、Windows上で本コマンドを使って伸長した場合、次のように展開されます。
  - 圧縮元で書込みが誰にも許可されていない場合は、Windowsでは読み取り専用属性になります。
  - 圧縮元で書込みがowner/group/otherのいずれかに許されていた場合は、Windowsでは読み書き可能な属性になります。
- 本コマンドで圧縮したファイルをUNIX版 Systemwalker Operation Managerが動作しているサーバ上で伸長した場合、展開後のファイル情報は以下の表のようになります。

圧縮前のWindows上でのファイル属性	UNIX上で展開後のファイルの情報
読み取り専用	・所有者／グループ情報は、コマンドを実行したユーザの情報となる。 ・モードはすべてのユーザが読込み可能な0444となる。(注1)
読み取り専用以外	・所有者／グループ情報は、コマンドを実行したユーザの情報となる。 ・モードはすべてのユーザが読み書き可能な0666となる。(注2)

注1）
　すべてのユーザから読込み可能な状態で、ls -lコマンドで-r--r--r--と表示されるファイルのパーミッションをさします。

注2）
　すべてのユーザから読み書き可能な状態で、ls -lコマンドで-rw-rw-rw-と表示されるファイルのパーミッションをさします。

- 圧縮するユーザに、読み取り権限が設定されていない場合は圧縮することはできません。ただし、システム管理者(Administratorsグループに属するユーザ)は圧縮可能です。
- コマンドの強制終了は、1つのファイル圧縮/伸長が完了した時点で行われます。このため、大きなファイルを圧縮中に強制終了を行うと、コマンドの終了を待たされる場合があります。
- 本コマンドの/sパラメタに指定したファイル名に、ワイルドカードを指定した場合は、次の注意が必要です。

  一度に処理するファイル数は、同時に実行するほかのサーバ業務連携コマンドで処理するファイル数を含めて、合計500ファイル以下としてください。
- パラメタに指定するファイル名にワイルドカードを使用する場合は、ファイル名全体をワイルドカード(アスタリスクのみ)とすることをお勧めします。ファイル名の一部をワイルドカードにする場合は、圧縮/伸長したファイルが想定するファイル名となるかどうかを確認のうえ利用してください。

・ 本コマンドの実行時には、サーバ上の以下の場所に、一時ファイルが作成されます。

ー 環境変数TEMPに指定されているディレクトリ

TEMP領域が不足した場合、圧縮・伸長することはできません。一時ファイルのサイズは、一度に圧縮・伸長するファイルサイズの合計となります。

・ 1ファイルあたりのサイズは、2GB以下としてください。

9.2 クライアント業務連携コマンド

クライアント業務連携コマンドは、クライアントとの業務連携を行うためのコマンドです。

参考

クライアント業務連携コマンドは、バッチファイルを作成して登録することも可能です。

9.2.1 PONCLI.EXE 電源投入コマンド

記述形式

PONCLI.EXE *HostName*
[/IP:*IPAddress* /MAC:*MACAddress* /SNM:*SubnetMask*]
[/WT:*WaitTime*] [/RC:*nn* /RI: *nnnn*]

機能説明

Wakeup on LANに対応したクライアントの電源を投入し、離れた場所にあるクライアントを自動的に起動することが可能になります。

また、TCP/IP環境であれば他のサブネット上のクライアントの電源を投入することもできます。

本コマンドは、IPv6通信環境ではサポートされていません。IPv4通信環境、またはデュアルスタックが有効な環境で利用してください。

パラメタ

HostName

電源を投入するクライアントのホスト名を指定します。

HostNameに、半角英数字15文字以内でホスト名を指定します。空白文字を含む場合はダブルクォーテーション(")で囲んでください。

/IP:IPAddress

電源を投入するクライアントのIPアドレスを指定します。IPアドレスは、10進数の文字列で“133.161.222.110”のように指定します。

省略した場合は、ホスト情報定義ファイルから対象クライアントのIPアドレスを読み出します。

/MAC:MACAddress

電源を投入するクライアントのMACアドレスを指定します。MACアドレスは、16進数の文字列で“00-00-0E-9D-2C-38”のように指定します。省略した場合は、対象クライアントのMACアドレスをホスト情報定義ファイルから読み出します。

/SNM:SubnetMask

電源を投入するクライアントのサブネットマスクを指定します。サブネットマスクは、10進数の文字列で“255.255.255.0”のように指定します。

省略した場合は、対象クライアントのサブネットマスクをホスト情報定義ファイルから読み出します。

/WT:*WaitTime*

電源投入確認までの待ち合わせ時間を指定します。WaitTimeに1～999までの整数を秒単位で指定してください。

本パラメタを省略した場合は、電源投入の待ち合わせを行いません。

本パラメタを指定すると、指定した待ち合わせ時間の経過後に電源投入確認が行われます。そのため、待ち合わせ時間より早く電源投入が完了していても、コマンドは指定された待ち合わせ時間を待ち合わせます。

待ち合わせ時間の経過後、電源投入が確認できた場合は正常終了、できなかった場合は異常終了します。

/RC:*nn*

クライアントの電源投入でエラーが発生した場合のリトライ回数を指定します。

nnには0～99の整数を指定してください。それ以外の値を設定した場合は、エラーになります。省略した場合は、リトライを行いません。

/RI:*nnnn*

エラーが発生した場合のリトライ間隔を指定します。本パラメタは、/RC:nnに1以上を指定した場合に有効となります。

nnnnには0～9999の整数を秒単位で指定してください。それ以外の値を設定した場合は、エラーになります。省略値は0です。

復帰値

復帰値	意 味
0	正常終了
0以外	異常終了

コマンド格納場所

Systemwalker Operation Managerインストールディレクトリ¥MpWalker.JM¥bin

記述例

クライアント“foo”に対して電源の投入を行います。

以下の情報を使用します。

IPアドレス:

133.161.222.110

MACアドレス:

00-00-0E-9D-2C-38

サブネットマスク:

255.255.255.0

```
PONCLI.EXE foo /IP:133.161.222.110 /MAC:00-00-0E-9D-2C-38 /SNM:
255.255.255.0
```

処理結果

コマンドの処理が終了すると、終了時の情報を標準出力に出力します。

出力される内容を以下に示します。

```
[処理開始日時] YYYY-MM-DD HH:MM:SS
[ホスト名] HostName
[MACアドレス] MACアドレス
[IPアドレス] IPアドレス
[サブネットマスク] サブネットマスク
[リトライ回数] リトライ回数
[発生したエラー] エラーメッセージ
[処理終了日時] YYYY-MM-DD HH:MM:SS
```

[発生したエラー]には、正常終了の場合は“なし”が出力されます。強制終了やエラーの場合はメッセージが出力されます。メッセージ内容については“Systemwalker Operation Manager メッセージ集”を参照してください。

電源投入を行う前にエラーが発生した場合は、以下の形式でメッセージが出力されます。

```
[処理開始日時] YYYY-MM-DD HH:MM:SS
[コマンドエラー] エラーメッセージ
[処理終了日時] YYYY-MM-DD HH:MM:SS
```

メッセージ内容については“Systemwalker Operation Manager メッセージ集”を参照してください。

使用上の注意

- ホスト情報定義ファイルの情報を使用して本コマンドを実行する場合は、あらかじめホスト情報定義ファイルを定義しておいてください。詳細は、“Systemwalker Operation Manager 導入ガイド”を参照してください。
- 本コマンドにより電源投入が可能なクライアントは、Wakeup on LANに対応した機種に限ります。他のサブネット上のクライアントに電源を投入する場合は、サーバから電源投入対象端末のネットワークに対してブロードキャストパケットが到達できるようにネットワーク機器(ルータなど)の設定が必要です。詳しい設定方法はご使用のネットワーク機器のマニュアルを参照してください。

  ブロードキャストパケットが到達するかどうかは、コマンドを実行するSystemwalker Operation Managerのサーバ側のコマンドプロンプト画面でpingコマンドなどを投入して確認してください。応答がない場合はブロードキャストパケットは到達しません。

  投入先IPアドレスが、10.90.100.100で、サブネットマスクが255.255.255.0の場合の投入コマンドは、“ping 10.90.100.255”となります。
- 本コマンドは、対象クライアントの属するネットワークに対して、投入命令であるWakeup on LANパケットが通るとすぐに電源投入成功として復帰します。電源が投入されシステムが立ち上がるまでを確実に待ち合わせたい場合は、/wtオプションを指定してください。

9.2.2 POFFCLI.EXE 電源切断コマンド

記述形式

```
POFFCLI.EXE HostName [/SD:ShutdownMode] [/EF]
        [/RC:nn /RI:nnnn]
```

機能説明

Windowsからの電源切断に対応したクライアントの電源を切断します。離れた場所にあるクライアントの電源を自動的に切断することが可能となります。

パラメタ

HostName

電源を切断するクライアントのホスト名を指定します。

HostNameに、半角英数字15文字以内でホスト名を指定します。空白文字を含む場合はダブルクォーテーション(")で囲んでください。

/SD:*ShutdownMode*

電源の切断モードを指定します。以下のいずれかを指定してください。

poff:

システムの電源を切断します。

reboot:

システムを再起動します。

down:

システムをシャットダウンします。

省略した場合は、poffが指定されたものとして、システムの電源を切断します。

/EF

電源切断の強制実行を指定します。

本指定により、起動中のアプリケーションを強制的に終了します。なお、アプリケーションが使用中のデータは保存されません。

クライアントのアクティブデスクトップは無効にしてください。アクティブデスクトップを有効にしている場合は、次回OS起動時に、アクティブデスクトップの修復機能が自動起動される場合があります。

/RC:*nn*

クライアントの電源切断でエラーが発生した場合のリトライ回数を指定します。

nnに0～99の整数を指定してください。それ以外の値を設定した場合は、エラーになります。省略した場合は、リトライを行いません。

/RI:*nnnn*

エラーが発生した場合のリトライ間隔を指定します。

nnnnに0～9999の整数を秒単位で指定してください。それ以外の値を設定した場合は、エラーになります。省略値は0です。

本パラメタは、/RC:nnに1以上が指定された場合に有効となります。

復帰値

復帰値	意 味
0	正常終了
0以外	異常終了

コマンド格納場所

Systemwalker Operation Managerインストールディレクトリ¥MpWalker.JM¥bin

記述例

クライアント“foo”に対して、電源の切断を行います。

```
POFFCLI.EXE foo /
SD:POFF
```

処理結果

コマンドの処理が終了すると、終了時の情報を標準出力に出力します。

出力される内容を以下に示します。

```
[処理開始日時] YYYY-MM-DD HH:MM:SS
[クライアントホスト名] HostName
[リトライ回数] リトライ回数
[発生したエラー] エラーメッセージ
[処理終了日時] YYYY-MM-DD HH:MM:SS
```

[発生したエラー]には、正常終了の場合は“なし”が出力されます。強制終了やエラーの場合はメッセージが出力されます。メッセージ内容については“Systemwalker Operation Manager メッセージ集”を参照してください。

電源切断を行う前にエラーが発生した場合は、以下の形式でメッセージが出力されます。

```
[処理開始日時] YYYY-MM-DD HH:MM:SS
[コマンドエラー] エラーメッセージ
[処理終了日時] YYYY-MM-DD HH:MM:SS
```

メッセージ内容については“Systemwalker Operation Manager メッセージ集”を参照してください。

使用上の注意

- 本コマンドを使用する場合は、クライアント側でクライアント業務連携を起動させておいてください。この時、電源切断可能なユーザ権限でクライアント業務連携が起動されている必要があります。
- 電源切断が可能なクライアントは、Windowsの電源切断機構に対応している機種に限ります。
- Windowsからの終了要求に対して、動作中のアプリケーションが終了しない場合は、クライアントの電源が切断できないことがあります。
- /EFパラメタが指定されている場合、Windowsからの終了要求に対して応答しないアプリケーションはWindowsから強制的に終了されます。
- /RCパラメタおよび/RIパラメタの指定が有効となるのは、Windows OSにおいて、対象クライアントのログインユーザがシャットダウン権限を持っていない場合です。

9.2.3 EXECACLI.EXE アプリケーション起動コマンド

記述形式

```
EXECACLI.EXE HostName [/AM:AppMode] [/WE:WaitEnd]
        [/RC:nn /RI:nnnn] /AE:ExecuteAppli
        [AppliParam]
```

機能説明

クライアントにインストールされている任意のアプリケーションを起動します。アプリケーションの終了を待ち合わせることができます。

パラメタ

HostName

アプリケーションを起動するクライアントのホスト名を指定します。

HostNameに、半角英数字15文字以内でホスト名を指定し、空白文字を含む場合はダブルクォーテーション(")で囲んでください。

/AM:*AppMode*

アプリケーションの起動モードを指定します。

以下のいずれかを指定してください。

NORMAL:

通常のウィンドウの大きさで起動します。

ICON:

ウィンドウが最小化された状態で起動します。

本パラメタを省略した場合は、NORMALとなり、通常のウィンドウの大きさで起動します。

/WE:*WaitEnd*

起動したアプリケーションの終了までの待ち合わせ時間を指定します。WaitEndに1～9600の整数を秒単位で指定します。それ以外の値を指定するとエラーとなります。

本パラメタを省略した場合、アプリケーションの待ち合わせを行いません。

指定した時間内にアプリケーションが0で復帰した場合、コマンドは正常終了します。

指定した時間内にアプリケーションが0以外で復帰した場合、または指定した時間内にアプリケーションが終了しなかった場合、コマンドは異常終了します。

/RC:*nn*

アプリケーションの起動でエラーが発生した場合のリトライ回数を指定します。

nnに0～99の整数を指定してください。それ以外の値を設定した場合は、エラーとなります。省略した場合はリトライを行いません。

/RI:*nnnn*

エラーが発生した場合のリトライ間隔を指定します。本パラメタは、/RC:nnで1以上を指定した場合に有効となります。

nnnnに0～9999の整数を秒単位で指定してください。それ以外の値を設定した場合は、エラーとなります。省略値は0です。

/AE:*ExecuteAppli*

起動アプリケーション名をローカルドライブパスで指定します。ExecuteAppliに、半角511文字以内のアプリケーション名を指定します。起動アプリケーション名やアプリケーションに渡すパラメタに空白文字がある場合は、それぞれをダブルクォーテーション(")で囲んでください。

AppliParam

ExecuteAppliで指定したアプリケーションに対するパラメタを指定します。パラメタを複数指定する場合は、それぞれの間に空白文字を入れてください。各パラメタに空白文字を含む場合は、その両端をダブルクォーテーション(")で囲んでください。

復帰値

復帰値	意 味
0	正常終了
0以外	異常終了(または、/WEオプション指定時に起動したアプリケーションが0以外で復帰した場合)

コマンド格納場所

Systemwalker Operation Managerインストールディレクトリ¥MpWalker.JM¥bin

記述例

クライアント“foo”に対して、アプリケーション“C:¥Program files¥app¥App.exe”を通常のウィンドウの大きさで起動し、60秒間終了を待ち合わせます。

```
EXECACLI.EXE foo /AM:NORMAL /WE:60 /AE:"C:¥Program files¥app
¥App.exe"
```

処理結果

コマンドの処理が終了すると、終了時の情報を標準出力に出力します。

出力される内容を以下に示します。

```
[処理開始日時] YYYY-MM-DD HH:MM:SS
[クライアントホスト名] HostName
[起動アプリケーション] 起動アプリケーション名
[リトライ回数] リトライ回数
[発生したエラー] エラーメッセージ
[処理終了日時] YYYY-MM-DD HH:MM:SS
```

[発生したエラー]には、正常終了の場合は“なし”が出力されます。強制終了やエラーの場合はメッセージが出力されます。メッセージ内容については“Systemwalker Operation Manager メッセージ集”を参照してください。

アプリケーション起動を行う前にエラーが発生した場合は、以下の形式でメッセージが出力されます。

```
[処理開始日時] YYYY-MM-DD HH:MM:SS
[コマンドエラー] エラーメッセージ
[処理終了日時] YYYY-MM-DD HH:MM:SS
```

メッセージ内容については“Systemwalker Operation Manager メッセージ集”を参照してください。

使用上の注意

- 本コマンドを使用する場合は、クライアント側でクライアント業務連携を起動させておいてください。
- /WEオプションを指定した場合は、起動したアプリケーションの終了を待ち合わせますが、/WEオプションを省略した場合は、起動したアプリケーションの完了を待ち合わせません。
- 本コマンドを強制終了した場合、起動したアプリケーションに対しての強制終了を行うことはできません。
- コマンドの強制終了は、アプリケーションの起動が完了した時点で行われます。このため、起動に時間がかかるアプリケーションの場合、コマンドの終了を待たされる場合があります。

9.2.4 FCOPYCLI.EXE ファイル転送コマンド

記述形式

```
FCOPYCLI.EXE HostName /S:SourceName [/SU:SourceUserName]
        /D:DestName [/DU:DestUserName]
        [/LG:LoginGroup] [/A | /B] [/RC:nn /RI:nnnn]
```

機能説明

クライアントによるファイルの転送を行います。クライアント上のファイルをサーバに転送する、またはサーバ上のファイルをクライアントへ転送するコマンドです。Windowsのディレクトリ共有を利用したファイル転送(Windows共有方式)、またはFTPによるファイル転送(FTP方式)を行います。

Systemwalker Operation Managerがインストールされていない転送先のサーバにも転送できます。

パラメタ

HostName

ファイルの転送を要求するクライアントのホスト名を指定します。HostNameにホスト名を半角英数字15文字以内で指定してください。

HostNameに空白文字を含む場合は、ダブルクォーテーション(")で囲んでください。

/S:SourceName

転送元のファイルのパス名をファイル名も含めて指定します。ファイル名には、ワイルドカードも指定できます。

SourceNameに、URL形式のファイルのパス名を次のいずれかの方法で指定してください。パス名部分は255文字以内で指定し、空白文字を含む場合は、ダブルクォーテーション(")で囲んでください。

ローカルドライブパス指定:

ドライブ名を含んだパスで指定します。

“file://”文字列に続いて、任意のファイルのパス名を指定します。

例 file://C:¥DATABASE¥DATABASE.DB

UNC形式指定(Windows共有方式):

“¥¥コンピュータ名¥共有名”の形式でパスを指定します。コンピュータ名はWindowsのコンピュータ名を半角英数字15文字以内で指定してください。共有名は80バイト以内で指定してください。

“file://”文字列に続いて、任意のファイルのパス名を指定します。

例 file://¥¥foo¥SHARE¥DATABASE¥DATABASE.DB

FTPサーバ上のパス指定(FTP方式):

FTPサーバに対するパス名を指定します。

パス名は、フルパスまたはFTPでログイン時のカレントディレクトリからの相対パスで指定することができます。“ftp://”文字列に続いて、“サーバ名/任意のファイルのパス名”の形式で指定します。

例)フルパスで指定する場合：

ftp://foo.bar.com//Usr/DataBase/DataBase.DB

相対パスで指定する場合：

ftp://foo.bar.com/DataBase/DataBase.DB

/SU:SourceUserName

SourceNameで指定したファイルにアクセスするためのユーザ名を指定します。

SourceUserNameは、パスワード管理簿にあらかじめ設定したユーザ名を指定します。SourceUserNameに、ユーザ名を半角英数字20文字以内で指定してください。空白文字を含む場合は、ダブルクォーテーション(")で囲んでください。

SourceNameの指定がローカルドライブパスまたはUNC形式指定の場合は、本指定は無視されます。FTPサーバ上のパス指定の場合は、必ず本指定を行ってください。

/D:DestName

転送先のファイルのパス名をファイル名も含めて指定します。ファイル名には、ワイルドカードで指定もできます。

DestNameに、ファイルのパス名をURL形式で指定してください。指定方法は/S:SourceNameと同様です。/S:SourceNameを参照してください。

すでに存在するファイル名を指定した場合は、上書きされます。また、/Sに指定するファイルのパス名(SourceName)と同じパス名を指定しないでください。

/DU:*DestUserName*

DestNameで指定したファイルにアクセスするためのユーザ名を指定します。

DestUserNameは、パスワード管理簿にあらかじめ設定したユーザ名を指定します。DestUserNameに、ユーザ名を半角英数字20文字以内で指定ください。空白文字を含む場合は、ダブルクォーテーション(")で囲んでください。

DestNameの指定がローカルドライブパスまたはUNC形式の場合は、本指定は無視されます。この場合、クライアント業務連携が動作しているクライアントにログインしているユーザの権限でアクセスします。FTPサーバ上のパス指定の場合は、必ず本指定を行ってください。

/LG:*LoginGroup*

パスワード管理簿のログイン定義名を指定します。LoginGroupに、ログイン定義名を256文字以内で指定してください。空白文字を含む場合は、ダブルクォーテーション(")で囲んでください。

/SU、/DUの指定をした場合は必ず本指定を行ってください。

/A

ASCIIテキストモードでファイルを転送します。本パラメタはFTP方式の場合のみ指定できます。

/B

バイナリモードでファイルを転送します。本パラメタはFTP方式の場合のみ指定できます。

/A、/Bともに指定されていない場合は、バイナリモードでファイルを転送します。

/RC:*nn*

ファイルの転送でエラーが発生した場合のリトライ回数を指定します。

nnに0～99の整数を指定してください。それ以外の値を設定した場合は、エラーになります。省略した場合は、リトライを行いません。

転送元ファイルおよび転送先ファイルにワイルドカードが指定された場合、失敗したファイルのみリトライを行います。

/RI:*nnnn*

エラーが発生した場合のリトライ間隔を指定します。本パラメタは、/RC:nnで1以上を指定した場合に有効となります。

nnnnに0～9999の整数を秒単位で指定してください。それ以外の値を設定した場合は、エラーになります。省略値は0です。

復帰値

復帰値	意 味
0	正常終了
0以外	異常終了

コマンド格納場所

Systemwalker Operation Managerインストールディレクトリ¥MpWalker.JM¥bin

記述例

クライアント“foo”に対して、クライアント上のローカルファイル“c:¥database¥master¥database.db”をローカルファイル“d:¥database¥database.db”へ転送します。転送に失敗した場合は、60秒間隔で最大3回リトライします。

```
FCOPYCLI.EXE foo /S:file://c:¥database¥master¥database.db
        /D:file://d:¥database¥database.db /RC:3 /RI:60
```

処理結果

コマンドの処理が終了すると、終了時の情報を標準出力に出力します。

出力される内容を以下に示します。

```
[処理開始日時]　YYYY-MM-DD HH:MM:SS
[クライアントホスト名]　HostName
[サーバホスト／ユーザ名]　サーバホスト名／ユーザ名 (*)
[リトライ回数]　リトライ回数 (*)
[発生したエラー]　エラーメッセージ (*)
[ファイル転送]　転送元ファイル名 転送先ファイル名 (**)
[リトライ回数]　リトライ回数 (**)
[発生したエラー]　エラーメッセージ (**)
[処理終了日時]　YYYY-MM-DD HH:MM:SS
```

(*):

/SU、/DUを指定した場合に出力される情報です。

(**):

ワイルドカードが指定された場合、それぞれのファイルについての情報も出力されます。

[発生したエラー]には、正常終了の場合は“なし”が出力されます。強制終了やエラーの場合はメッセージが出力されます。メッセージ内容については“Systemwalker Operation Manager メッセージ集”を参照してください。

ファイルの転送を行う前にエラーが発生した場合は、以下の形式でメッセージが出力されます。

```
[処理開始日時]　YYYY-MM-DD HH:MM:SS
[コマンドエラー]　エラーメッセージ
[処理終了日時]　YYYY-MM-DD HH:MM:SS
```

メッセージ内容については“Systemwalker Operation Manager メッセージ集”を参照してください。

使用上の注意

- 本コマンドを使用する場合は、クライアント側でクライアント業務連携を動作させておいてください。
- 事前にパスワード管理コマンドを使って、パスワード管理簿の設定を行ってください。詳細は“9.3.1 LGMANAGE.EXE パスワード管理コマンド”を参照してください。
- FTPサーバ上のパス指定でファイルを転送する場合、転送元および転送先の両方をFTPサーバ上のパス指定で転送することはできません。転送元と転送先に指定できる組み合わせは以下のとおりです。いずれかの組み合わせで指定してください。

	転送元ファイル	転送先ファイル
1	ローカルパス形式	ローカルパス形式
2	ローカルパス形式	UNC形式
3	ローカルパス形式	FTPサーバパス形式
4	UNC形式	ローカルパス形式
5	UNC形式	UNC形式
6	UNC形式	FTPサーバパス形式
7	FTPサーバパス形式	ローカルパス形式
8	FTPサーバパス形式	UNC形式

- FTPサーバ上のパス指定でファイルを操作できるFTPサーバおよびファイルは、“DIR”コマンドの結果が以下の形式(パーミッション、リンク数、所有者、グループ、サイズ、日付、時刻、ファイル名の順に英数字表記)で表示されるサーバおよびファイルに限ります。日本語を含むファイル名については、“DIR”コマンドの結果でファイル名が正しく表示されるファイルを扱うことができます。ファイル名が文字化けして表示されるファイルは扱うことができません。

```
-rw-rw-r--  1 xxxxx  xxxxxx     605877 Dec 19 18:44 xxxxx.xxx
-rw-rw-r--  1 xxxxx  xxxxxx     560661 Dec 19 18:59 xxxxx.xxx
drwx------  4 xxxxx  xxxxx          96 Jan 20  1998 xxxxx
-rw-rw-r--  1 xxxxx  xxxxx        6042 Jan 25 13:38 xxxxxxxxxxx
-rw-rw-r--  1 xxxxx  xxxxx        5433 Feb 3  17:11 xxxxx.xxx
-rw-rw-r--  1 xxxxx  xxxxx        2155 Feb 3  17:35 xxxxx.xxx
-rw-rw-r--  1 xxxxx  xxxxx        3090 Feb 1  20:13 xxxxx.xxx
```

  FTPサーバ上のパス指定が可能であるかどうかの確認方法は、“9.1.2 FCOPYSVR.EXE ファイル転送コマンド”を参照してください。

  出荷状態の日本語Solaris 2.5以降に付属のFTPサーバでは、ディレクトリの日付・時刻が日本語で表示されます。上記の形式で表示するためには、/etc/default/initファイルに以下の記述を追加してください。

```
LC_TIME=C
```

- コマンドの強制終了は、1つのファイルの転送が完了した時点で行われます。このため、大きなファイルを転送中に強制終了を行うと、コマンドの終了を待たされる場合があります。
- 本コマンドの/sパラメタに指定したファイル名に、ワイルドカードを指定した場合は、次の注意が必要です。

  同じクライアントに対して、同時に実行するクライアント業務連携コマンドで処理するファイル数を含めて、合計500ファイル以下となるようにしてください。
- パラメタに指定するファイル名にワイルドカードを使用する場合は、ファイル名全体をワイルドカード(アスタリスクのみ)とすることをお勧めします。ファイル名の一部をワイルドカードにする場合は、転送したファイルが想定するファイル名となるかどうかを確認のうえ利用してください。
- 1ファイルあたりのサイズは、2GB以下としてください。

9.2.5 FCTRLCLI.EXE ファイル操作コマンド

記述形式

```
FCTRLCLI.EXE HostName {/DELETE | /RENAME | /MKDIR | /RMDIR}
        /F:FilePathName [/FN:FileName]
        [/UN:UserName] [/LG:LoginGroup]
        [/RC:nn /RI:nnnn]
```

機能説明

クライアント上のファイルまたはディレクトリを操作するコマンドです。ファイル操作では、ファイルの削除およびファイル名を変更することができます。ディレクトリ操作では、ディレクトリを作成および削除することができます。Windowsのディレクトリ共有を利用したファイル操作(Windows共有方式)、またはFTPによるファイル操作(FTP方式)を行います。

Systemwalker Operation Managerがインストールされていない操作先サーバを操作することもできます。

パラメタ

HostName

ファイル操作元のクライアントのホスト名を指定します。HostNameに、半角英数字15文字以内でホスト名を指定してください。空白文字を含む場合は、ダブルクォーテーション(")で囲んでください。

/DELETE

ファイルを削除します。

/RENAME

ファイル名を変更します。

/MKDIR

ディレクトリを作成します。

/RMDIR

ディレクトリを削除します。

/F:FilePathName

対象のファイルまたはディレクトリのパス名を指定します。/DELETEまたは/RENAMEを指定した場合は、ファイル名には、ワイルドカードも指定できます。

FilePathNameに、URL形式のファイルのパス名を次のいずれかの方法で指定してください。パス名は255文字以内で指定し、空白文字を含む場合はダブルクォーテーション(")で囲んでください。

ローカルドライブパス指定:

ドライブ名を含んだパス名を指定します。

“file://”文字列に続いて、任意のファイルのパス名を指定します。

例 file://C:¥DATABASE¥DATABASE.DB

UNC形式指定(Windows共有方式):

“¥¥コンピュータ名¥共有名”の形式でパスを指定します。コンピュータ名はWindowsのコンピュータ名を半角英数字15文字以内で指定してください。共有名は80バイト以内で指定してください。

“file://”文字列に続いて、任意のファイルのパス名を指定します。

例 file://¥¥foo¥SHARE¥DATABASE¥DATABASE.DB

FTPサーバ上のパス指定(FTP方式):

FTPサーバに対するパス名を指定します。

パス名は、フルパスまたはFTPでログイン時のカレントディレクトリからの相対パスで指定することができます。“ftp://”文字列に続いて、“サーバ名/任意のファイルのパス名”の形式で指定します。

例)フルパスで指定する場合：

ftp://foo.bar.com//Usr/DataBase/DataBase.DB

相対パスで指定する場合：

ftp://foo.bar.com/DataBase/DataBase.DB

/FN:FileName

変更先のファイル名を指定します。ワイルドカードも指定できます。FileNameに、ファイル名を255文字以内で指定し、空白文字を含む場合はダブルクォーテーション(")で囲んでください。

本パラメタは、/RENAMEを指定した場合のみ有効です。

/UN:UserName

FilePathNameで指定したファイルまたはディレクトリにアクセスするユーザ名を指定します。

UserNameは、パスワード管理簿にあらかじめ設定したユーザ名を指定します。UserNameに、ユーザ名を半角英数字20文字以内で指定してください。空白文字を含む場合は、ダブルクォーテーション(")で囲んでください。

FilePathNameがローカルドライブパスまたはUNC形式で指定した場合は本指定は無視されます。この場合、クライアント業務連携が動作しているクライアントにログインしているユーザ権限でアクセスします。FTPサーバ上のパスで指定した場合は、必ず本指定を行ってください。

/LG:*LoginGroup*

パスワード管理簿のログイン定義名を指定します。LoginGroupに、ログイン定義名を256文字以内で指定し、空白文字を含む場合はダブルクォーテーション(")で囲んでください。

/UNの指定をした場合は必ず本指定を行ってください。

/RC:*nn*

ファイルの操作でエラーが発生した場合のリトライ回数を指定します。

nnに0～99の整数を指定してください。それ以外の値を設定した場合は、エラーになります。省略した場合は、リトライを行いません。

/RI:*nnnn*

エラーが発生した場合のリトライ間隔を指定します。本指定は、/RC:nnで1以上を指定した場合のみ有効となります。

nnnnに0～9999の整数を秒単位で指定してください。それ以外の値を設定した場合は、エラーになります。省略値は0です。

復帰値

復帰値	意 味
0	正常終了
0以外	異常終了

コマンド格納場所

Systemwalker Operation Managerインストールディレクトリ¥MpWalker.JM¥bin

記述例

クライアント“foo”に対して、クライアント上のローカルファイル“c:¥database¥master¥database.db”を削除します。削除に失敗した場合は、60秒間隔で最大3回リトライします。

```
FCTRLCLI.EXE foo /DELETE /F:file://c:¥database¥master
¥database.db /RC:3 /RI:60
```

処理結果

コマンドの処理が終了すると、終了時の情報を標準出力に出力します。

出力される内容を以下に示します。

```
[処理開始日時] YYYY-MM-DD HH:MM:SS
[クライアントホスト名] HostName
[サーバホスト／ユーザ名] サーバホスト名／ユーザ名 (*)
[リトライ回数] リトライ回数 (*)
[発生したエラー] エラーメッセージ (*)
[ファイル操作種別] 操作ファイル／ディレクトリ名 変更後ファイル名 (**)
(***)
[リトライ回数] リトライ回数 (**)
[発生したエラー] エラーメッセージ (**)
[処理終了日時] YYYY-MM-DD HH:MM:SS
```

(*):

/UNを指定した場合に出力される情報です。

(**):

ワイルドカードが指定された場合、それぞれのファイルについての情報も出力されます。

(***):

変更後ファイル名は、/RENAMEが指定された場合のみ出力されます。

[発生したエラー]には、正常終了の場合は“なし”が出力されます。強制終了やエラーの場合はメッセージが出力されます。メッセージ内容については“Systemwalker Operation Manager メッセージ集”を参照してください。

“ファイル操作種別”は、“ファイル削除”、“ファイル名変更”、“ディレクトリ作成”、“ディレクトリ削除”のいずれかが出力されます。

ファイル操作を行う前にエラーが発生した場合は、以下の形式でメッセージが出力されます。

```
[処理開始日時]  YYYY-MM-DD HH:MM:SS
[コマンドエラー]  エラーメッセージ
[処理終了日時]  YYYY-MM-DD HH:MM:SS
```

メッセージ内容については“Systemwalker Operation Manager メッセージ集”を参照してください。

使用上の注意

- 本コマンドを使用する場合は、クライアント側でクライアント業務連携を起動させておいてください。
- 事前にパスワード管理コマンドを使って、パスワード管理簿の設定を行ってください。詳細は“9.3.1 LGMANAGE.EXE パスワード管理コマンド”を参照してください。
- FTPサーバ上のパス指定でファイルを操作できるFTPサーバおよびファイルは、“DIR”コマンドの結果が以下の形式(パーミッション、リンク数、所有者、グループ、サイズ、日付、時刻、ファイル名の順に英数字表記)で表示されるサーバおよびファイルに限ります。日本語を含むファイル名については、“DIR”コマンドの結果でファイル名が正しく表示されるファイルを扱うことができます。ファイル名が文字化けして表示されるファイルは扱うことができません。

  ```
  -rw-rw-r--  1 xxxxx  xxxxxx     605877 Dec 19 18:44 xxxxx.xxx
  -rw-rw-r--  1 xxxxx  xxxxxx     560661 Dec 19 18:59 xxxxx.xxx
  drwx------  4 xxxxx  xxxxx          96 Jan 20  1998 xxxxx
  -rw-rw-r--  1 xxxxx  xxxxx        6042 Jan 25 13:38 xxxxxxxxxx
  -rw-rw-r--  1 xxxxx  xxxxx        5433 Feb 3  17:11 xxxxx.xxx
  -rw-rw-r--  1 xxxxx  xxxxx        2155 Feb 3  17:35 xxxxx.xxx
  -rw-rw-r--  1 xxxxx  xxxxx        3090 Feb 1  20:13 xxxxx.xxx
  ```

  FTPサーバ上のパス指定が可能であるかどうかの確認方法は、“9.1.2 FCOPYSVR.EXE ファイル転送コマンド”を参照してください。

  出荷状態の日本語Solaris 2.5以降に付属のFTPサーバでは、ディレクトリの日付・時刻が日本語で表示されます。上記の形式で表示するためには、/etc/default/initファイルに以下の記述を追加してください。

  ```
  LC_TIME=C
  ```

  日付・時刻表示が日本語の場合、OSやFTPサーバのマニュアルを参照し英語表示となるように設定してください。

- 本コマンドの/fパラメタに指定したファイル名に、ワイルドカードを指定した場合は、次の注意が必要です。

  同じクライアントに対して、同時に実行するクライアント業務連携コマンドで処理するファイル数を含めて、合計500ファイル以下となるようにしてください。

- パラメタに指定するファイル名にワイルドカードを使用する場合は、ファイル名全体をワイルドカード(アスタリスクのみ)とすることをお勧めします。ファイル名の一部をワイルドカードにする場合は、操作したファイルが想定するファイル名となるかどうかを確認のうえ利用してください。

・ 1ファイルあたりのサイズは、2GB以下としてください。

9.2.6 FPRSCLI.EXE ファイル圧縮/伸長コマンド

記述形式

```
FPRSCLI.EXE HostName {/P | /E} /S:SourceName /D:DestName
        [/RC:nn /RI:nnnn]
```

機能説明

クライアントでファイルの圧縮および伸長を行います。

転送元でファイルを圧縮して、転送先でファイルを伸長するといった使い方ができ、ファイルの転送時間が短縮され、ネットワーク上の負荷を軽減できます。

パラメタ

HostName

ファイルの圧縮/伸長元のクライアントのホスト名を指定します。HostNameに、半角英数字15文字以内でホスト名を指定します。空白文字を含む場合は、ダブルクォーテーション(")で囲んでください。

/P

ファイルを圧縮します。

/E

ファイルを伸長します。

/S:SourceName

圧縮/伸長元ファイルのパス名をローカルドライブパスで指定します。圧縮元ファイルの場合は、ファイル名には、ワイルドカードも指定できます。

SourceNameに、255文字以内でパス名を指定します。空白文字を含む場合は、ダブルクォーテーション(")で囲んでください。

/D:DestName

圧縮/伸長先のファイルのパス名をローカルドライブパスで指定します。伸長先のファイルの場合は、ファイル名には、ワイルドカードで指定もできます。

DestNameに、255文字以内でパス名を指定します。空白文字を含む場合は、ダブルクォーテーション(")で囲んでください。

すでに存在するファイル名を指定した場合は、上書きされます。また、/Sに指定するファイルのパス名(SourceName)と同じパス名を指定しないでください。

/RC:nn

ファイルの圧縮/伸長でエラーが発生した場合のリトライ回数を指定します。

nnに0～99の整数を指定してください。それ以外の値を設定した場合は、エラーになります。省略した場合は、リトライを行いません。

圧縮元または伸長先のファイルにワイルドカードを指定した場合は、失敗したファイルのみリトライを行います。

/RI:nnnn

エラーが発生した場合のリトライ間隔を指定します。

nnnnに0～9999の整数を秒単位で指定してください。それ以外の値を設定した場合は、エラーになります。省略値は0です。

本パラメタは、/RCパラメタで1以上を指定した場合に有効となります。

復帰値

復帰値	意 味
0	正常終了
0以外	異常終了

コマンド格納場所

Systemwalker Operation Managerインストールディレクトリ¥MpWalker.JM¥bin

記述例

クライアント“foo”に対して、クライアント上のローカルファイル“c:¥database¥master¥*.*”を“d:¥database¥database.prs”に圧縮します。圧縮に失敗した場合は、60秒間隔で最大3回リトライします。

```
FPRSCLI.EXE foo /P /S:c:¥database¥master¥*.* /D:d:¥database
¥database.prs
        /RC:3 /RI:60
```

処理結果

コマンドの処理が終了すると、終了時の情報を標準出力に出力します。

出力される内容を以下に示します。

```
[処理開始日時] YYYY-MM-DD HH:MM:SS
[クライアントホスト名] HostName
[ファイル圧縮／伸長] 圧縮／伸長元ファイル名 圧縮／伸長先ファイル名 (*)
[リトライ回数] リトライ回数 (*)
[発生したエラー] エラーメッセージ (*)
[処理終了日時] YYYY-MM-DD HH:MM:SS
```

(*):

ワイルドカードが指定された場合、それぞれのファイルについての情報も出力されます。

[発生したエラー]には、正常終了の場合は“なし”が出力されます。強制終了やエラーの場合はメッセージが出力されます。メッセージ内容については“Systemwalker Operation Manager メッセージ集”を参照してください。

圧縮/伸長処理を行う前にエラーが発生した場合は、以下の形式でメッセージが出力されます。

```
[処理開始日時] YYYY-MM-DD HH:MM:SS
[コマンドエラー] エラーメッセージ
[処理終了日時] YYYY-MM-DD HH:MM:SS
```

メッセージ内容については“Systemwalker Operation Manager メッセージ集”を参照してください。

使用上の注意

- 本コマンドを使用する場合は、クライアント側でクライアント業務連携を起動させておいてください。
- UNIX版 Systemwalker Operation Managerが動作しているサーバ上でfprssvrコマンドの/wオプションをつけて圧縮されたファイルを、Windows上で本コマンドを使って伸長した場合、次のように展開されます。
  - 圧縮元で書込みが誰にも許可されていない場合は、Windows上では読み取り専用属性になります。

ー 圧縮元で書込みがowner/group/otherのいずれかに許されていた場合は、Windows上では読み書き可能な属性になります。

- 本コマンドで圧縮したファイルをUNIX版 Systemwalker Operation Managerが動作しているサーバ上で伸長した場合、展開後のファイル情報は以下の表のようになります。

圧縮前のWindows上でのファイル属性	UNIX上で展開後のファイルの情報
読み取り専用	・所有者／グループ情報は、コマンドを実行したユーザの情報となる。 ・モードはすべてのユーザが読込み可能な0444となる。(注1)
読み取り専用以外	・所有者／グループ情報は、コマンドを実行したユーザの情報となる。 ・モードはすべてのユーザが読み書き可能な0666となる。(注2)

注1）
すべてのユーザから読込み可能な状態で、ls -lコマンドで-r--r--r--と表示されるファイルのパーミッションをさします。

注2）
すべてのユーザから読み書き可能な状態で、ls -lコマンドで-rw-rw-rw-と表示されるファイルのパーミッションをさします。

- 圧縮するユーザに、読み取り可能のパーミッションが設定されていない場合は圧縮することはできません。ただし、スーパーユーザは圧縮可能です。
- コマンドの強制終了は、1つのファイルの圧縮/伸長が完了した時点で行われます。このため、大きなファイルを圧縮中に強制終了を行うと、コマンドの終了を待たされる場合があります。
- 本コマンドの/sパラメタに指定したファイル名に、ワイルドカードを指定した場合は、次の注意が必要です。

  同じクライアントに対して、同時に実行するクライアント業務連携コマンドで処理するファイル数を含めて、合計500ファイル以下となるようにしてください。
- パラメタに指定するファイル名にワイルドカードを使用する場合は、ファイル名全体をワイルドカード(アスタリスクのみ)とすることをお勧めします。ファイル名の一部をワイルドカードにする場合は、圧縮/伸長したファイルが想定するファイル名となるかどうかを確認のうえ利用してください。
- 本コマンドの実行時には、クライアント上の以下の場所に、一時ファイルが作成されます。

  ー 環境変数TEMPに指定されているディレクトリ

  TEMP領域が不足した場合、圧縮・伸長することはできません。一時ファイルのサイズは、一度に圧縮・伸長するファイルサイズの合計となります。
- 1ファイルあたりのサイズは、2GB以下としてください。

9.3 業務連携管理コマンド

業務連携管理コマンドについて説明します。

9.3.1 LGMANAGE.EXE パスワード管理コマンド

記述形式

```
LGMANAGE.EXE ADD /LG:LoginGroup /UN:UserName /PW:Password
LGMANAGE.EXE DEL /LG:LoginGroup /UN:UserName
LGMANAGE.EXE MOD /LG:LoginGroup /UN:UserName /PW:Password
```

```
LGMANAGE.EXE REF /LG:LoginGroup
LGMANAGE.EXE {HELP | ?}
LGMANAGE.EXE
```

機能説明

業務連携で使用するパスワード管理簿を管理するコマンドです。パスワード管理簿は、業務連携コマンドで指定するユーザとパスワードをログイン定義名ごとにファイルとして管理するものです。

パラメタ

ADD/DEL/MOD/REF

パスワード管理簿への操作種別を指定します。

ADD:

パスワード管理簿へユーザ名およびパスワードを追加します。

DEL:

パスワード管理簿からユーザ名およびパスワードを削除します。

MOD:

パスワード管理簿のユーザ名に対応したパスワードを変更します。

REF:

パスワード管理簿に登録されているユーザ名の一覧を参照します。

/LG:*LoginGroup*

パスワード管理簿のログイン定義名を指定します。

業務連携コマンドを実行するユーザが所属するOSのグループ名をLoginGroupに指定します。ログイン定義名は、英大文字と英小文字は区別されません。200文字以内で指定してください。

/UN:*UserName*

パスワード管理簿でパスワードを管理するユーザ名を指定します。UserNameで指定したユーザ名が、業務連携コマンドで指定した連携先のサーバ(FTPサーバなど)のログインユーザになります。UserNameに、20文字以内で指定します。

UNC形式指定の場合は、UserNameは“ドメイン名¥ユーザ名”または“サーバ名¥ユーザ名”のいずれかの形式で指定してください。この場合、ドメイン名またはサーバ名は半角英数字15文字以内で、ユーザ名は半角英数字20文字以内で指定してください。

なお、FTPサーバ上のパス指定の場合は、UserNameは“ユーザ名@ドメイン名”またはローカルアカウントを指定してください。

FTPサーバ上のパス指定の場合は、UserNameが半角英数字20文字以内になるように指定してください。

/PW:*Password*

パスワード文字列を指定します。Passwordに指定したパスワード文字列が、業務連携コマンドで指定した連携先のサーバ(FTPサーバなど)のログインパスワードになります。Passwordに1文字以上50文字以下の半角英数字を指定します。変更の場合は新しいパスワードを指定します。

HELP|?

“help”または“?”で起動した場合、本コマンドの使用方法を表示します。

パラメタ指定なし

対話型モードによりパラメタ入力を行います。

入力項目が表示され、すべての項目を対話型メニューで入力します。

復帰値

復帰値	意味
0	正常終了
0以外	異常終了。エラーの内容はエラーメッセージを確認してください。

コマンド格納場所

Systemwalker Operation Managerインストールディレクトリ¥MpWalker.JM¥bin

使用上の注意

- パスワード管理簿ファイルの格納先については、“Systemwalker Operation Manager 導入ガイド”の、“パスワード管理簿の定義”を参照してください。
- パスワード管理簿ファイルは、ジョブスケジューラのスケジュール情報の一部として、ポリシー情報の配付の対象となります。
- ユーザ名およびパスワードを追加する場合で、指定したログイン定義名のパスワード管理簿ファイルが存在しない時は、パスワード管理簿ファイルを作成し、ユーザ名およびパスワードを追加します。また、パスワード管理簿ファイルに同一ユーザ名が存在する場合は、ユーザ名を追加せずにエラーとなります。
- パスワード管理簿は、業務連携機能を使用するサーバごとに作成し、管理してください。

第10章　業務連携コマンド【UNIX版】

本章では、Systemwalker Operation Managerが提供する業務連携コマンドについて説明します。なお、コマンドは、Systemwalker Operation Managerのサーバ機能がインストールされているサーバで起動してください。

10.1　サーバ業務連携コマンド

サーバ業務連携コマンドは、サーバの業務システムの構築を支援するためのコマンドです。

参考

サーバ業務連携コマンドは、バッチファイルを作成して登録することも可能です。

10.1.1　execasvr アプリケーション起動コマンド

記述形式

```
execasvr [/un:UserName] [/rc:nn /ri:nnnn]
        /ae:ExecuteAppli [AppliParam]
```

機能説明

サーバにインストールされている任意のアプリケーションを起動します。起動アプリケーションは、バックグラウンドで動作し、画面には表示されません。また、起動したアプリケーションは、終了の待ち合わせは行いません。

パラメタ

/un:UserName

指定したアプリケーションをどのユーザの権限で起動するかを指定します。

UserNameが半角英数字20文字以内になるように指定してください。

「-」、「_」以外の特殊文字および空白文字は使用できません。

本パラメタを省略した場合は、コマンド起動時のユーザの権限でアプリケーションは起動されます。

/rc:nn

アプリケーションの起動でエラーが発生した場合のリトライ回数を指定します。

nnに0～99の整数を指定します。それ以外の値を設定すると、エラーになります。省略した場合はリトライを行いません。

/ri:nnnn

エラーが発生した場合のリトライ間隔を指定します。

nnnnに0～9999の整数を秒単位で指定します。それ以外の値を設定すると、エラーになります。省略値は0です。

本パラメタは、/rcパラメタで1以上を指定した場合に有効となります。

/ae:ExecuteAppli

起動アプリケーション名を絶対パス名で指定します。ExecuteAppliに、アプリケーション名を指定します。起動アプリケーション名に空白文字がある場合は、ダブルクォーテーション「"」で囲んでください。ExecuteAppliに指定できる文字数はAppliParamを含めて511文字までです。

AppliParam

ExecuteAppliで指定したアプリケーションに対するパラメタを指定します。パラメタを複数指定する場合は、それぞれの間に空白文字を入れてください。各パラメタに空白文字を含む場合は、その両端をダブルクォーテーション「"」で囲んでください。

復帰値

復帰値	意 味
0	正常終了
0以外	異常終了

コマンド格納場所

Solaris	/opt/FJSVsnjss/bin
HP-UX	/opt/FHPsnjss/bin
AIX	/opt/FAIXsnjss/bin
Linux	/opt/FJSVsnjss/bin

記述例

“/usr/local/bin/test”を“test.txt”のパラメタ付きで起動します。

```
execasvr /ae:/usr/local/bin/test test.txt
```

処理結果

コマンドの処理が終了すると、終了時の情報を標準出力に出力します。

出力される内容を以下に示します。

```
[処理開始日時] YYYY-MM-DD HH:MM:SS
[起動アプリケーション] 起動アプリケーション名
[リトライ回数] リトライ回数
[発生したエラー] エラーメッセージ
[処理終了日時] YYYY-MM-DD HH:MM:SS
```

[発生したエラー]には、正常終了の場合は“なし”が出力されます。強制終了やエラーの場合はメッセージが出力されます。メッセージ内容については、“Systemwalker Operation Manager メッセージ集”を参照してください。

アプリケーション起動を行う前にエラーが発生した場合は、以下の形式でメッセージが出力されます。

```
[処理開始日時] YYYY-MM-DD HH:MM:SS
[コマンドエラー] エラーメッセージ
[処理終了日時] YYYY-MM-DD HH:MM:SS
```

メッセージ内容については、“Systemwalker Operation Manager メッセージ集”を参照してください。

使用上の注意

- 本コマンドは、起動したアプリケーションの処理を監視しません。処理結果を知る必要がある場合は、アプリケーション自身に終了コードの通知機能を用意してください。
- 本コマンドでは、キーボードからの入力を必要とするアプリケーションは起動できません。
- 本コマンドを強制終了した場合、起動したアプリケーションに対しての強制終了を行うことはできません。

- コマンドの強制終了は、アプリケーションの起動が完了した時点で行われます。このため、起動に時間がかかるアプリケーションの場合、コマンドの終了を待たされる場合があります。
- /un:UserNameを指定し、ユーザ権限でアプリケーションを起動する場合、本コマンドを起動するユーザはroot権限を持っている必要があります。
- シェルスクリプトを起動する場合は、以下のようにスクリプトの先頭行にシェル名を記述してください。

  例：

```
#! /bin/sh
```

10.1.2 fcopysvr ファイル転送コマンド

記述形式

```
fcopysvr /s:SourceName [/su:SourceUserName]
        /d:DestName [/du:DestUserName]
        [/lg:LoginGroup] [/a | /b] [/rc:nn /ri:nnnn]
```

機能説明

サーバ間でのファイルの転送を行うコマンドです。ローカルファイル(NFS接続したディレクトリを含む)またはFTPサーバ上のファイルを転送することができます。Systemwalker Operation Managerがインストールされていない転送先のサーバにも転送できます。

パラメタ

/s:SourceName

転送元のファイルのパス名をファイル名も含めて指定します。ファイル名には、ワイルドカードも指定できます。

ワイルドカードは、シェルによって特別な意味を持つ文字(「*」など)があるため、必要に応じてSourceNameを一対の引用符(「'」または「"」)で囲む必要があります。

SourceNameには、ファイルのパス名を次のいずれかの方法で、URL形式で指定してください。パス名部分は255文字以内で指定し、空白文字を含む場合はダブルクォーテーション「"」で囲んでください。

ローカルパス指定:

ルートディレクトリからの絶対パスを指定します。

“file://”文字列に続いて、任意のファイルのパス名を指定します。

例)ファイルのパスが“/usr/database/database.db”であった場合

file:///usr/database/database.db

FTPサーバ上のパス指定:

FTPサーバに対するパスを指定します。

パス名は、フルパスまたはFTPでログイン時のカレントディレクトリからの相対パスで指定することができます。“ftp://”文字列に続いて、“サーバ名/任意のファイルのパス名”の形式で指定します。

例)フルパスで指定する場合：

ftp://foo.bar.com//usr/database/database.db

相対パスで指定する場合：

ftp://foo.bar.com/database/database.db

/su:SourceUserName

SourceNameで指定したファイルにアクセスするためのユーザ名を指定します。

SourceUserNameは、パスワード管理簿にあらかじめ設定したユーザ名を指定します。SourceUserNameが半角英数字20文字以内になるように指定してください。空白文字を含む場合は、ダブルクォーテーション「"」で囲んでください。

SourceNameの指定がローカルパスの場合は、本指定は無視されます。FTPサーバ上のパスを指定した場合は、必ず本指定を行ってください。

/d:*DestName*

転送先のファイルのパス名をファイル名も含めて指定します。ファイル名には、ワイルドカードで指定もできます。

ワイルドカードは、シェルによって特別な意味を持つ文字(「*」など)があるため、必要に応じてDestNameを一対の引用符(「'」または「"」)で囲む必要があります。

ファイルのパス名の指定方法は/s:SourceNameと同様です。/s:SourceNameを参照してください。

すでに存在するファイル名を指定した場合は、上書きされます。また、/sに指定するパス名(SourceName)と同じパス名を指定しないでください。

/du:*DestUserName*

DestNameで指定したファイルにアクセスするためのユーザ名を指定します。

DestUserNameには、パスワード管理簿にあらかじめ設定したユーザ名を指定します。DestUserNameが半角英数字20文字以内になるように指定してください。空白文字を含む場合は、ダブルクォーテーション「"」で囲んでください。

DestNameの指定がローカルパスの場合は、本指定は無視されます。FTPサーバ上のパスの場合は、必ず本指定を行ってください。

/lg:*LoginGroup*

パスワード管理簿のログイン定義名を指定します。LoginGroupに、256文字以内でログイン定義名を指定します。空白文字を含む場合はダブルクォーテーション「"」で囲んでください。

/suまたは/duを指定した場合は必ず本指定を行ってください。

/a

ASCIIテキストモードでファイルを転送します。

本指定は、FTPサーバ上のファイルを転送する場合のみ有効となります。

/b

バイナリモードでファイルを転送します。

本指定は、FTPサーバ上のファイルを転送する場合のみ有効となります。

/a、/bともに指定されていない場合は、バイナリモードでファイルを転送します。

/rc:*nn*

ファイルの転送でエラーが発生した場合のリトライ回数を指定します。

nnに0～99の整数を指定します。それ以外の値を設定すると、エラーとなります。省略した場合はリトライを行いません。

転送元ファイルまたは転送先ファイルにワイルドカードを指定した場合、失敗したファイルのみリトライが行われます。

/ri:*nnnn*

エラーが発生した場合のリトライ間隔を指定します。

nnnnに0～9999の整数を秒単位で指定します。それ以外の値を設定した場合は、エラーとなります。省略値は0です。

本パラメタは、/rcパラメタで1以上を指定した場合に有効となります。

復帰値

復帰値	意 味
0	正常終了
0以外	異常終了

コマンド格納場所

Solaris	/opt/FJSVsnjss/bin
HP-UX	/opt/FHPsnjss/bin
AIX	/opt/FAIXsnjss/bin
Linux	/opt/FJSVsnjss/bin

記述例

サーバ上のローカルファイル“/usr/database/master/database.db”をローカルファイル“/usr/database/database.db”へ転送します。転送に失敗した場合は、60秒間隔で最大3回リトライします。

```
fcopysvr /s:file:///usr/database/master/database.db
        /d:file:///usr/database/database.db /rc:3 /ri:60
```

処理結果

コマンドの処理が終了すると、終了時の情報を標準出力に出力します。

出力される内容を以下に示します。

```
[処理開始日時] YYYY-MM-DD HH:MM:SS
[サーバホスト／ユーザ名] サーバホスト名／ユーザ名 (*)
[リトライ回数] リトライ回数 (*)
[発生したエラー] エラーメッセージ (*)
[ファイル転送] 転送元ファイル名 転送先ファイル名 (**)
[リトライ回数] リトライ回数 (**)
[発生したエラー] エラーメッセージ (**)
[処理終了日時] YYYY-MM-DD HH:MM:SS
```

(*):

/suまたは/duを指定した場合に出力される情報です。

(**):

ワイルドカードが指定された場合、それぞれのファイルについての情報も出力されます。

[発生したエラー]には、正常終了の場合は“なし”が出力されます。強制終了やエラーになった場合はメッセージが出力されます。メッセージ内容については、“Systemwalker Operation Manager メッセージ集”を参照してください。

ファイルの転送を行う前にエラーが発生した場合は、以下の形式でメッセージが出力されます。

```
[処理開始日時] YYYY-MM-DD HH:MM:SS
[コマンドエラー] エラーメッセージ
[処理終了日時] YYYY-MM-DD HH:MM:SS
```

メッセージ内容については、“Systemwalker Operation Manager メッセージ集”を参照してください。

使用上の注意

- 事前にパスワード管理コマンドを使って、パスワード管理簿の設定を行ってください。詳細は“10.3.1 lgmanage パスワード管理コマンド”を参照してください。
- FTPサーバ上のパス指定でファイルを転送する場合、転送元および転送先の両方をFTPサーバ上のパス指定で転送することはできません。転送元と転送先に指定できる組み合わせは以下のとおりです。いずれかの組み合わせで指定してください。

	転送元ファイル	転送先ファイル
1	ローカルパス形式	ローカルパス形式
2	ローカルパス形式	FTPサーバパス形式
3	FTPサーバパス形式	ローカルパス形式

- FTPサーバ上のパス指定でファイル転送ができるFTPサーバおよびファイルは、“dir”コマンドの結果が以下の形式(パーミッション、リンク数、所有者、グループ、サイズ、日付、時刻、ファイル名の順に英数字表記)で表示されるサーバおよびファイルに限ります。日本語を含むファイル名については、“dir”コマンドの結果でファイル名が正しく表示されるファイルを扱うことができます。ファイル名が文字化けして表示されるファイルは扱うことができません。

```
-rw-rw-r--  1 xxxxx  xxxxxx     605877 Dec 19 18:44 xxxxx.xxx
-rw-rw-r--  1 xxxxx  xxxxxx     560661 Dec 19 18:59 xxxxx.xxx
drwx------  4 xxxxx  xxxxx          96 Jan 20  1998 xxxxx
-rw-rw-r--  1 xxxxx  xxxxx        6042 Jan 25 13:38 xxxxxxxxxx
-rw-rw-r--  1 xxxxx  xxxxx        5433 Feb 3  17:11 xxxxx.xxx
-rw-rw-r--  1 xxxxx  xxxxx        2155 Feb 3  17:35 xxxxx.xxx
-rw-rw-r--  1 xxxxx  xxxxx        3090 Feb 1  20:13 xxxxx.xxx
```

FTPサーバ上のパス指定でファイル転送が可能かどうかの確認は、次の方法で行ってください。

Windowsの場合:

－ コマンドプロンプト上でftpコマンドを実行します。

UNIXの場合:

－ サーバコンソールやtelnet接続したコンソール上でftpコマンドを実行します。

以下に、UNIX上で確認する場合の手順を示します。

```
$ ftp XXXXX                          ←ftpコマンド(XXXXX:ftpサーバ名)
Connected to XXXXX.
220 XXXXX FTP server (UNIX(r) System V Release 4.0) ready.
Name (XXXXX:YYYYY): YYYYY            ←ftpサーバのユーザ名指定 (YYYYY:ユーザ名)
331 Password required for YYYYY.
Password:                            ←パスワード入力
230 User YYYYY logged in.
ftp> dir                             ←dirコマンド入力
200 PORT command successful.
          ↓
dirの結果が出力されます。
```

日付・時刻表示が日本語の場合、OSやFTPサーバのマニュアルを参照し英語表示となるように設定してください。

なお、出荷状態の日本語Solaris 2.5 オペレーティングシステム以上に付属のFTPサーバでは、ディレクトリの日付・時刻が日本語で表示されます。上記の形式で表示するためには、/etc/default/initファイルに以下の記述を追加してください。

```
LC_TIME=C
```

- コマンドの強制終了は、1つのファイルの転送が完了した時点で行われます。このため、大きなファイルを転送中に強制終了を行うと、コマンドの終了を待たされる場合があります。

- 本コマンドの/sパラメタに指定したファイル名に、ワイルドカードを指定した場合は、次の注意が必要です。

  一度に処理するファイル数は、同時に実行するほかのサーバ業務連携コマンドで処理するファイル数を含めて、合計500ファイル以下としてください。

- パラメタに指定するファイル名にワイルドカードを使用する場合は、ファイル名全体をワイルドカード(アスタリスクのみ)とすることをお勧めします。ファイル名の一部をワイルドカードにする場合は、転送したファイルが想定するファイル名となるかどうかを確認のうえ利用してください。
- 1ファイルあたりのサイズは、2GB以下としてください。

10.1.3 fctrlsvr ファイル操作コマンド

記述形式

```
fctrlsvr {/delete | /rename | /mkdir | /rmdir} /f:FilePathName
        [/fn:FileName] [/un:UserName] [/lg:LoginGroup]
        [/rc:nn /ri:nnnn]
```

機能説明

サーバ上のファイルまたはディレクトリを操作するコマンドです。ファイル操作では、ファイルの削除およびファイル名を変更することができます。ディレクトリ操作では、ディレクトリの作成および削除することができます。本コマンドで操作可能なファイルは、ローカルファイル(NFS接続ディレクトリを含む)およびFTPサーバ上のファイルです。

Systemwalker Operation Managerがインストールされていない操作先サーバを操作することもできます。

パラメタ

/delete

ファイルを削除します。

/rename

ファイル名を変更します。

/mkdir

ディレクトリを作成します。

/rmdir

ディレクトリを削除します。

/f:FilePathName

対象のファイルまたはディレクトリのパス名を指定します。/deleteまたは/renameを指定した場合は、ファイル名には、ワイルドカードも指定できます。

ワイルドカードは、シェルによって特別な意味を持つ文字(「*」など)があるため、必要に応じてFilePathNameを一対の引用符(「'」または「"」)で囲む必要があります。

FilePathNameは、URL形式のファイルのパス名を次のいずれかの方法で指定してください。パス名は255文字以内で指定し、空白文字を含む場合はダブルクォーテーション「"」で囲んでください。

ローカルパス指定:

ルートディレクトリからの絶対パス名を指定します。

“file://”文字列に続いて、任意のファイルのパス名を指定します。

例) file:///usr/database/database.db

FTPサーバ上のパス指定:

FTPサーバに対するパス名を指定します。

パス名は、フルパスまたはFTPでログイン時のカレントディレクトリからの相対パスで指定することができます。“ftp://”文字列に続いて、“サーバ名/任意のファイルのパス名”の形式で指定します。

例)フルパスで指定する場合：

ftp://foo.bar.com//usr/database/database.db

相対パスで指定する場合：

ftp://foo.bar.com/database/database.db

/fn:FileName

変更先のファイル名を指定します。ワイルドカードも指定できます。

ワイルドカードは、シェルによって特別な意味を持つ文字(「*」など)があるため、必要に応じてFileNameを一対の引用符(「'」または「"」)で囲む必要があります。

FileNameに、255文字以内でファイル名を指定します。空白文字を含む場合はダブルクォーテーション「"」で囲んでください。

本パラメタは、/renameを指定した場合のみ有効です。

/un:UserName

FilePathNameで指定したファイルまたはディレクトリにアクセスするユーザ名を指定します。

UserNameには、パスワード管理簿にあらかじめ設定したユーザ名を指定します。UserNameが半角英数字20文字以内になるように指定してください。空白文字を含む場合は、ダブルクォーテーション「"」で囲んでください。

FilePathNameがローカルパスで指定した場合は本指定は無視されます。

FTPサーバ上のパスで指定した場合は、必ず本指定を行ってください。

/lg:LoginGroup

パスワード管理簿のログイン定義名を指定します。LoginGroupは、256文字以内でログイン定義名を指定します。空白文字を含む場合はダブルクォーテーション「"」で囲んでください。

/unの指定をした場合は必ず本指定を行ってください。

/rc:nn

ファイルの操作でエラーが発生した場合のリトライ回数を指定します。

nnは0～99の整数を指定してください。それ以外の値を設定した場合は、エラーになります。省略した場合はリトライを行いません。

/ri:nnnn

エラーが発生した場合のリトライ間隔を指定します。

nnnnに0～9999の整数を秒単位で指定してください。それ以外の値を設定した場合は、エラーになります。省略値は0です。

本パラメタは、/rcパラメタで1以上を指定した場合に有効となります。

復帰値

復帰値	意 味
0	正常終了
0以外	異常終了

コマンド格納場所

Solaris	/opt/FJSVsnjss/bin
HP-UX	/opt/FHPsnjss/bin
AIX	/opt/FAIXsnjss/bin
Linux	/opt/FJSVsnjss/bin

記述例

サーバ上のローカルファイル“/usr/database/master/database.db”をファイル名“database_backup.db”へ変更します。ファイル名変更に失敗した場合は、60秒間隔で最大3回リトライします。

```
fctrlsvr /rename /f:file:///usr/database/master/database.db
        /fn:database_backup.db /rc:3 /ri:60
```

処理結果

コマンドの処理が終了すると、終了時の情報を標準出力に出力します。

出力される内容を以下に示します。

```
[処理開始日時]  YYYY-MM-DD HH:MM:SS
[サーバホスト／ユーザ名]  サーバホスト名／ユーザ名 (*)
[リトライ回数]  リトライ回数 (*)
[発生したエラー]  エラーメッセージ (*)
[ファイル操作種別]  操作ファイル／ディレクトリ名 変更後ファイル名 (**)
(***)
[リトライ回数]  リトライ回数 (**)
[発生したエラー]  エラーメッセージ (**)
[処理終了日時]  YYYY-MM-DD HH:MM:SS
```

(*):

/unを指定した場合に出力される情報です。

(**):

ワイルドカードが指定された場合、それぞれのファイルについての情報も出力されます。

(***):

変更後ファイル名は、/renameが指定された場合のみ出力されます。

[発生したエラー]には、正常終了の場合は“なし”が出力されます。強制終了やエラーの場合はメッセージが出力されます。メッセージ内容については、“Systemwalker Operation Manager メッセージ集”を参照してください。

[ファイル操作種別]は、“ファイル削除”“ファイル名変更”“ディレクトリ作成”“ディレクトリ削除”のいずれかが出力されます。

ファイル操作を行う前にエラーが発生した場合は、以下の形式でメッセージが出力されます。

```
[処理開始日時]  YYYY-MM-DD HH:MM:SS
[コマンドエラー]  エラーメッセージ
[処理終了日時]  YYYY-MM-DD HH:MM:SS
```

メッセージ内容については、“Systemwalker Operation Manager メッセージ集”を参照してください。

使用上の注意

- 事前にパスワード管理コマンドを使って、パスワード管理簿の設定を行ってください。詳細は“10.3.1 lgmanage パスワード管理コマンド”を参照してください。

- FTPサーバ上のパス指定でファイルを操作できるFTPサーバおよびファイルは、“dir”コマンドの結果が以下の形式(パーミッション、リンク数、所有者、グループ、サイズ、日付、時刻、ファイル名の順に英数字表記)で表示されるサーバおよびファイルに限ります。日本語を含むファイル名については、“dir”コマンドの結果でファイル名が正しく表示されるファイルを扱うことができます。ファイル名が文字化けして表示されるファイルは扱うことができません。

```
-rw-rw-r--  1 xxxxx  xxxxxx     605877 Dec 19 18:44 xxxxx.xxx
-rw-rw-r--  1 xxxxx  xxxxxx     560661 Dec 19 18:59 xxxxx.xxx
drwx------  4 xxxxx  xxxxx          96 Jan 20  1998 xxxxx
-rw-rw-r--  1 xxxxx  xxxxx        6042 Jan 25 13:38 xxxxxxxxxxx
-rw-rw-r--  1 xxxxx  xxxxx        5433 Feb 3  17:11 xxxxx.xxx
-rw-rw-r--  1 xxxxx  xxxxx        2155 Feb 3  17:35 xxxxx.xxx
-rw-rw-r--  1 xxxxx  xxxxx        3090 Feb 1  20:13 xxxxx.xxx
```

  FTPサーバ上のパス指定が可能であるかどうかの確認方法は、“10.1.2 fcopysvr ファイル転送コマンド”を参照してください。

  日付・時刻表示が日本語の場合、OSやFTPサーバのマニュアルを参照し英語表示となるように設定してください。

  なお、出荷状態の日本語Solaris 2.5 オペレーティングシステム以上に付属のFTPサーバでは、ディレクトリの日付・時刻が日本語で表示されます。上記の形式で表示するためには、/etc/default/initファイルに以下の記述を追加してください。

```
LC_TIME=C
```

- 本コマンドの/fパラメタに指定したファイル名に、ワイルドカードを指定した場合は、次の注意が必要です。

  一度に処理するファイル数は、同時に実行するほかのサーバ業務連携コマンドで処理するファイル数を含めて、合計500ファイル以下としてください。

- パラメタに指定するファイル名にワイルドカードを使用する場合は、ファイル名全体をワイルドカード(アスタリスクのみ)とすることをお勧めします。ファイル名の一部をワイルドカードにする場合は、操作したファイルが想定するファイル名となるかどうかを確認のうえ利用してください。

- 1ファイルあたりのサイズは、2GB以下としてください。

10.1.4 fprssvr ファイル圧縮/伸長コマンド

記述形式

```
fprssvr {/p [/w] | /e} /s:SourceName /d:DestName
      [/rc:nn /ri:nnnn]
```

機能説明

サーバ上のローカルファイル(NFS接続したディレクトリを含む)の圧縮および伸長を行います。

パラメタ

/p

ファイルを圧縮します。

/w

Windows上で展開できるWindowsプラットフォーム形式で圧縮します。本オプションを指定しない場合は、UNIXプラットフォーム形式で圧縮し、Windows上では伸長できません。

/e

ファイルを伸長します。

/s:SourceName

圧縮/伸長元ファイルのパス名をローカルパスで指定します。ローカルパスは、ルートディレクトリからの絶対パスで、“/usr/database/database.db”のように指定します。FTPサーバ上のパス指定はできません。圧縮元ファイルの場合は、ファイル名には、ワイルドカードも指定できます。

ワイルドカードは、シェルによって特別な意味を持つ文字(「*」など)があるため、必要に応じてSourceNameを一対の引用符(「'」または「"」)で囲む必要があります。

SourceNameは、255文字以内で指定してください。SourceNameに空白文字を含む場合は、ダブルクォーテーション「"」で囲んでください。

/d:DestName

圧縮/伸長先のファイルのパス名をローカルパスで指定します。FTPサーバ上のパス指定はできません。伸長先ファイルの場合は、ファイル名には、ワイルドカードで指定もできます。

ワイルドカードは、シェルによって特別な意味を持つ文字(「*」など)があるため、必要に応じてDestNameを一対の引用符(「'」または「"」)で囲む必要があります。

DestNameは、255文字以内で指定してください。DestNameに空白文字を含む場合は、ダブルクォーテーション「"」で囲んでください。

すでに存在するファイル名を指定した場合は、上書きされます。また、/sに指定したファイルのパス名(SourceName)と同じパス名を指定しないでください。

/rc:nn

ファイルの圧縮/伸長でエラーが発生した場合のリトライ回数を指定します。nnは0～99の整数で指定してください。それ以外の値を設定すると、エラーになります。省略した場合は、リトライを行いません。

圧縮元ファイル、または伸長先ファイルにワイルドカードが指定された場合、失敗したファイルのみリトライを行います。

/ri:nnnn

エラーが発生した場合のリトライ間隔を指定します。nnnnには0～9999の整数を秒単位で指定してください。それ以外の値を設定すると、エラーになります。省略値は0です。

復帰値

復帰値	意 味
0	正常終了
0以外	異常終了

コマンド格納場所

Solaris	/opt/FJSVsnjss/bin
HP-UX	/opt/FHPsnjss/bin
AIX	/opt/FAIXsnjss/bin
Linux	/opt/FJSVsnjss/bin

記述例

サーバ上のローカルファイル“/database/database.prs”を“/database/copy/*.*”に伸長します。伸長に失敗した場合は、60秒間隔で最大3回リトライします。

```
fprssvr /e /s:/database/database.prs /d:/database/copy/*.* /rc:3 /ri:60
```

処理結果

コマンドの処理が終了すると、終了時の情報を標準出力に出力します。

出力される内容を以下に示します。

```
[処理開始日時] YYYY-MM-DD HH:MM:SS
[ファイル圧縮／伸長] 圧縮／伸長元ファイル名 圧縮／伸長先ファイル名 (*)
[リトライ回数] リトライ回数 (*)
[発生したエラー] エラーメッセージ (*)
[処理終了日時] YYYY-MM-DD HH:MM:SS
```

(*):

ワイルドカードが指定された場合、それぞれのファイルについての情報も出力されます。

[発生したエラー]には、正常終了の場合は“なし”が出力されます。強制終了やエラーの場合はメッセージが出力されます。メッセージ内容については、“Systemwalker Operation Manager メッセージ集”を参照してください。

圧縮/伸長処理を行う前にエラーが発生した場合は、以下の形式でメッセージが出力されます。

```
[処理開始日時] YYYY-MM-DD HH:MM:SS
[コマンドエラー] エラーメッセージ
[処理終了日時] YYYY-MM-DD HH:MM:SS
```

メッセージ内容については、“Systemwalker Operation Manager メッセージ集”を参照してください。

使用上の注意

- 本コマンドで圧縮できるファイルは通常のファイルのみです。シンボリックリンクなどのリンクファイルは圧縮することができません。
- 本コマンドで/wオプションを付けて圧縮されたファイル(Windowsプラットフォーム形式)をWindows上で伸長した場合、次のように展開されます。
  - 圧縮元で書込みが誰にも許可されていない場合は、Windowsでは読み取り専用属性になります。
  - 圧縮元で書込みがowner/group/otherのいずれかに許されていた場合は、Windowsでは読み書き可能な属性になります。
- UNIXプラットフォーム形式、Windowsプラットフォーム形式の圧縮ファイルを本コマンドを使用してUNIX上で伸長した場合、展開後のファイル情報は以下の表のようになります。

圧縮ファイルの形式	展開後のファイルの情報 (スーパーユーザが展開した場合)	展開後のファイルの情報 (一般ユーザが展開した場合)
UNIXプラットフォーム形式	所有者／グループ情報は、圧縮元の情報が継承される。 モードは圧縮元の情報が継承される。	所有者／グループ情報は、実行したユーザの情報になる。 モードは圧縮元の情報が継承される。
Windowsプラットフォーム形式	・圧縮前のファイルが読み取り専用のファイルの場合 ・所有者／グループ情報は、コマンドを実行したユーザの情報となる。 ・モードはすべてのユーザが読込み可能な0444となる。(注1) ・圧縮前のファイルが読み取り専用以外のファイルの場合 ・所有者／グループ情報は、コマンドを実行したユーザの情報となる。 ・モードはすべてのユーザが読み書き可能な0666となる。(注2)	

注1)

すべてのユーザから読込み可能な状態で、ls -lコマンドで-r--r--r--と表示されるファイルのパーミッションをさします。

注2)

すべてのユーザから読み書き可能な状態で、ls -lコマンドで-rw-rw-rw-と表示されるファイルのパーミッションをさします。

- 圧縮するユーザに、読み取り可能のパーミッションが設定されていない場合は圧縮することはできません。ただし、スーパーユーザは圧縮可能です。
- コマンドの強制終了は、1つのファイルの圧縮/伸長が完了した時点で行われます。このため、大きなファイルを圧縮中に強制終了を行うと、コマンドの終了を待たされる場合があります。
- 本コマンドの/sパラメタに指定したファイル名に、ワイルドカードを指定した場合は、次の注意が必要です。

  一度に処理するファイル数は、同時に実行するほかのサーバ業務連携コマンドで処理するファイル数を含めて、合計500ファイル以下としてください。
- パラメタに指定するファイル名にワイルドカードを使用する場合は、ファイル名全体をワイルドカード(アスタリスクのみ)とすることをお勧めします。ファイル名の一部をワイルドカードにする場合は、圧縮/伸長したファイルが想定するファイル名となるかどうかを確認のうえ利用してください。
- 本コマンドの実行時には、サーバ上の以下の場所に、一時ファイルが作成されます。

  ー /tmp(Solaris版の場合は/var/tmp)

  /tmp領域が不足した場合、圧縮・伸長することはできません。一時ファイルのサイズは、一度に圧縮・伸長するファイルサイズの合計となります。
- 1ファイルあたりのサイズは、2GB以下としてください。

10.2 クライアント業務連携コマンド

クライアント業務連携コマンドは、クライアントとの業務連携を行うためのコマンドです。

参考

クライアント業務連携コマンドは、バッチファイルを作成して登録することも可能です。

10.2.1 poncli 電源投入コマンド

記述形式

```
poncli HostName
      [/ip:IPAddress /mac:MACAddress /snm:SubnetMask]
      [/wt:WaitTime] [/rc:nn /ri:nnnn]
```

機能説明

Wakeup on LANに対応したクライアントの電源を投入し、離れた場所にあるクライアントを自動的に起動することが可能になります。

また、TCP/IP環境であれば他のサブネット上のクライアントの電源を投入することもできます。

本コマンドは、IPv6通信環境ではサポートされていません。IPv4通信環境、またはデュアルスタックが有効な環境で利用してください。

パラメタ

HostName

電源を投入するクライアントのホスト名を指定します。

HostNameに、半角英数字15文字以内でホスト名を指定します。空白文字を含む場合はダブルクォーテーション「"」で囲んでください。

/ip:*IPAddress*

電源を投入するクライアントのIPアドレスを指定します。IPアドレスは、10進数の文字列で“133.161.222.110”のように指定します。

省略した場合は、ホスト情報定義ファイルから対象クライアントのIPアドレスを読み出します。

/mac:*MACAddress*

電源を投入するクライアントのMACアドレスを指定します。MACアドレスは、16進数の文字列で“00-00-0e-9d-2c-38”のように指定します。省略した場合は、ホスト情報定義ファイルから対象クライアントのMACアドレスを読み出します。

/snm:*SubnetMask*

電源を投入するクライアントのサブネットマスクを指定します。サブネットマスクは、10進数の文字列で“255.255.255.0”のように指定します。

省略した場合は、ホスト情報定義ファイルから対象クライアントのサブネットマスクを読み出します。

/wt:*WaitTime*

電源投入確認までの待ち合わせ時間を指定します。WaitTimeに1～999までの整数を秒単位で指定してください。

本パラメタを省略した場合は、電源投入の待ち合わせを行いません。

本パラメタを指定すると、指定した待ち合わせ時間の経過後に電源投入確認が行われます。そのため、待ち合わせ時間より早く電源投入が完了していても、コマンドは指定された待ち合わせ時間を待ち合わせます。

待ち合わせ時間の経過後、電源投入が確認できた場合は正常終了、できなかった場合は異常終了します。

/rc:*nn*

クライアントの電源投入でエラーが発生した場合のリトライ回数を指定します。

nnには0～99の整数を指定してください。それ以外の値を設定した場合は、エラーになります。省略した場合は、リトライを行いません。

/ri:*nnnn*

エラーが発生した場合のリトライ間隔を指定します。本パラメタは、/rc:nnに1以上を指定した場合に有効となります。

nnnnには0～9999の整数を秒単位で指定してください。それ以外の値を設定した場合は、エラーになります。省略値は0です。

復帰値

復帰値	意 味
0	正常終了
0以外	異常終了

コマンド格納場所

Solaris	/opt/FJSVsnjss/bin
HP-UX	/opt/FHPsnjss/bin
AIX	/opt/FAIXsnjss/bin
Linux	/opt/FJSVsnjss/bin

記述例

クライアント“foo”に対して電源の投入を行います。

MACアドレス:

00-00-0e-9d-2c-38

IPアドレス:

133.161.222.110

サブネットマスク:

255.255.255.0

```
poncli foo /mac:00-00-0e-9d-2c-38 /ip:133.161.222.110 /snm:255.255.255.0
```

処理結果

コマンドの処理が終了すると、終了時の情報を標準出力に出力します。
出力される内容を以下に示します。

```
[処理開始日時] YYYY-MM-DD HH:MM:SS
[ホスト名] HostName
[MACアドレス] MACアドレス
[IPアドレス] IPアドレス
[サブネットマスク] サブネットマスク
[リトライ回数] リトライ回数
[発生したエラー] エラーメッセージ
[処理終了日時] YYYY-MM-DD HH:MM:SS
```

[発生したエラー]には、正常終了の場合は“なし”が出力されます。強制終了やエラーの場合はメッセージが出力されます。メッセージ内容については、“Systemwalker Operation Manager メッセージ集”を参照してください。

使用上の注意

- ホスト情報定義ファイルの情報を使用して本コマンドを実行する場合は、あらかじめホスト情報定義ファイルを定義しておいてください。詳細は、“Systemwalker Operation Manager 導入ガイド”を参照してください。
- 本コマンドにより電源投入が可能なクライアントは、Wakeup on LANに対応した機種に限ります。他のサブネット上のクライアントに電源を投入する場合は、サーバから電源投入対象端末のネットワークに対してブロードキャストパケットが到達できるようにネットワーク機器(ルータなど)の設定が必要です。詳しい設定方法はご使用のネットワーク機器のマニュアルを参照してください。

  ブロードキャストパケットが到達するかどうかは、コマンドを実行するSystemwalker Operation Managerのサーバ側のコマンドプロンプト画面でpingコマンドなどを投入して確認してください。応答がない場合はブロードキャストパケットは到達しません。

  投入先IPアドレスが、10.90.100.100で、サブネットマスクが255.255.255.0の場合の投入コマンドは、“ping 10.90.100.255”となります。
- 本コマンドは、対象クライアントの属するネットワークに対して、投入命令であるWakeup on LANパケットが通るとすぐに電源投入成功として復帰します。電源が投入されシステムが立ち上がるまでを確実に待ち合わせたい場合は、/wtオプションを指定してください。

10.2.2 poffcli 電源切断コマンド

記述形式

```
poffcli HostName [/sd:ShutdownMode] [/ef]
      [/rc:nn /ri:nnnn]
```

機能説明

Windowsからの電源切断に対応したクライアントの電源を切断します。離れた場所にあるクライアントの電源を自動的に切断することが可能となります。

パラメタ

HostName

電源を切断するクライアントのホスト名を指定します。

HostNameに、半角英数字15文字以内でホスト名を指定します。空白文字を含む場合はダブルクォーテーション「"」で囲んでください。

/sd:*ShutdownMode*

電源の切断モードを指定します。以下のいずれかを指定してください。

poff:

システムの電源を切断します。

reboot:

システムを再起動します。

down:

システムをシャットダウンします。

省略した場合は、poffが指定されたものとして、システムの電源を切断します。

/ef

電源切断の強制実行を指定します。

本指定により、起動中のアプリケーションを強制的に終了します。なお、アプリケーションが使用中のデータは保存されません。

クライアントのアクティブデスクトップは無効にしてください。アクティブデスクトップを有効にしている場合は、次回OS起動時に、アクティブデスクトップの修復機能が自動起動される場合があります。

/rc:*nn*

クライアントの電源切断でエラーが発生した場合のリトライ回数を指定します。

nnに0～99の整数を指定してください。それ以外の値を設定した場合は、エラーになります。省略した場合は、リトライを行いません。

/ri:*nnnn*

エラーが発生した場合のリトライ間隔を指定します。

nnnnに0～9999の整数を秒単位で指定してください。それ以外の値を設定した場合は、エラーになります。省略値は0です。

本パラメタは、/rc:nnに1以上が指定された場合に有効となります。

復帰値

復帰値	意 味
0	正常終了
0以外	異常終了

コマンド格納場所

Solaris	/opt/FJSVsnjss/bin
HP-UX	/opt/FHPsnjss/bin
AIX	/opt/FAIXsnjss/bin
Linux	/opt/FJSVsnjss/bin

記述例

クライアント“foo”に対して、電源の切断を行います。

```
poffcli foo /sd:poff
```

処理結果

コマンドの処理が終了すると、終了時の情報を標準出力に出力します。

出力される内容を以下に示します。

```
[処理開始日時] YYYY-MM-DD HH:MM:SS
[クライアントホスト名] HostName
[リトライ回数] リトライ回数
[発生したエラー] エラーメッセージ
[処理終了日時] YYYY-MM-DD HH:MM:SS
```

[発生したエラー]には、正常終了した場合は“なし”が出力されます。強制終了やエラーの場合はメッセージが出力されます。メッセージ内容については、“Systemwalker Operation Manager メッセージ集”を参照してください。

電源切断を行う前にエラーが発生した場合は、以下の形式でメッセージが出力されます。

```
[処理開始日時] YYYY-MM-DD HH:MM:SS
[コマンドエラー] エラーメッセージ
[処理終了日時] YYYY-MM-DD HH:MM:SS
```

メッセージ内容については、“Systemwalker Operation Manager メッセージ集”を参照してください。

使用上の注意

- 本コマンドを使用する場合は、クライアント側でクライアント業務連携を起動させておいてください。この時、電源切断可能なユーザ権限でクライアント業務連携が起動されている必要があります。
- 電源切断が可能なクライアントは、Windowsの電源切断機構に対応している機種に限ります。
- Windowsからの終了要求に対して、動作中のアプリケーションが終了しない場合は、クライアントの電源が切断できないことがあります。
- /efパラメタが指定されている場合、Windowsからの終了要求に対して応答しないアプリケーションはWindowsから強制的に終了されます。
- /rcパラメタおよび/riパラメタの指定が有効となるのは、Windows OSにおいて、対象クライアントのログインユーザがシャットダウン権限を持っていない場合です。

10.2.3 execacli アプリケーション起動コマンド

記述形式

execacli *HostName* [/am:*AppMode*] [/we:*WaitEnd*]
[/rc: *nn* /ri:*nnnn*]
/ae:*ExecuteAppli* [AppliParam]

機能説明

クライアントにインストールされている任意のアプリケーションを起動します。アプリケーションの終了を待ち合わせることができます。

パラメタ

HostName

アプリケーションを起動するクライアントのホスト名を指定します。

HostNameに、半角英数字15文字以内でホスト名を指定します。空白文字を含む場合はダブルクォーテーション「"」で囲んでください。

/am:*AppMode*

アプリケーションの起動モードを指定します。

以下のいずれかを指定してください。

normal:

通常のウィンドウの大きさで起動します。

icon:

ウィンドウが最小化された状態で起動します。

本パラメタを省略した場合は、normalとなり、通常のウィンドウの大きさで起動します。

/we:*WaitEnd*

起動したアプリケーションの終了までの待ち合わせ時間を指定します。WaitEndに1～9600の整数を秒単位で指定します。それ以外の値を指定するとエラーとなります。

本パラメタを省略した場合、アプリケーションの待ち合わせを行いません。

指定した時間内にアプリケーションが0で復帰した場合、コマンドは正常終了します。

指定した時間内にアプリケーションが0以外で復帰した場合、または指定した時間内にアプリケーションが終了しなかった場合、コマンドは異常終了します。

/rc:*nn*

アプリケーションの起動でエラーが発生した場合のリトライ回数を指定します。

nnに0～99の整数を指定してください。それ以外の値を設定した場合は、エラーとなります。省略した場合はリトライを行いません。

/ri:*nnnn*

エラーが発生した場合のリトライ間隔を指定します。本パラメタは、/rc:nnで1以上を指定した場合に有効となります。

nnnnに0～9999の整数を秒単位で指定してください。それ以外の値を設定した場合は、エラーとなります。省略値は0です。

/ae:*ExecuteAppli*

起動アプリケーション名をローカルドライブパスで指定します。ExecuteAppliに、半角511文字以内のアプリケーション名を指定します。起動アプリケーション名やアプリケーションに渡すパラメタに空白文字がある場合は、それぞれをダブルクォーテーション(")で囲んでください。

AppliParam

ExecuteAppliで指定したアプリケーションに対するパラメタを指定します。パラメタを複数指定する場合は、それぞれの間に空白文字を入れてください。各パラメタに空白文字を含む場合は、その両端をダブルクォーテーション「"」で囲んでください。

復帰値

復帰値	意 味
0	正常終了
0以外	異常終了(または、/WEオプション指定時に起動したアプリケーションが0以外で復帰した場合)

コマンド格納場所

Solaris	/opt/FJSVsnjss/bin
HP-UX	/opt/FHPsnjss/bin
AIX	/opt/FAIXsnjss/bin
Linux	/opt/FJSVsnjss/bin

記述例

クライアント“foo”に対して、アプリケーション“c:¥program files¥app.exe”を通常のウィンドウの大きさで起動し、60秒間終了を待ち合わせます。test.txtはapp.exeに対するパラメタです。

```
execacli foo /am:normal /we:60 /ae:"C:¥¥program files¥¥app.exe" test.txt
```

処理結果

コマンドの処理が終了すると、終了時の情報を標準出力に出力します。

出力される内容を以下に示します。

```
[処理開始日時] YYYY-MM-DD HH:MM:SS
[クライアントホスト名] HostName
[起動アプリケーション] 起動アプリケーション名
[リトライ回数] リトライ回数
[発生したエラー] エラーメッセージ
[処理終了日時] YYYY-MM-DD HH:MM:SS
```

[発生したエラー]には、正常終了の場合は“なし”が出力されます。強制終了やエラーの場合はメッセージが出力されます。メッセージ内容については、“Systemwalker Operation Manager メッセージ集”を参照してください。

アプリケーション起動を行う前にエラーが発生した場合は、以下の形式でメッセージが出力されます。

```
[処理開始日時] YYYY-MM-DD HH:MM:SS
[コマンドエラー] エラーメッセージ
[処理終了日時] YYYY-MM-DD HH:MM:SS
```

メッセージ内容については、“Systemwalker Operation Manager メッセージ集”を参照してください。

使用上の注意

- 本コマンドを使用する場合は、クライアント側でクライアント業務連携を起動させておいてください。
- /weオプションを指定した場合は、起動したアプリケーションの終了を待ち合わせますが、/weオプションを省略した場合は、起動したアプリケーションの完了を待ち合わせません。

- 本コマンドを強制終了した場合、起動したアプリケーションに対しての強制終了を行うことはできません。
- コマンドの強制終了は、アプリケーションの起動が完了した時点で行われます。このため、起動に時間がかかるアプリケーションの場合、コマンドの終了を待たされる場合があります。
- UNIXのシェルによっては、PC用のパス名の「¥」がエスケープシーケンスとなります。この場合は、「¥¥」と入力してください。

10.2.4 fcopycli ファイル転送コマンド

記述形式

```
fcopycli HostName /s:SourceName [/su:SourceUserName]
        /d:DestName [/du:DestUserName]
        [/lg:LoginGroup] [/a | /b] [/rc:nn /ri:nnnn]
```

機能説明

クライアントによるファイルの転送を行います。クライアント上のファイルをサーバに転送する、またはサーバ上のファイルをクライアントへ転送するコマンドです。Windowsのディレクトリ共有を利用したファイル転送(Windows共有方式)、またはFTPによるファイル転送(FTP方式)でファイルを転送します。

Systemwalker Operation Managerがインストールされていない転送先のサーバにも転送できます。

パラメタ

HostName

ファイルの転送を要求するクライアントのホスト名を指定します。HostNameにホスト名を半角英数字15文字以内で指定してください。

HostNameに空白文字を含む場合は、ダブルクォーテーション「"」で囲んでください。

/s:SourceName

転送元のファイルのパス名をファイル名も含めて指定します。ファイル名には、ワイルドカードも指定できます。

ワイルドカードは、シェルによって特別な意味を持つ文字(「*」など)があるため、必要に応じてSourceNameを一対の引用符(「'」または「"」)で囲む必要があります。

SourceNameに、URL形式のファイルのパス名を次のいずれかの方法で指定してください。パス名部分は255文字以内で指定し、空白文字を含む場合は、ダブルクォーテーション「"」で囲んでください。

ローカルドライブパス指定:

ドライブ名を含んだパスで指定します。

“file://”文字列に続いて、任意のファイルのパス名を指定します。

例) file://c:¥database¥database.db

UNC形式指定(Windows共有方式):

“¥¥コンピュータ名¥共有名”の形式でパスを指定します。コンピュータ名はWindowsのコンピュータ名を半角英数字15文字以内で指定してください。共有名は80バイト以内で指定してください。

“file://”文字列に続いて、任意のファイルのパス名を指定します。

例) file://¥¥foo¥share¥database¥database.db

FTPサーバ上のパス指定(FTP方式):

FTPサーバに対するパス名を指定します。

パス名は、フルパスまたはFTPでログイン時のカレントディレクトリからの相対パスで指定することができます。“ftp://”文字列に続いて、“サーバ名/任意のファイルのパス名”の形式で指定します。

例)フルパスで指定する場合：

ftp://foo.bar.com//usr/database/database.db

相対パスで指定する場合：

ftp://foo.bar.com/database/database.db

/su:*SourceUserName*

SourceNameで指定したファイルにアクセスするためのユーザ名を指定します。

SourceUserNameには、パスワード管理簿にあらかじめ設定したユーザ名を指定します。SourceUserNameが半角英数字20文字以内になるように指定してください。空白文字を含む場合は、ダブルクォーテーション「"」で囲んでください。

SourceNameの指定がローカルドライブパスまたはUNC形式指定の場合は、本指定は無視されます。この場合、クライアント業務連携が動作しているクライアントにログインしているユーザの権限でアクセスします。

FTPサーバ上のパス指定の場合は、必ず本指定を行ってください。

/d:*DestName*

転送先のファイルのパス名をファイル名も含めて指定します。ファイル名には、ワイルドカードで指定もできます。

ワイルドカードは、シェルによって特別な意味を持つ文字(「*」など)があるため、必要に応じてDestNameを一対の引用符(「'」または「"」)で囲む必要があります。

DestNameに、ファイルのパス名をURL形式で指定してください。指定方法は/s:SourceNameと同様です。/s:SourceNameを参照してください。

すでに存在するファイル名を指定した場合は、上書きされます。また、/sに指定するファイルのパス名(SourceName)と同じパス名を指定しないでください。

/du:*DestUserName*

DestNameで指定したファイルにアクセスするためのユーザ名を指定します。

DestUserNameには、パスワード管理簿にあらかじめ設定したユーザ名を指定します。DestUserNameが半角英数字20文字以内になるように指定してください。空白文字を含む場合は、ダブルクォーテーション「"」で囲んでください。

DestNameの指定がローカルドライブパスまたはUNC形式の場合は、本指定は無視されます。この場合、クライアント業務連携が動作しているクライアントにログインしているユーザの権限でアクセスします。

FTPサーバ上のパス指定の場合は、必ず本指定を行ってください。

/lg:*LoginGroup*

パスワード管理簿のログイン定義名を指定します。LoginGroupに、ログイン定義名を256文字以内で指定してください。空白文字を含む場合は、ダブルクォーテーション「"」で囲んでください。

/suまたは/duを指定した場合は必ず本指定を行ってください。

/a

ASCIIテキストモードでファイルを転送します。本パラメタはFTP方式の場合のみ指定できます。

/b

バイナリモードでファイルを転送します。本パラメタはFTP方式の場合のみ指定できます。

/a、/bともに指定されていない場合は、バイナリモードでファイルを転送します。

/rc:*nn*

ファイルの転送でエラーが発生した場合のリトライ回数を指定します。

nnに0～99の整数を指定してください。それ以外の値を設定した場合は、エラーになります。省略した場合は、リトライを行いません。

転送元ファイルおよび転送先ファイルにワイルドカードが指定された場合、失敗したファイルのみリトライを行います。

/ri:nnnn

エラーが発生した場合のリトライ間隔を指定します。本パラメタは、/rc:nnで1以上を指定した場合に有効となります。

nnnnに0～9999の整数を秒単位で指定してください。それ以外の値を設定した場合は、エラーになります。省略値は0です。

復帰値

復帰値	意 味
0	正常終了
0以外	異常終了

コマンド格納場所

Solaris	/opt/FJSVsnjss/bin
HP-UX	/opt/FHPsnjss/bin
AIX	/opt/FAIXsnjss/bin
Linux	/opt/FJSVsnjss/bin

記述例

クライアント“foo”に対して、クライアント上のローカルファイル“c:¥database¥master¥database.db”をローカルファイル“d:¥database¥database.db”へ転送します。転送に失敗した場合は、60秒間隔で最大3回リトライします。

```
fcopycli foo /s:file://c:¥¥database¥¥master¥¥database.db
        /d:file://d:¥¥database¥¥database.db /rc:3 /ri:60
```

処理結果

コマンドの処理が終了すると、終了時の情報を標準出力に出力します。

出力される内容を以下に示します。

```
[処理開始日時]  YYYY-MM-DD HH:MM:SS
[クライアントホスト名]  HostName
[サーバホスト／ユーザ名]  サーバホスト名／ユーザ名 (*)
[リトライ回数]  リトライ回数 (*)
[発生したエラー]  エラーメッセージ (*)
[ファイル転送]  転送元ファイル名 転送先ファイル名 (**)
[リトライ回数]  リトライ回数 (**)
[発生したエラー]  エラーメッセージ (**)
[処理終了日時]  YYYY-MM-DD HH:MM:SS
```

(*):

/suおよび/duを指定した場合に出力される情報です。

(**):

ワイルドカードが指定された場合、それぞれのファイルについての情報も出力されます。

[発生したエラー]には、正常終了の場合は“なし”が出力されます。強制終了やエラーの場合はメッセージが出力されます。メッセージ内容については、“Systemwalker Operation Manager メッセージ集”を参照してください。

ファイルの転送を行う前にエラーが発生した場合は、以下の形式でメッセージが出力されます。

```
[処理開始日時]　YYYY-MM-DD HH:MM:SS
[コマンドエラー]　エラーメッセージ
[処理終了日時]　YYYY-MM-DD HH:MM:SS
```

メッセージ内容については、“Systemwalker Operation Manager メッセージ集”を参照してください。

使用上の注意

- 本コマンドを使用する場合は、クライアント側でクライアント業務連携を動作させておいてください。
- 事前にパスワード管理コマンドを使って、パスワード管理簿の設定を行ってください。詳細は“10.3.1 lgmanage パスワード管理コマンド”を参照してください。
- FTPサーバ上のパス指定でファイルを転送する場合、転送元および転送先の両方をFTPサーバ上のパス指定で転送することはできません。転送元と転送先に指定できる組み合わせは以下のとおりです。いずれかの組み合わせで指定してください。

	転送元ファイル	転送先ファイル
1	ローカルパス形式	ローカルパス形式
2	ローカルパス形式	UNC形式
3	ローカルパス形式	FTPサーバパス形式
4	UNC形式	ローカルパス形式
5	UNC形式	UNC形式
6	UNC形式	FTPサーバパス形式
7	FTPサーバパス形式	ローカルパス形式
8	FTPサーバパス形式	UNC形式

- FTPサーバ上のパス指定でファイル転送ができるFTPサーバおよびファイルは、“dir”コマンドの結果が以下の形式(パーミッション、リンク数、所有者、グループ、サイズ、日付、時刻、ファイル名の順に英数字表記)で表示されるサーバおよびファイルに限ります。日本語を含むファイル名については、“dir”コマンドの結果でファイル名が正しく表示されるファイルを扱うことができます。ファイル名が文字化けして表示されるファイルは扱うことができません。

```
-rw-rw-r--  1 xxxxx  xxxxxx     605877 Dec 19 18:44 xxxxx.xxx
-rw-rw-r--  1 xxxxx  xxxxxx     560661 Dec 19 18:59 xxxxx.xxx
drwx------  4 xxxxx  xxxxx          96 Jan 20  1998 xxxxx
-rw-rw-r--  1 xxxxx  xxxxx        6042 Jan 25 13:38 xxxxxxxxxx
-rw-rw-r--  1 xxxxx  xxxxx        5433 Feb 3  17:11 xxxxx.xxx
-rw-rw-r--  1 xxxxx  xxxxx        2155 Feb 3  17:35 xxxxx.xxx
-rw-rw-r--  1 xxxxx  xxxxx        3090 Feb 1  20:13 xxxxx.xxx
```

FTPサーバ上のパス指定が可能であるかどうかの確認方法は、“10.1.2 fcopysvr ファイル転送コマンド”を参照してください。

日付・時刻表示が日本語の場合、OSやFTPサーバのマニュアルを参照し英語表示となるように設定してください。

なお、出荷状態の日本語Solaris 2.5 オペレーティングシステム以上に付属のFTPサーバでは、ディレクトリの日付・時刻が日本語で表示されます。上記の形式で表示するためには、/etc/default/initファイルに以下の記述を追加してください。

```
LC_TIME=C
```

- コマンドの強制終了は、1つのファイルの転送が完了した時点で行われます。このため、大きなファイルを転送中に強制終了を行うと、コマンドの終了を待たされる場合があります。
- UNIXのシェルによっては、PC用のパス名の「¥」がエスケープシーケンスとなります。この場合は、「¥¥」と入力してください。
- 本コマンドの/sパラメタに指定したファイル名に、ワイルドカードを指定した場合は、次の注意が必要です。

  同じクライアントに対して、同時に実行するクライアント業務連携コマンドで処理するファイル数を含めて、合計500ファイル以下となるようにしてください。
- パラメタに指定するファイル名にワイルドカードを使用する場合は、ファイル名全体をワイルドカード(アスタリスクのみ)とすることをお勧めします。ファイル名の一部をワイルドカードにする場合は、転送したファイルが想定するファイル名となるかどうかを確認のうえ利用してください。
- 1ファイルあたりのサイズは、2GB以下としてください。

10.2.5 fctrlcli ファイル操作コマンド

記述形式

```
fctrlcli HostName {/delete | /rename | /mkdir | /rmdir}
        /f:FilePathName [/fn:FileName] [/un:UserName]
        [/lg:LoginGroup] [/rc:nn /ri:nnnn]
```

機能説明

クライアント上のファイルまたはディレクトリを操作するコマンドです。ファイル操作では、ファイルの削除およびファイル名を変更することができます。ディレクトリ操作では、ディレクトリを作成および削除することができます。Windowsのディレクトリ共有を利用したファイル操作(Windows共有方式)、またはFTPによるファイル操作(FTP方式)でファイルを操作します。

Systemwalker Operation Managerがインストールされていない操作先サーバを操作することもできます。

パラメタ

HostName

ファイル操作元のクライアントのホスト名を指定します。HostNameに、半角英数字15文字以内でホスト名を指定してください。空白文字を含む場合は、ダブルクォーテーション「"」で囲んでください。

/delete

ファイルを削除します。

/rename

ファイル名を変更します。

/mkdir

ディレクトリを作成します。

/rmdir

ディレクトリを削除します。

/f:FilePathName

対象のファイルまたはディレクトリのパス名を指定します。/deleteまたは/renameの指定の場合は、ファイル名には、ワイルドカードも指定できます。

ワイルドカードは、シェルによって特別な意味を持つ文字(「*」など)があるため、必要に応じてFilePathNameを一対の引用符(「'」または「"」)で囲む必要があります。

FilePathNameに、URL形式のファイルのパス名を次のいずれかの方法で指定してください。パス名は255文字以内で指定し、空白文字を含む場合はダブルクォーテーション「"」で囲んでください。

ローカルドライブパス指定:

ドライブ名を含んだパス名を指定します。

“file://”文字列に続いて、任意のファイルのパス名を指定します。

例) file://c:¥database¥database.db

UNC形式指定(Windows共有方式):

“¥¥コンピュータ名¥共有名”の形式でパスを指定します。コンピュータ名はWindowsのコンピュータ名を半角英数字15文字以内で指定してください。共有名は80バイト以内で指定してください。

“file://”文字列に続いて、任意のファイルのパス名を指定します。

例) file://¥¥foo¥share¥database¥database.db

FTPサーバ上のパス指定(FTP方式):

FTPサーバに対するパス名を指定します。

パス名は、フルパスまたはFTPでログイン時のカレントディレクトリからの相対パスで指定することができます。“ftp://”文字列に続いて、“サーバ名/任意のファイルのパス名”の形式で指定します。

例)フルパスで指定する場合：

ftp://foo.bar.com//usr/database/database.db

相対パスで指定する場合：

ftp://foo.bar.com/database/database.db

/fn:*FileName*

変更先のファイル名を指定します。ワイルドカードも指定できます。

ワイルドカードは、シェルによって特別な意味を持つ文字(「*」など)があるため、必要に応じてFileNameを一対の引用符(「'」または「"」)で囲む必要があります。

FileNameに、ファイル名を255文字以内で指定します。空白文字を含む場合はダブルクォーテーション「"」で囲んでください。

本パラメタは、/renameを指定した場合のみ有効です。

/un:*UserName*

FilePathNameで指定したファイルまたはディレクトリにアクセスするユーザ名を指定します。

UserNameには、パスワード管理簿にあらかじめ設定したユーザ名を指定します。UserNameが半角英数字20文字以内になるように指定してください。空白文字を含む場合は、ダブルクォーテーション「"」で囲んでください。

FilePathNameがローカルドライブパスまたはUNC形式で指定した場合は本指定は無視されます。この場合、クライアント業務連携が動作しているクライアントにログインしているユーザ権限でアクセスします。FTPサーバ上のパスで指定した場合は、必ず本指定を行ってください。

/lg:*LoginGroup*

パスワード管理簿のログイン定義名を指定します。LoginGroupに、ログイン定義名を256文字以内で指定します。空白文字を含む場合は、ダブルクォーテーション「"」で囲んでください。

/unの指定をした場合は必ず本指定を行ってください。

/rc:*nn*

ファイルの操作でエラーが発生した場合のリトライ回数を指定します。

nnに0～99の整数を指定してください。それ以外の値を設定した場合は、エラーになります。省略した場合は、リトライを行いません。

/ri:nnnn

エラーが発生した場合のリトライ間隔を指定します。本指定は、/rc:nnで1以上を指定した場合のみ有効となります。

nnnnに0～9999の整数を秒単位で指定してください。それ以外の値を設定した場合は、エラーになります。省略値は0です。

復帰値

復帰値	意 味
0	正常終了
0以外	異常終了

コマンド格納場所

Solaris	/opt/FJSVsnjss/bin
HP-UX	/opt/FHPsnjss/bin
AIX	/opt/FAIXsnjss/bin
Linux	/opt/FJSVsnjss/bin

記述例

クライアント“foo”に対して、クライアント上のローカルファイル“c:¥database¥master¥database.db”を削除します。削除に失敗した場合は、60秒間隔で最大3回リトライします。

```
fctrlcli foo /delete /f:file://c:¥¥database¥¥master¥¥database.db
/rc:3 /ri:60
```

処理結果

コマンドの処理が終了すると、終了時の情報を標準出力に出力します。

出力される内容を以下に示します。

```
[処理開始日時] YYYY-MM-DD HH:MM:SS
[クライアントホスト名] HostName
[サーバホスト／ユーザ名] サーバホスト名／ユーザ名 (***)
[リトライ回数] リトライ回数 (***)
[発生したエラー] エラーメッセージ (***)
[ファイル操作種別] 操作ファイル／ディレクトリ名 変更後ファイル名 (*)(**)
[リトライ回数] リトライ回数 (*)
[発生したエラー] エラーメッセージ (*)
[処理終了日時] YYYY-MM-DD HH:MM:SS
```

(*):

ワイルドカードが指定された場合、それぞれのファイルについての情報も出力されます。

(**):

変更後ファイル名は、/renameが指定された場合のみ出力されます。

(***):

/unを指定した場合に出力される情報です。

[発生したエラー]には、正常終了の場合は“なし”が出力されます。強制終了やエラーの場合はメッセージが出力されます。メッセージ内容については、“Systemwalker Operation Manager メッセージ集”を参照してください。

[ファイル操作種別]は、“ファイル削除”“ファイル名変更”“ディレクトリ作成”“ディレクトリ削除”のいずれかが出力されます。

ファイル操作を行う前にエラーが発生した場合は、以下の形式でメッセージが出力されます。

```
[処理開始日時]　YYYY-MM-DD HH:MM:SS
[コマンドエラー]　エラーメッセージ
[処理終了日時]　YYYY-MM-DD HH:MM:SS
```

メッセージ内容については、“Systemwalker Operation Manager メッセージ集”を参照してください。

使用上の注意

- 本コマンドを使用する場合は、クライアント側でクライアント業務連携を起動させておいてください。
- 事前にパスワード管理コマンドを使って、パスワード管理簿の設定を行ってください。詳細は“10.3.1 lgmanage パスワード管理コマンド”を参照してください。
- FTPサーバ上のパス指定でファイルを操作できるFTPサーバおよびファイルは、“dir”コマンドの結果が以下の形式(パーミッション、リンク数、所有者、グループ、サイズ、日付、時刻、ファイル名の順に英数字表記)で表示されるサーバおよびファイルに限ります。日本語を含むファイル名については、“dir”コマンドの結果でファイル名が正しく表示されるファイルを扱うことができます。ファイル名が文字化けして表示されるファイルは扱うことができません。

  ```
  -rw-rw-r--  1 xxxxx  xxxxxx     605877 Dec 19 18:44 xxxxx.xxx
  -rw-rw-r--  1 xxxxx  xxxxxx     560661 Dec 19 18:59 xxxxx.xxx
  drwx------  4 xxxxx  xxxxx          96 Jan 20  1998 xxxxx
  -rw-rw-r--  1 xxxxx  xxxxx        6042 Jan 25 13:38 xxxxxxxxxxx
  -rw-rw-r--  1 xxxxx  xxxxx        5433 Feb 3  17:11 xxxxx.xxx
  -rw-rw-r--  1 xxxxx  xxxxx        2155 Feb 3  17:35 xxxxx.xxx
  -rw-rw-r--  1 xxxxx  xxxxx        3090 Feb 1  20:13 xxxxx.xxx
  ```

  FTPサーバ上のパス指定が可能であるかどうかの確認方法は、“10.1.2 fcopysvr ファイル転送コマンド”を参照してください。

  日付・時刻表示が日本語の場合、OSやFTPサーバのマニュアルを参照し英語表示となるように設定してください。

  なお、出荷状態の日本語Solaris 2.5 オペレーティングシステム以上に付属のFTPサーバでは、ディレクトリの日付・時刻が日本語で表示されます。上記の形式で表示するためには、/etc/default/initファイルに以下の記述を追加してください。

  ```
  LC_TIME=C
  ```

- UNIXのシェルによっては、PC用のパス名の「¥」がエスケープシーケンスとなります。この場合は、「¥¥」と入力してください。
- 本コマンドの/fパラメタに指定したファイル名に、ワイルドカードを指定した場合は、次の注意が必要です。

  同じクライアントに対して、同時に実行するクライアント業務連携コマンドで処理するファイル数を含めて、合計500ファイル以下となるようにしてください。
- パラメタに指定するファイル名にワイルドカードを使用する場合は、ファイル名全体をワイルドカード(アスタリスクのみ)とすることをお勧めします。ファイル名の一部をワイルドカードにする場合は、操作したファイルが想定するファイル名となるかどうかを確認のうえ利用してください。
- 1ファイルあたりのサイズは、2GB以下としてください。

10.2.6　fprscli ファイル圧縮/伸長コマンド

記述形式

```
fprscli HostName {/p | /e} /s:SourceName /d:DestName
      [/rc:nn /ri:nnnn]
```

機能説明

クライアントでファイルの圧縮および伸長を行います。

転送元でファイルを圧縮して、転送先でファイルを伸長するといった使い方ができ、ファイルの転送時間が短縮され、ネットワーク上の負荷を軽減できます。

パラメタ

HostName

ファイルの圧縮/伸長元クライアントのホスト名を指定します。HostNameに、半角英数字15文字以内でホスト名を指定します。空白文字を含む場合は、ダブルクォーテーション「"」で囲んでください。

/p

ファイルを圧縮します。

/e

ファイルを伸長します。

/s:*SourceName*

圧縮/伸長元ファイルのパス名をローカルドライブパスで指定します。圧縮元ファイルの場合は、ファイル名には、ワイルドカードも指定できます。

ワイルドカードは、シェルによって特別な意味を持つ文字(「*」など)があるため、必要に応じてSourceNameを一対の引用符(「'」または「"」)で囲む必要があります。

SourceNameに、255文字以内でパス名を指定します。空白文字を含む場合は、ダブルクォーテーション「"」で囲んでください。

/d:*DestName*

圧縮/伸長先のファイルのパス名をローカルドライブパスで指定します。伸長先ファイルの場合は、ファイル名には、ワイルドカードで指定もできます。

ワイルドカードは、シェルによって特別な意味を持つ文字(「*」など)があるため、必要に応じてDestNameを一対の引用符(「'」または「"」)で囲む必要があります。

DestNameに、255文字以内でパス名を指定します。空白文字を含む場合は、ダブルクォーテーション「"」で囲んでください。

すでに存在するファイル名を指定した場合は、上書きされます。また、/sに指定するファイルのパス名(SourceName)と同じパス名を指定しないでください。

/rc:*nn*

ファイルの圧縮/伸長でエラーが発生した場合のリトライ回数を指定します。

nnに0～99の整数を指定してください。それ以外の値を設定した場合は、エラーになります。省略した場合は、リトライを行いません。

圧縮元または伸長先ファイルにワイルドカードを指定した場合は、失敗したファイルのみリトライを行います。

/ri:*nnnn*

エラーが発生した場合のリトライ間隔を指定します。

nnnnに0～9999の整数を秒単位で指定してください。それ以外の値を設定した場合は、エラーになります。省略値は0です。

本パラメタは、/RCパラメタで1以上を指定した場合に有効となります。

復帰値

復帰値	意 味
0	正常終了
0以外	異常終了

コマンド格納場所

Solaris	/opt/FJSVsnjss/bin
HP-UX	/opt/FHPsnjss/bin
AIX	/opt/FAIXsnjss/bin
Linux	/opt/FJSVsnjss/bin

記述例

クライアント“foo”に対して、クライアント上のローカルファイル“c:¥database¥master¥*.*”を“d:¥database¥database.prs”に圧縮します。圧縮に失敗した場合は、60秒間隔で最大3回リトライします。

```
fprscli foo /p /s:c:¥¥database¥¥master¥¥*.* /d:d:¥¥database¥
¥database.prs
        /rc:3 /ri:60
```

処理結果

コマンドの処理が終了すると、終了時の情報を標準出力に出力します。

出力される内容を以下に示します。

```
[処理開始日時]　YYYY-MM-DD HH:MM:SS
[クライアントホスト名]　HostName
[ファイル圧縮／伸長]　圧縮／伸長元ファイル名 圧縮／伸長先ファイル名 (*)
[リトライ回数]　リトライ回数 (*)
[発生したエラー]　エラーメッセージ (*)
[処理終了日時]　YYYY-MM-DD HH:MM:SS
```

(*):

　ワイルドカードが指定された場合、それぞれのファイルについての情報も出力されます。

[発生したエラー]には、正常終了の場合は“なし”が出力されます。強制終了やエラーの場合はメッセージが出力されます。メッセージ内容については“Systemwalker Operation Manager メッセージ集”を参照してください。

圧縮/伸長処理を行う前にエラーが発生した場合は、以下の形式でメッセージが出力されます。

```
[処理開始日時]　YYYY-MM-DD HH:MM:SS
[コマンドエラー]　エラーメッセージ
[処理終了日時]　YYYY-MM-DD HH:MM:SS
```

メッセージ内容については、“Systemwalker Operation Manager メッセージ集”を参照してください。

使用上の注意

- 本コマンドを使用する場合は、クライアント側でクライアント業務連携を起動させておいてください。

- UNIXのシェルによっては、PC用のパス名の「¥」がエスケープシーケンスとなります。この場合は、「¥¥」と入力してください。
- 本バージョンレベル以降のSystemwalker Operation ManagerのUNIXサーバで/wオプションを付けずに作成された圧縮ファイルを、クライアントで伸長することはできません。
- 本コマンドで作成した圧縮ファイルは、UNIXサーバで/wオプションを付けて作成したファイルの形式と同じWindowsプラットフォーム形式となります。
- サーバ圧縮/伸長コマンドで/wオプションを付けて圧縮されたファイルを本コマンドを使ってWindows上で伸長した場合、次のように展開されます。
  - 圧縮元で書込みが誰にも許可されていない場合は、Windows上では読み取り専用属性になります。
  - 圧縮元で書込みがowner/group/otherのいずれかに許されていた場合は、Windows上では読み書き可能な属性になります。
- 本コマンドで作成したWindowsプラットフォーム形式の圧縮ファイルをUNIX上で伸長した場合、次のように展開されます。

圧縮前のファイル属性	展開後のファイルの情報
読み取り専用	所有者／グループ情報は、コマンドを実行したユーザの情報となる。 モードはすべてのユーザが読込み可能な0444となる。(注1)
読み取り専用 以外	所有者／グループ情報は、コマンドを実行したユーザの情報となる。 モードはすべてのユーザが読み書き可能な0666となる。(注2)

注1)

すべてのユーザから読込み可能な状態で、ls -lコマンドで-r--r--r--と表示されるファイルのパーミッションをさします。

注2)

すべてのユーザから読み書き可能な状態で、ls -lコマンドで-rw-rw-rw-と表示されるファイルのパーミッションをさします。

- 圧縮するユーザに、読み取り可能のパーミッションが設定されていない場合は圧縮することはできません。ただし、スーパーユーザは圧縮可能です。
- コマンドの強制終了は、1つのファイル圧縮/伸長が完了した時点で行われます。このため、大きなファイルを圧縮中に強制終了を行うと、コマンドの終了を待たされる場合があります。
- 本コマンドの/sパラメタに指定したファイル名に、ワイルドカードを指定した場合は、次の注意が必要です。

  同じクライアントに対して、同時に実行するクライアント業務連携コマンドで処理するファイル数を含めて、合計500ファイル以下となるようにしてください。
- パラメタに指定するファイル名にワイルドカードを使用する場合は、ファイル名全体をワイルドカード(アスタリスクのみ)とすることをお勧めします。ファイル名の一部をワイルドカードにする場合は、圧縮/伸長したファイルが想定するファイル名となるかどうかを確認のうえ利用してください。
- 本コマンドの実行時には、クライアント上の以下の場所に、一時ファイルが作成されます。
  - 環境変数TEMPに指定されているディレクトリ

  TEMP領域が不足した場合、圧縮・伸長することはできません。一時ファイルのサイズは、一度に圧縮・伸長するファイルサイズの合計となります。
- 1ファイルあたりのサイズは、2GB以下としてください。

10.3　業務連携管理コマンド

業務連携管理コマンドについて説明します。

10.3.1　lgmanage パスワード管理コマンド

記述形式

```
lgmanage add /lg:LoginGroup /un:UserName /pw:Password
lgmanage del /lg:LoginGroup /un:UserName
lgmanage mod /lg:LoginGroup /un:UserName /pw:Password
lgmanage ref /lg:LoginGroup
lgmanage {help | ?}
lgmanage
```

機能説明

業務連携で使用するパスワード管理簿を管理するコマンドです。パスワード管理簿は、業務連携コマンドで指定するユーザとパスワードをログイン定義名ごとにファイルとして管理するものです。

パラメタ

add/del/mod/ref

パスワード管理簿への操作種別を指定します。

add:

パスワード管理簿へユーザ名およびパスワードを追加します。

del:

パスワード管理簿からユーザ名およびパスワードを削除します。

mod:

パスワード管理簿のユーザ名に対応したパスワードを変更します。

ref:

パスワード管理簿に登録されているユーザ名の一覧を参照します。

/lg:LoginGroup

パスワード管理簿のログイン定義名を指定します。

業務連携コマンドを実行するユーザが所属するOSのグループ名をLoginGroupに指定します。ログイン定義名は英大文字と英小文字が区別されます。LoginGroupに200文字以内で指定してください。

/un:UserName

パスワード管理簿でパスワードを管理するユーザ名を指定します。UserNameで指定したユーザ名が、業務連携コマンドで指定した連携先のサーバ(FTPサーバなど)のログインユーザになります。UserNameに、20文字以内で指定します。サーバ名・ドメイン名を含む形式の場合は、サーバ名またはドメイン名で15文字、ユーザ名で20文字以内となります。ユーザ名は英大文字と英小文字は区別され、日本語は使用できません。

/pw:Password

パスワード文字列を指定します。Passwordに指定したパスワード文字列が、業務連携コマンドで指定した連携先のサーバ(FTPサーバなど)のログインパスワードになります。Passwordに1文字以上50文字以下の半角英数字を指定します。変更の場合は新しいパスワードを指定します。

help|?

“help”または“?”で起動した場合、本コマンドの使用方法を表示します。

パラメタ指定なし

対話型モードによりパラメタ入力を行います。

入力項目が表示され、すべての項目を対話型メニューで入力します。

復帰値

復帰値	意味
0	正常終了
0以外	異常終了。エラーの内容はエラーメッセージを確認してください。

コマンド格納場所

Solaris	/opt/FJSVsnjss/bin
HP-UX	/opt/FHPsnjss/bin
AIX	/opt/FAIXsnjss/bin
Linux	/opt/FJSVsnjss/bin

使用上の注意

- パスワード管理簿ファイルの格納先については、“Systemwalker Operation Manager 導入ガイド”の、“パスワード管理簿の定義”を参照してください。
- ユーザ名およびパスワードを追加する時に、指定したログイン定義名のパスワード管理簿ファイルが存在しない場合は、パスワード管理簿ファイルを作成します。作成したパスワード管理簿に、ユーザ名およびパスワードを追加します。また、パスワード管理簿ファイルに同一ユーザ名が存在する場合は、ユーザ名を追加せずにエラーとなります。
- パスワード管理簿は、業務連携機能を使用するサーバごとに作成し、管理してください。
- パスワード管理簿ファイルは、ジョブスケジューラのスケジュール情報の一部として、ポリシー情報の配付の対象となります。
- UNIX版では、ユーザ名、グループ名(ログイン定義名)には日本語は使用できません。半角英数字と「-」および「_」のみ使用できます。
- パラメタ指定によりユーザ名/パスワードを登録する場合、以下の状態では、パスワードがシェルの履歴ファイルに残り、他のユーザから読み取られる可能性があります。

  － シェルの実行コマンドの履歴機能を使用している場合

  － 履歴ファイルのアクセス権に、他のユーザの読み取り権限が付いている場合

  ご使用のシェルの仕様が不明な場合は、lgmanageコマンドをパラメタ指定なしで起動し、対話形式でユーザ名/パスワードを登録してください。

第11章　ユーティリティコマンド【Windows版】

本章では、ユーティリティコマンドについて説明します。

注意

WindowsのUAC機能を有効にしている環境でコマンドを実行する場合、OSにログインしたユーザの権限によって権限の昇格を求めるダイアログが表示され、対処が必要となります。

詳細は、“Systemwalker Operation Manager 運用ガイド”の、“Windowsでコマンドを実行する場合の注意事項【Windows版】”を参照してください。

11.1　chksrv サービス状態確認コマンド

サービスの状態を確認するコマンドです。サービスを起動/停止させるバッチファイルの中などで使用します。

記述形式

```
chksrv サービス名 {start|stop}
```

オペランド

サービス名

状態を確認するサービス名を指定します。

start

サービスが開始しているかを確認します。

stop

サービスが停止しているかを確認します。

復帰値

復帰値	意味
0	“start”オペランドを指定した場合は、サービスが開始していることを示します。“stop”オペランドを指定した場合は、サービスが停止していることを示します。
10	“start”オペランドを指定した場合は、サービスが開始していないことを示します。“stop”オペランドを指定した場合は、サービスが停止していないことを示します。
20	サービスが起動処理中であることを示します。
30	サービスが停止処理中であることを示します。
40	サービスが再開処理中であることを示します。
50	サービスが一時停止処理中であることを示します。
60	サービスが一時停止していることを示します。
-1	その他のエラーが発生したことを示します。

標準出力

chksrvコマンドを起動すると、以下の文字列が標準出力(stdout)に出力されます。

サービスが開始している場合：	SERVICE_RUNNING
サービスが停止している場合：	SERVICE_STOPPED
サービスが起動処理中の場合：	SERVICE_START_PENDING
サービスが停止処理中の場合：	SERVICE_STOP_PENDING
サービスが再開処理中の場合：	SERVICE_CONTINUE_PENDING
サービスが一時停止処理中の場合：	SERVICE_PAUSE_PENDING
サービスが一時停止している場合：	SERVICE_PAUSED

実行に必要な権限

システム管理者(Administratorsグループに属するユーザ)権限が必要です。

使用例

バッチファイルでの使用例を以下に示します。

サービスを開始させるバッチファイルの例

ジョブの正常終了コード上限値を0とした場合、以下のバッチファイルは、サービスが開始された時(終了コードが0)は正常終了となり、サービスが開始されなかった時(終了コードが10)は異常終了となります。

```
net start サービス名
chksrv サービス名 start
```

サービスを停止させるバッチファイルの例

ジョブの正常終了コード上限値を0とした場合、以下のバッチファイルは、サービスが停止した時(終了コードが0)は正常終了となり、サービスが停止しなかった時(終了コードが10)は異常終了となります。

```
net stop サービス名
chksrv サービス名 stop
```

11.2 f3crhsst サービス開始コマンド

サービス制御マネージャに、サービスの開始を指示するコマンドです。

記述形式

```
f3crhsst $Serv:サービス名
```

オペランド

サービス名

開始するサービス名を指定します。

復帰値

復帰値	意味
0	正常に指示されたことを示します。
3	すでに開始されたことを示します。
7	サービスの操作に失敗したことを示します。
8	指定されたサービス名は存在しないことを示します。

復帰値	意味
9	指定されたサービスが無効であるか、または有効なデバイスが関連づけられていないため、開始できないことを示します。
10	パラメタに誤りがあることを示します。
20	システムエラーが発生したことを示します。

実行に必要な権限

システム管理者(Administratorsグループに属するユーザ)権限が必要です。

使用例

ジョブスケジューラサービスを起動する例を以下に示します。

```
f3crhsst $Serv:Systemwalker Mpjobsch
```

11.3 f3crhssp サービス停止コマンド

サービス制御マネージャに、サービスの停止を指示するコマンドです。

記述形式

```
f3crhssp $Serv:サービスの表示名
```

オペランド

サービス名

停止するサービスの表示名を指定します。

復帰値

復帰値	意味
0	正常に指示されたことを示します。
6	別の実行中のサービスが依存関係にあることを示します。
7	すでに停止されたことを示します。
8	サービスの操作に失敗したことを示します。
9	指定されたサービスの表示名は存在しないことを示します。
10	パラメタに誤りがあることを示します。
20	システムエラーが発生したことを示します。

実行に必要な権限

システム管理者(Administratorsグループに属するユーザ)権限が必要です。

使用例

ジョブスケジューラサービスを停止する例を以下に示します。

```
f3crhssp $Serv:Systemwalker Mpjobsch
```

第12章　SSL環境設定コマンド

12.1　ahsregistupin パスフレーズ登録コマンド

記述形式

```
ahsregistupin -f upinfile
```

機能説明

パスフレーズを暗号化してパスフレーズ管理ファイルに登録します。

オプション

-f upinfile

暗号化したパスフレーズを格納するパスフレーズ管理ファイルを指定します。

コマンド格納場所

Windows	Systemwalker Operation Managerインストール先ディレクトリ¥mpwalker.jm¥mpahs¥bin
Solaris Linux	/opt/FJSVftlo/mpahs/bin

実行に必要な権限

システム管理者(Administratorsグループに属するユーザ/スーパーユーザ)権限が必要です。

注意事項

- パスフレーズには、秘密鍵管理環境を作成したときのパスフレーズと同じ文字列を指定してください。
- パスフレーズ管理ファイルは、上書き保存されます。
- 本コマンドは、ネットワーク上でのパスフレーズの漏洩を防ぐため、コンソールで実行するか、または外部から接続されないLANで接続された端末で実行することを推奨します。
- 本コマンドの実行後、セキュリティ対策のため、以下の手順でパスフレーズ管理ファイルのアクセス権限を変更することを推奨します。【Windows版】
  1. エクスプローラを起動します。
  2. パスフレーズ管理ファイルを選択して右クリックし、[プロパティ]をクリックします。
  3. [セキュリティ]タブをクリックします。
  4. プロパティの[セキュリティ]画面で、以下のグループ以外の[アクセス許可]設定に“拒否”を選択して、[OK]ボタンをクリックします。
     - Administrators
     - SYSTEM

使用例

パスフレーズの登録手順は、以下のとおりです。

1. ahsregistupinコマンドを起動します。

2. パスフレーズの問い合せがあります。

   パスフレーズとして、128バイト以内の半角英数字の文字列を入力します。このとき、入力文字列は表示されません。

3. 再度、パスフレーズの問い合せがあります。

   2.と同じパスフレーズを入力します。このとき、入力文字列は表示されません。

【Windows版】

パスフレーズ(対話入力)を暗号化して、パスフレーズ管理ファイル“d:¥mpahso¥sslcert¥upinfile”に登録する場合

```
C:¥Systemwalker¥MPWALKER.JM¥mpahs¥bin¥ahsregistupin.exe -f d:¥mpahso¥sslcert¥upinfile
  UserPIN: (注)
  Re-type UserPIN: (注)
```

注) 本文字列が表示された場合は、パスフレーズを入力してください。なお、入力される文字はエコーバックされません。

【UNIX版】

パスフレーズ(対話入力)を暗号化して、パスフレーズ管理ファイル“/export/home/mpahso/sslcert/upinfile”に登録する場合

```
/opt/FJSVftlo/mpahs/bin/ahsregistupin -f /export/home/mpahso/sslcert/upinfile
  UserPIN: (注)
  Re-type UserPIN: (注)
```

注) 本文字列が表示された場合は、パスフレーズを入力してください。なお、入力される文字はエコーバックされません。

12.2 openssl RSA秘密鍵の作成/CSRの作成/証明書のフィンガープリント表示/DHパラメータファイル作成/ECDSA秘密鍵の作成コマンド

記述形式

(1) RSA秘密鍵の作成

```
openssl genrsa [-aes256] [-out filename] [numbits]
```

(2) CSR(証明書取得申請書)の作成

```
openssl req -new [-key filename] [-subj args] -config configfilename
```

(3) 証明書のフィンガープリント表示

```
openssl x509 [-sha1|-sha256|-md5] -fingerprint -noout [-in file]
```

(4) DHパラメータファイル作成

```
openssl dhparam [-outform DER | PEM] [-out filename] numbits
```

(5) ECDSA秘密鍵の作成

```
openssl genpkey [-aes256] [-out filename] -algorithm EC -pkeyopt
opt:value
```

機能説明

本コマンドでは、RSA秘密鍵の作成、CSR(証明書取得申請書)の作成、証明書のフィンガープリントの表示、DHパラメータファイルの作成、ECDSA秘密鍵の作成をします。

オプション

genrsaサブコマンド

-aes256

aes256で暗号化します。本オプションを省略した場合は、暗号化されません。

-out filename

出力ファイル名を指定します。本オプションを省略した場合は、標準出力に出力されます。

numbits

生成する秘密鍵のbitサイズを指定します。これは最後に指定する必要があります。デフォルトは2048で、512未満の値は許可されていません。

reqサブコマンド

-new

新しくCSR(証明書取得申請書)を生成します。

-key filename

入力する秘密鍵ファイル名を指定します。

-subj args

証明書に記載されるサーバ情報を設定します。指定する内容は、/ で区切り、以下の書式で指定します。

/属性１=値１/属性２=値２/属性３=...

属性と値は以下を指定します。

C : 2文字の国コード 日本は“JP”

ST : 都道府県

L : 市区町村

O : 組織名

OU : 部門名

CN : サーバの FQDN

本オプションを省略した場合は、コマンド実行時に以下のように入力を求められます。斜体が入力箇所です。

```
# /opt/FJSVftlo/mpahs/oss/openssl/bin/openssl req -new -key server.key -config /opt/FJSVftlo/mpahs/oss/
openssl/ssl/openssl.cnf
Country Name  (2 letter code) [GB]: JP
State or Province Name (full name) [Berkshire]:Tokyo
Locality Name (eg, city) [Newbury]:Shinagawa
Organization Name (eg, company) [My Company Ltd]:Fujitsu
Organizational Unit Name (eg, section) []:soft
Common Name (eg, your name or your server's hostname) []:myhost.example.com
Email Address []:Enterキー
Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:Enterキー
An optional company name []:Enterキー
```

-config configfilename

openssl設定ファイル(openssl.cnf)の絶対パスを指定します。本オプションを省略しないでください。

openssl設定ファイルは以下に格納されています。

【Windows版】

Systemwalker Operation Managerインストール先ディレクトリ¥MPWALKER.JM¥mpahs¥conf¥openssl.cnf

【UNIX版】

/opt/FJSVftlo/mpahs/oss/openssl/ssl/openssl.cnf

x509サブコマンド

-sha1|-sha256|-md5

フィンガープリント表示で使用するダイジェストを指定します。本オプションを省略した場合は、sha256となります。

-fingerprint

指定されたダイジェストの計算結果を出力します。

-noout

証明書の出力を抑止します。

-in file

入力する証明書ファイル名を指定します。

dhparamサブコマンド

-outform DER | PEM

出力フォーマットを指定します。本オプションを省略した場合は、PEMとなります。

-out filename

出力ファイル名を指定します。本オプションを省略した場合は、標準出力に出力されます。

numbits

生成するパラメータbitサイズを指定します。必ず最後に指定してください。

genpkeyサブコマンド

-aes256

aes256で暗号化します。本オプションを省略した場合は、暗号化されません。

-out filename

出力ファイル名を指定します。本オプションを省略した場合は、標準出力に出力されます。

-algorithm EC

楕円暗号アルゴリズムを指定します。

-pkeyopt

暗号アルゴリズムのオプションを指定します。

opt

ec_paramgen_curve を指定します。

value

以下に示す楕円暗号から認証局がサポートするものを使用してください。

- TLS1.2 の場合

  prime256v1

  sect283k1

  sect283r1

  secp384r1

  sect409k1

  sect409r1

  secp521r1

  sect571k1

  sect571r1

- TLS1.3 の場合

  prime256v1

  secp384r1

  secp521r1

コマンド格納場所

Windows	Systemwalker Operation Managerインストール先ディレクトリ¥mpwalker.jm¥mpahs¥bin
Solaris Linux	/opt/FJSVftlo/mpahs/oss/openssl/bin

実行に必要な権限

システム管理者(Administratorsグループに属するユーザ/スーパーユーザ)権限が必要です。

注意事項

本コマンドは、ネットワーク上でのパスフレーズの漏洩を防ぐため、コンソールで実行するか、または外部から接続されないLANで接続された端末で実行することを推奨します。

使用例

(1) RSA秘密鍵の作成

【Windows版】

```
c:¥Systemwalker¥MPWALKER.JM¥mpahs¥bin¥openssl.exe genrsa -aes256 -out d:¥mpahso¥sslcert¥server.key 2048
(以下はコマンド実行結果です)
Enter pass phrase for server.key: (注)
Verifying - Enter pass phrase for server.key: (注)
```

【UNIX版】

```
# /opt/FJSVftlo/mpahs/oss/openssl/bin/openssl genrsa -aes256 -out /export/home/mpahso/sslcert/server.key
2048
(以下はコマンド実行結果です)
Enter pass phrase for server.key: (注)
Verifying - Enter pass phrase for server.key: (注)
```

注) 本文字列が表示された場合は、パスフレーズを入力してください。なお、入力される文字はエコーバックされません。

(2) CSR(証明書取得申請書)の作成

【Windows版】

```
c:¥Systemwalker¥MPWALKER.JM¥mpahs¥bin¥openssl.exe req -new -key d:¥mpahso¥sslcert¥server.key -config c:
¥Systemwalker¥MPWALKER.JM¥mpahs¥conf¥openssl.cnf -subj "/C=JP/ST=Shizuoka/L=Shizuoka-shi/O=fujitsu/
OU=4-1f/CN=www.example.com"
(以下はコマンド実行結果です)
Enter pass phrase for server.key: (注)
```

【UNIX版】

```
# /opt/FJSVftlo/mpahs/oss/openssl/bin/openssl req -new -key /export/home/mpahso/sslcert/server.key -
config /opt/FJSVftlo/mpahs/oss/openssl/ssl/openssl.cnf -subj "/C=JP/ST=Shizuoka/L=Shizuoka-shi/O=fujitsu/
OU=4-1f/CN=www.example.com"
(以下はコマンド実行結果です)
Enter pass phrase for server.key: (注)
```

注) 本文字列が表示された場合は、パスフレーズを入力してください。なお、入力される文字はエコーバックされません。

(3) 証明書のフィンガープリント表示

【Windows版】

```
c:¥Systemwalker¥MPWALKER.JM¥mpahs¥bin¥openssl.exe x509 -sha256 -fingerprint -noout -in d:¥mpahso¥sslcert
¥ca.crt
sha256 Fingerprint=F7 16 00 6E A1 6E A2 14 ... ←(16進数で32バイト分表示されます。)
```

【UNIX版】

```
# /opt/FJSVftlo/mpahs/oss/openssl/bin/openssl x509 -sha256 -fingerprint -noout -in /export/home/mpahso/
sslcert/ca.crt
sha256 Fingerprint=F7 16 00 6E A1 6E A2 14 ... ←(16進数で32バイト分表示されます。)
```

(4) DHパラメータファイル作成

【Windows版】

```
c:¥Systemwalker¥MPWALKER.JM¥mpahs¥bin¥openssl.exe dhparam -out d:¥mpahso¥sslcert¥dh2048.pem -outform PEM
2048
```

【UNIX版】

```
# /opt/FJSVftlo/mpahs/oss/openssl/bin/openssl dhparam -out /export/home/mpahso/sslcert/dh2048.pem -
outform PEM 2048
```

(5) ECDSA秘密鍵の作成

【Windows版】

```
c:¥Systemwalker¥MPWALKER.JM¥mpahs¥bin¥openssl.exe genpkey -aes256 -out d:¥mpahso¥sslcert¥server.key -
algorithm EC -pkeyopt ec_paramgen_curve:prime256v1
(以下はコマンド実行結果です)
Enter pass phrase for server.key: (注)
Verifying - Enter pass phrase for server.key: (注)
```

【UNIX版】

```
# /opt/FJSVftlo/mpahs/oss/openssl/bin/openssl genpkey -aes256 -out /export/home/mpahso/sslcert/server.key
-algorithm EC -pkeyopt ec_paramgen_curve:prime256v1
Enter pass phrase for server.key: (注)
Verifying - Enter pass phrase for server.key: (注)
```

注) 本文字列が表示された場合は、パスフレーズを入力してください。なお、入力される文字はエコーバックされません。

第13章　保守コマンド

本章では、Systemwalker Operation Managerでトラブルが発生した場合や保守のために必要なコマンドについて説明します。

注意

WindowsのUAC機能を有効にしている環境でコマンドを実行する場合、OSにログインしたユーザの権限によって権限の昇格を求めるダイアログが表示され、対処が必要となります。

詳細は、“Systemwalker Operation Manager 運用ガイド”の、“Windowsでコマンドを実行する場合の注意事項【Windows版】”を参照してください。

13.1　mpipcso IPC資源状態表示コマンド【UNIX版】

記述形式

```
mpipcso
```

機能説明

Systemwalker Operation Managerが現在使用しているIPC資源の情報と、過去に使用して回収できなかったIPC資源の情報を表示します。

復帰値

復帰値	意 味
0	正常終了
0以外	異常終了

コマンド格納場所

/opt/systemwalker/bin

注意事項

- IPC資源の確認はシステムのコマンドipcsも併用して行い、削除するIPC資源の判断を間違えないようにしてください。
- 本コマンドは、システム管理者(スーパーユーザ)が、Systemwalker Operation Managerサーバ上で実行してください。

使用例

```
Mpipcso
```

出力例

コマンドの出力結果の例を以下に示します。

```
# /opt/systemwalker/bin/mpipcso
IPC status from<running Systemwalker> as of Tue Apr 11 10:27:44 2006
T         ID     KEY        DATE       TIME     PID     CMD
s     131075 0x102029b  2006/04/11 10:23:01   11473 MpFwsec
s     131076 0x102029c  2006/04/11 10:23:01   11473 MpFwsec

Status of undeletion IPC from<stopping Systemwalker>
```

```
T        ID    KEY        DATE      TIME     PID     CMD       DETECTION DATE
s     65536 0x1020297  2006/04/10 22:16:46    611 MpFwsec      2006/04/11 10:23:01
s     65537 0x1020299  2006/04/10 22:16:47    611 MpFwsec      2006/04/11 10:23:01

IPC status from<running Systemwalker Centric Manager> as of Tue Apr 11 10:27:44 2006
T        ID    KEY        DATE      TIME     PID     CMD
m       209 0x2097c    2006/04/11 10:23:24  11798 opagtd
m       111 0x1020b46  2006/04/11 10:24:01  12969 mppmon
s        10 0x1020b48  2006/04/11 10:24:00  12969 mppmon

m       108 0x32027da8 2006/04/11 10:23:46  12697 drmsdemon
m       109 0x64027da8 2006/04/11 10:23:46  12697 drmsdemon
s     65541 0x35027da8 2006/04/11 10:23:46  12697 drmsdemon
s         9 0x62027da8 2006/04/11 10:23:46  12697 drmsdemon
s     65542 0x63027da8 2006/04/11 10:23:46  12697 drmsdemon
s         8 0x66027da8 2006/04/11 10:23:46  12697 drmsdemon
q       912 0x30027da8 2006/04/11 10:23:46  12697 drmsdemon
q       115 0x31027da8 2006/04/11 10:23:46  12697 drmsdemon
q       116 0x36027da8 2006/04/11 10:23:46  12697 drmsdemon
q       117 0x38027da8 2006/04/11 10:23:46  12697 drmsdemon
q       118 0x39027da8 2006/04/11 10:23:46  12697 drmsdemon
q       119 0x61027da8 2006/04/11 10:23:46  12697 drmsdemon
q       120 0x65027da8 2006/04/11 10:23:46  12697 drmsdemon
s     65543 0x67027da8 2006/04/11 10:23:46  12697 drmsdemon

Status of undeletion IPC from<stopping Systemwalker Centric Manager>
T        ID    KEY        DATE      TIME     PID     CMD       DETECTION DATE
m         3 0x1020b47  2006/04/10 22:17:37   1946 mppmon       2006/04/11 09:57:34
s     65538 0x1020b46  2006/04/10 22:17:37   1946 mppmon       2006/04/11 09:57:34

IPC status from<running Systemwalker Operation Manager> as of Tue Apr 11 10:27:45 2006
T        ID    KEY        DATE      TIME     PID     CMD
m        10 0x1020fbf  2006/04/11 10:25:07  14002 tskmnsrv
q        67 0x2020fbf  2006/04/11 10:25:10  14210 tskussrv
q       180 0xa020fbf  2006/04/11 10:25:10  14209 tskbasrv
q       121 0x5020fbf  2006/04/11 10:25:10  14212 tskgrsrv
q       122 0x7020fbf  2006/04/11 10:25:10  14213 tskevsrv
m       112 0x1020833  2006/04/11 10:24:50  13613 mpjmsrv

Status of undeletion IPC from<stopping Systemwalker Operation Manager>
T        ID    KEY        DATE      TIME     PID     CMD       DETECTION DATE

There is no IPC resource of the collection leakage.

#
```

IPC status from<running Systemwalker> as of Tue Apr 11 10:27:44 2006

意味：以下に表示する内容が、実行中のSystemwalker Centric Manager およびSystemwalker Operation Managerの共通機能のIPC資源です。日時はmpipcsoコマンドを実行した現在の日時です。

Status of undeletion IPC from<stopping Systemwalker>

意味：以下に表示する内容が、過去に回収できなかったSystemwalker Centric Managerおよび Systemwalker Operation Managerの共通機能のIPC資源です。

IPC status from<running Systemwalker Centric Manager> as of Tue Apr 11 10:27:44 2006

意味：以下に表示する内容が、実行中のSystemwalker Centric ManagerのIPC資源です。日時はmpipcsoコマンドを実行した現在の日時です。

Status of undeletion IPC from<stopping Systemwalker Centric Manager>

意味：以下に表示する内容が、過去に回収できなかったSystemwalker Centric ManagerのIPC資源です。

IPC status from<running Systemwalker Operation Manager> as of Tue Apr 11 10:27:45 2006

意味：以下に表示する内容が、実行中のSystemwalker Operation ManagerのIPC資源です。日時はmpipcsoコマンドを実行した現在の日時です。

Status of undeletion IPC from<stopping Systemwalker Operation Manager>

意味：以下に表示する内容が、過去に回収できなかったSystemwalker Operation ManagerのIPC資源です。

T ID KEY DATE TIME PID CMD

意味：左から順に、種別、ID、キー、獲得日付、獲得時刻、プロセスID、プロセス名を表しています。

T ID KEY DATE TIME PID CMD DETECTION DATE

意味：左から順に、種別、ID、キー、獲得日付、獲得時刻、プロセスID、プロセス名、未回収認識日時を表しています。

There is no IPC resource of the collection leakage.

意味：未回収のIPC資源はありません。

13.2 mppviewo プロセス動作状況表示コマンド

記述形式

【Windows版】

```
mppviewo.exe [-e]
```

【UNIX版】

```
mppviewo [-e]
```

機能説明

Systemwalker Operation Manager自身のプロセスの稼働状況を標準出力へ表示します。

オプション

-e

異常が発生しているプロセス情報だけを表示します。

復帰値

復帰値	意 味
0	正常終了
0以外	異常終了

コマンド格納場所

Windows	Systemwalker Operation Managerインストールディレクトリ¥mpwalker.jm¥bin
Solaris HP-UX AIX Linux	/opt/systemwalker/bin

実行に必要な権限

システム管理者(“Administrators”グループに属するユーザ/スーパーユーザ)

注意事項

- 以下の機能に関しては一覧に表示されません。
  - インストールされていない機能
  - 起動を無効にしている機能。Windows版では、サービスの“スタートアップの種類”で“手動”や“無効”にしている機能
  - 監視を行わない設定を行っている機能
- Systemwalker Centric Managerとの混在環境の場合、以下のプロセス情報は、本コマンドではなくSystemwalker Centric Managerのプロセス動作状況表示コマンド(mppviewc)で表示します。
  - Systemwalker MpAosfP
  - Systemwalker MpAosfB

  Systemwalker Centric Managerのプロセス動作状況表示コマンド(mppviewc)の詳細については、“Systemwalker Centric Manager リファレンスマニュアル”を参照してください。

出力例

コマンドを実行すると以下のようにプロセス稼働状況が表示されます。以下の例では、プロセス“mjsdaemon -sys 1”が異常であることを示しています。

```
####################### PROCESS INFORMATION BEGIN#######################
** Package  Name **
PROCESS-NAME                                PID      RUNNING-TIME

** FJSVfwseo **
MpFwsec                                     2345      0:19

** FJSVjmcal **
f3crhcs2                                    2241      0:18

** FJSVJMCMN **
jmnetsv                                     2245      0:15
mpjmsrv                                     2265      0:17

** FJSVJOBSC **
[Subsystem0]
tskmnsrv                                    2745      0:14
[Subsystem1]
tskmnsrv                                    2749      0:12
[Subsystem2]
tskmnsrv                                    2751      0:09

** FJSVMJS **
[Subsystem0]
mjsdaemon -sys 0                            2821      0:21
mjsdaemon -sys 0                            2822      0:20
mjsdaemon -sys 0                            2823      0:19
mjsnetsv 0                                  2825      0:18
[Subsystem1]
mjsdaemon -sys 1                            2921      0:20
mjsdaemon -sys 1                            2923      0:19
>>>>> ERROR:Process NOT Found : mjsdaemon -sys 1
mjsnetsv 0                                  2925      0:17
[Subsystem2]
mjsdaemon -sys 2                            3024      0:15
mjsdaemon -sys 2                            3025      0:14
```

```
mjsdaemon -sys 2                          3026      0:14
mjsnetsv 0                                3016      0:12
:
###################### PROCESS INFORMATION END  ######################
```

13.3 swcolinf 保守情報収集コマンド(サーバ用)【UNIX版】

記述形式

```
[保守情報の収集の場合]
  swcolinf [-i Name] -o OutPath [-c Comment]
[保守情報の世代管理の設定の場合]
  swcolinf -w Count
[保守情報の一覧表示の場合]
  swcolinf -l OutPath
```

機能説明

Systemwalker Operation Manager上でトラブルが発生した場合に、調査に必要な情報を収集します。収集する情報の世代管理ができるよう、保存する世代数を設定することが可能です。また、一覧表示のオプションで、採取済の資料の情報(採取日付、採取機能など)を世代別に表示できます。

インストールの失敗などで環境が作成されていない場合は、製品メディア内のコマンドを使用して資料を採取できます。

オプション

-i *Name*

採取する機能を選択します。*Name*に指定可能な機能を以下に示します。複数選択する場合は、コンマ(,)で区切ります。ただし、“all”は単一で指定してください。“all”以外の機能と同時に選択できません。

通常は“all”を指定して、すべての機能の情報を採取してください。採取すべき機能が明確になっている、または、富士通技術員からの指示があった場合に限り、個々の機能の情報を採取するようにしてください。

本オプションを省略した場合は、“-i all”を指定した場合と同様に全機能の情報が採取されます。

製品メディア内のコマンドを使用する場合は“all”のみ指定可能です。

all

すべての機能の情報を採取します。

framework

セキュリティ、スクリプトなどの基盤部分の情報を採取します。

server

ジョブスケジューラ、ジョブ実行制御、カレンダ、電源制御など、サーバに導入される機能の情報を採取します。

tool

インストール関連、バックアップ関連、保守ツールなどの情報を採取します。

-o *OutPath*

採取した情報を格納するディレクトリを英数字のフルパスで指定します。*OutPath*は、50バイト以内で、途中に空白は指定できません。

-c *Comment*

採取する情報にコメントを付けたい場合に指定します。コメントは50バイト以内で、英数字および“_”(アンダーバー)を使って指定します。本オプションを省略した場合は、コメントには何も設定されません。

-w *Count*

情報を保存する世代数を指定します。*Count*は、1～10の範囲で指定してください。初期値は3世代が設定されています。

製品メディア内のコマンドを使用する場合は、指定できません。

世代管理を行わない場合は、世代を1と設定してください。

-l *OutPath*

採取資料の一覧を表示させます。採取資料が格納されているディレクトリを指定します。指定された格納先配下の採取資料について、世代別に採取日付/採取機能/コメントが一覧として表示されます。

復帰値

復帰値	意 味
0	正常終了
0以外	異常終了

コマンド格納場所

Solaris	/opt/FJSVftlo/swcolinf
HP-UX	/opt/FJSVftlo/swcolinf
AIX	/opt/FJSVftlo/swcolinf
Linux	/opt/FJSVftlo/swcolinf
製品メディア(Solaris)	<ドライブ>/Solaris/tool_unx/swcolinf
製品メディア(Linux)	<ドライブ>/Linux/tool_unx/swcolinf

実行に必要な権限

システム管理者(スーパーユーザ)

注意事項

- 採取に失敗した機能がある場合は、その旨画面に表示されます。画面の指示に従って、対象ファイルを手動で格納先にコピーしてください。画面の表示例については、“Systemwalker Operation Manager トラブルシューティングガイド”を参照してください。
- このコマンドはサイレントのため、エラーがある場合は、以下のログに記述しています。
  - コマンドのオプションエラー:正しいオプションを指定してください。

    【Solaris版】

    ```
    /var/tmp/swcolinf.log
    ```

    【HP-UX版/AIX版/Linux版】

    ```
    /tmp/swcolinf.log
    ```

  - その他のエラー:失敗したファイルを手動で格納先にコピーしてください。

    ```
    情報格納先ディレクトリ/OperationManager/世代番号/
    swcolinf.log
    ```

使用例1

サーバ機能とツールの情報を採取し、/workに格納します。

```
/opt/FJSVftlo/swcolinf/swcolinf -i server,tool -o /work
```

使用例2

すべての機能の情報を採取し、コメントを付けて/workに格納します。

```
/opt/FJSVftlo/swcolinf/swcolinf -i all -o /work -c 2002.2.2_all_SE001
```

使用例3

/work配下の採取資料についての情報一覧を表示させます。

```
/opt/FJSVftlo/swcolinf/swcolinf -l /work
```

実行結果

```
***  Systemwalker Operation Manager Information ***
OUTPATH:/work
FOLDER                DATE              FUNCTION    COMMENT
/work/OperationManager/0001  2002/05/05 10:10:10    all        all
/work/OperationManager/0002 2002/05/06 10:10:10   tool     tool:error
```

出力項目について、以下に説明します。

OUTPATH:

指定した格納先

FOLDER:

格納されている世代ディレクトリ

DATE:

採取した日付

FUNCTION:

採取した機能

COMMENT:

コマンドで指定したコメント

13.4　swcolinf 保守情報収集コマンド

記述形式

```
[保守情報の収集の場合]
  swcolinf.exe [/i Name] /o OutPath [/c Comment]
[GUIを利用した保守情報の収集の場合]
  swcolinf.exe
[保守情報の世代管理の設定の場合]
  swcolinf.exe /w Count
```

機能説明

Systemwalker Operation Manager上でトラブルが発生した場合に、調査に必要な情報を収集します。収集する情報の世代管理ができるよう、保存する世代数を設定することが可能です。

インストールの失敗などで環境が作成されていない場合は、製品メディア内のコマンドを使用して資料を採取できます。

Server Core環境では、オプションなしのswcolinf.exeコマンドを実行することで、GUIを利用して保守情報を収集してください。

オプション

/i *Name*

採取する機能を選択します。*Name*に指定可能な機能を以下に示します。複数選択する場合は、コンマ(,)で区切ります。ただし、“all”は単一で指定してください。“all”以外の機能と同時に選択できません。

通常は“all”を指定して、すべての機能の情報を採取してください。採取すべき機能が明確になっている、または、富士通技術員からの指示があった場合に限り、個々の機能の情報を採取するようにしてください。

本オプションを省略した場合は、“/i all”を指定した場合と同様に全機能の情報が採取されます。

製品メディア内のコマンドを使用する場合は“all”のみ指定可能です。

all

すべての機能の情報を採取します。

framework

セキュリティなどの基盤部分の情報を採取します。

network

ネットワーク関連の情報を採取します。

event

イベント監視、アクション管理などの情報を採取します。

auto【Windows版】

カレンダ、電源制御などの情報を採取します。

server【Windows版】

ジョブスケジューラ、ジョブ実行制御、バックアップ連携など、サーバに導入される機能の情報を採取します。

client

Systemwalker Operation Managerクライアント機能、業務連携などの情報を採取します。

tool

インストール関連、バックアップ関連、保守ツールなどの情報を採取します。

/o *OutPath*

採取した情報を格納するディレクトリをフルパスで指定します。*OutPath*は、50バイト以内で途中に空白は指定できません。

/c *Comment*

採取する情報にコメントを付けたい場合に指定します。*Comment*は、50バイト以内で途中に空白と“#”は指定できません。本オプションを省略した場合は、コメントには何も設定されません。

/w *Count*

情報を保存する世代数を指定します。*Count*は、1～10の範囲で指定してください。初期値は3世代が設定されています。

製品メディア内のコマンドを使用する場合は、指定できません。

世代管理を行わない場合は、世代を1と設定してください。

なし

[保守情報収集ツール]ウィンドウが表示されます。Server Core環境の場合は、オプションなしで実行してください。

復帰値

復帰値	意 味
0	正常終了
0以外	異常終了

コマンド格納場所

Windows	Systemwalker Operation Managerインストールディレクトリ ¥mpwalker.jm¥mpcmtool¥swcolinf
製品メディア(Windows x86)	<ドライブ>¥MAIN¥tool¥swcolinf
製品メディア(Windows x64)	<ドライブ>¥Client¥tool¥swcolinf
製品メディア(UNIX版 クライアント)	<ドライブ>¥Client¥tool¥swcolinf

実行に必要な権限/実行環境

- システム管理者(Administratorsグループに属するユーザ)
- Windows OS上で本コマンドを実行する場合、管理者権限で起動したコマンドプロンプト上で本コマンドを実行してください。

注意事項

- このコマンドはサイレントのため、エラーがある場合は、以下のログに記述しています。
  - コマンドのオプションエラー:正しいオプションを指定してください。

    ```
    Windowsディレクトリ¥swcolinf.log
    ```

  - その他のエラー:失敗したファイルを手動で格納先にコピーしてください。

    ```
    情報格納先ディレクトリ¥OperationManager¥世代番号
    ¥swcolinf.log
    ```

- Systemwalker Centric Managerとの混在環境の場合は、バージョンが新しい製品の保守情報収集ツールを使用して古いバージョンの製品の資料採取を行ってください。

使用例1

サーバ機能とツールの情報を採取し、c:¥tmpに格納します。

```
c:¥win32app¥mpwalker.jm¥mpcmtool¥swcolinf¥swcolinf.exe /i
server,tool
/o c:¥tmp
```

使用例2

すべての機能の情報を採取し、コメントを付けてc:¥tmpに格納します。

```
c:¥win32app¥mpwalker.jm¥mpcmtool¥swcolinf¥swcolinf.exe /i all /o
c:¥tmp
/c 2002.2.2_all_SE001
```

使用例3

Server Core環境において、GUIを利用して保守情報を収集します。

```
c:¥win32app¥mpwalker.jm¥mpcmtool¥swcolinf¥swcolinf.exe
```

13.5 mpenvset 運用環境保守ウィザード起動コマンド【Windows版】

記述形式

```
mpenvset.exe
```

機能説明

運用環境保守ウィザードを起動し、運用環境の退避および復元を行います。コマンドを実行すると[運用環境保守ウィザード]画面が表示されます。画面に従って操作をすることでSystemwalker Operation Manager環境の退避/復元を行うことができます。

オプション

なし

復帰値

0	正常終了
0以外	異常終了

コマンド格納場所

Windows	Systemwalker Operation Managerインストールディレクトリ¥mpwalker.jm¥mpcmtool¥common

実行に必要な権限/実行環境

- システム管理者(Administratorsグループに属するユーザ)権限が必要です。
- サーバで実行可能です。

使用例

```
Systemwalker Operation Managerインストールディレクトリ¥mpwalker.jm
¥mpcmtool¥common¥mpenvset.exe
```

GEE

第14章　グローバルサーバジョブ連携コマンド

本章では、Systemwalker Operation Managerのグローバルサーバジョブ連携コマンドについて説明します。

14.1　コマンド一覧

グローバルサーバジョブ連携コマンドで使用するコマンドには以下のものがあります。

ジョブ起動コマンド

グローバルサーバジョブを起動するコマンドについて説明します。ジョブ起動コマンドの一覧を以下に示します。

mfexecu 未定義ジョブ起動コマンド

SymfoNETに業務定義していないジョブの起動を依頼するコマンドです。

mfexecd 定義済みジョブ起動コマンド

SymfoNETに業務定義してあるジョブの起動を依頼するコマンドです。

注意

ジョブ起動コマンドは、[登録－ジョブ]ウィンドウから登録されることにより起動されるコマンドです。シェルからの直接コマンド入力で起動しないでください。

情報定義コマンド

グローバルサーバジョブの情報定義コマンドについて説明します。情報定義コマンドの一覧を以下に示します。

mfusradd 利用者登録コマンド

グローバルサーバジョブ連携用の利用者を登録するコマンドです。

mfusrdel 利用者削除コマンド

グローバルサーバジョブ連携用の利用者を削除するコマンドです。

mfusrmod 利用者変更コマンド

グローバルサーバジョブ連携用の利用者を変更するコマンドです。

mfusrpasswd 利用者パスワード登録コマンド

グローバルサーバジョブ連携用の利用者のパスワードを登録するコマンドです。

mfjobadd 業務情報登録コマンド

ジョブの業務情報を登録するコマンドです。

mfjobdel 業務情報削除コマンド

ジョブの業務情報を削除するコマンドです。

mfjobmod 業務情報変更コマンド

ジョブの業務情報を変更するコマンドです。

mfjobpasswd 業務パスワード登録コマンド

ジョブの業務情報にパスワードを登録するコマンドです。

14.2　mfexecu 未定義ジョブ起動コマンド

記述形式

```
mfexecu -j 業務ID -u ユーザ定義名 -h 実行ホスト名 -s OS種別 [-t 業務種別]
    [-p リトライ間隔] [-a 代替ホスト名] [-o ジョブへのパラメタ]
    実行ファイル名
```

機能説明

SymfoNETで業務定義されていないグローバルサーバジョブの起動をグローバルサーバに依頼するコマンドです。本コマンドは、[登録－ジョブ]ウィンドウから登録されることによって起動するため、実際には使用しないコマンドです。

オペランド

-j 業務ID

ジョブの業務IDを指定します。業務IDは、長さ8バイト以内の英数字です。グローバルサーバ上のJCL中のジョブ名を指定してください。

-u ユーザ定義名

グローバルサーバへの接続に使用するユーザ名を指定します。

ユーザ定義名は、グローバルサーバへ接続するためのUNIXシステム上のグループ名であり、あらかじめmfusraddコマンド、mfusrpasswdコマンドを使ってグローバルサーバのユーザ名と対応づけて登録しておいてください。

-h 実行ホスト名

ジョブを実行するグローバルサーバのホスト名を指定します。

-s OS種別

ジョブを実行するサーバのOS名です。

MSPまたはXSPのいずれかを指定します。

-t 業務種別

実行する業務の種別を指定します。

業務種別にはbatch(バッチ業務)またはjcm(JCM業務)のいずれかを指定します。省略した場合は、バッチ業務として実行依頼します。OS種別にMSPを指定した場合には、jcmの指定は無効となります。

-p リトライ間隔

グローバルサーバへ接続できなかった際に、リトライする間隔を秒単位(5～30)で指定します。

本項目が省略された場合は、リトライを行いません。リトライ回数は10回です。

-a 代替ホスト名

指定したグローバルサーバに接続できなかった場合に、代わりに接続するサーバのホスト名を指定します。

本項目が省略された場合、代替接続は行いません。

-o ジョブへのパラメタ

ジョブに渡すパラメタを指定します。

実行ファイル名

ジョブとして起動する、グローバルサーバ上のファイル名です。OS種別がXSPの場合は実行ファイル名またはJCM名を、MSPの場合はデータセット名を指定します。

注意事項

- 本コマンドは、[登録−ジョブ]ウィンドウから登録されることによって起動するため、実際には使用しません。シェルからの直接コマンド入力で起動しないでください。
- グローバルサーバのOSがXSPの場合、同じ業務IDのジョブを同時に実行することはできません。
- 本コマンドは終了待ち合わせを行うため、グローバルサーバジョブの終了まで復帰しません。

14.3 mfexecd 定義済みジョブ起動コマンド

記述形式

```
mfexecd -j 業務ID -u ユーザ定義名 -h 実行ホスト名 -s OS種別
[-p リトライ間隔] [-a 代替ホスト名] [-o ジョブへのパラメタ]
```

機能説明

SymfoNETで業務定義されたグローバルサーバジョブの起動をグローバルサーバに対して依頼するコマンドです。

オペランド

-j 業務ID

SymfoNETに定義されている業務IDを8バイト以内の英数字で指定します。

-u ユーザ定義名

グローバルサーバへの接続に使用するユーザ名を指定します。

ユーザ定義名は、グローバルサーバへ接続するためのUNIXシステム上のグループ名であり、あらかじめmfusraddコマンド、mfusrpasswdコマンドを使ってグローバルサーバのユーザ名と対応づけて登録しておいてください。

-h 実行ホスト名

ジョブを実行するグローバルサーバのホスト名を指定します。

-s OS種別

ジョブを実行するサーバのOS種別です。

MSPまたはXSPのいずれかを指定します。

-p リトライ間隔

グローバルサーバへ接続できなかった際に、リトライする間隔を秒単位(5〜30)で指定します。

本項目が省略された場合は、リトライを行いません。リトライ回数は10回です。

-a 代替ホスト名

指定したグローバルサーバに接続できなかった場合に、代わりに接続するサーバのホスト名を指定します。

本項目が省略された場合は、代替接続は行いません。

-o ジョブへのパラメタ

ジョブに渡すパラメタを指定します。

注意事項

- 本コマンドは、[登録−ジョブ]ウィンドウから登録されることによって起動するため、実際には使用しません。シェルからの直接コマンド入力で起動しないでください。

- 本コマンドでは、グローバルサーバのOSがXSPの場合、同じ業務IDのジョブを同時に実行することはできません。
- 本コマンドは終了待ち合わせを行うため、グローバルサーバジョブの終了まで復帰しません。

14.4 mfusradd 利用者登録コマンド

記述形式

```
mfusradd -u ジョブ実行ユーザ名 [-a 注釈] ユーザ定義名
```

機能説明

グローバルサーバと連携する時に使用する利用者を登録します。

Systemwalker Operation Managerはグローバルサーバとの連携を、グローバルサーバに登録されているユーザ名を使って行います。本コマンドを使って、グローバルサーバに登録されているユーザ名(ジョブ実行ユーザ名)に対応した利用者名(ユーザ定義名)を、Systemwalker Operation Manager側に登録します。

本コマンドで利用者を登録した後に、“14.7 mfusrpasswd 利用者パスワード登録コマンド”でパスワードを登録してください。

オペランド

-u ジョブ実行ユーザ名

ジョブの実行を依頼するグローバルサーバと接続するために使用する、グローバルサーバに登録されているユーザ名を指定します。ジョブ実行ユーザ名は、長さ8バイト以内の英数字です。

-a 注釈

注釈を64バイト以内で指定します。

ユーザ定義名

グローバルサーバ接続用のユーザをグループ名単位で指定します。ユーザ定義名は、長さ8バイト以内の英数字です。

ユーザ定義名で指定したグループに属するユーザは、指定されたグローバルサーバへジョブの実行を依頼することができます。

注意事項

- 本コマンドはフルパスで使用してください。本コマンドは“/opt/FJSVsjblk/sbin”配下に格納されています。
- 本コマンドは、シェルからシステム管理者が実行してください。
- 各オペランドに使用できる文字は英数字だけです。
- 登録済みのユーザ定義名を指定するとエラーとなります。必ず未登録のユーザ定義名を指定してください。

14.5 mfusrdel 利用者削除コマンド

記述形式

```
mfusrdel ユーザ定義名
```

機能説明

登録したグローバルサーバ連携用の利用者を削除します。

オペランド

ユーザ定義名

“14.4 mfusradd 利用者登録コマンド”で登録したユーザ定義名の中で、削除するユーザ定義名を指定します。

注意事項

- 本コマンドはフルパスで使用してください。本コマンドは“/opt/FJSVsjblk/sbin”配下に格納されています。
- 本コマンドは、シェルからシステム管理者が実行してください。
- 各オペランドに使用できる文字は英数字だけです。

14.6 mfusrmod 利用者変更コマンド

記述形式

```
mfusrmod -u ジョブ実行ユーザ名 [-a 注釈] ユーザ定義名
```

機能説明

グローバルサーバ連携用の利用者情報を変更します。

オペランド

-u ジョブ実行ユーザ名

ジョブの実行を依頼するグローバルサーバと接続するために使用する、グローバルサーバに登録されているユーザ名を指定します。ジョブ実行ユーザ名は、長さ8バイト以内の英数字です。

-a 注釈

注釈を64バイト以内で指定します。

ユーザ定義名

グローバルサーバに接続するユーザをグループ名単位で指定します。

ユーザ定義名で指定したグループに属するユーザは、指定されたグローバルサーバへジョブの実行を依頼することができます。

注意事項

- 本コマンドはフルパスで使用してください。本コマンドは“/opt/FJSVsjblk/sbin”配下に格納されています。
- 本コマンドは、シェルからシステム管理者が実行してください。
- 各項目中に使用できる文字は英数字だけです。
- 本コマンドで、利用者登録されていないユーザ定義名を指定した場合はエラーとなります。必ず登録済みのユーザ定義名を指定してください。

14.7 mfusrpasswd 利用者パスワード登録コマンド

記述形式

```
mfusrpasswd ユーザ定義名
```

機能説明

グローバルサーバ連携用の利用者のパスワード登録を行います。

本コマンドを起動すると、以下のようにパスワード入力画面が表示されます。画面の出力に対して8文字以内の文字列を入力します。

```
New MF passwd:          ← 8文字以内の文字列を入力します。
Retype new MF passwd:   ← 先ほど入力した文字列を再度入力します。
```

オペランド

ユーザ定義名

パスワードを設定するユーザ定義名を指定します。“14.4 mfusradd 利用者登録コマンド”で登録済みのユーザ定義名を指定してください。

注意事項

- 本コマンドはフルパスで使用してください。本コマンドは“/opt/FJSVsjblk/sbin”配下に格納されています。
- 本コマンドは、シェルからシステム管理者が実行してください。
- 各項目中に使用できる文字は英数字だけです。
- すでにパスワードが登録されているユーザ定義名を指定した場合も、パスワード入力画面が表示され新しいパスワードに更新されます。

14.8 mfjobadd 業務情報登録コマンド

記述形式

```
mfjobadd [-a 注釈] 業務ID
```

機能説明

SymfoNET側で業務パスワードを設定した業務に対して、Systemwalker Operation Managerから実行依頼できるようにジョブの業務IDを登録します。本コマンドで業務IDを登録した後に、“14.11 mfjobpasswd 業務パスワード登録コマンド”を使って業務パスワードを登録してください。

業務パスワードが設定されていない業務を実行する場合は、本コマンドによる登録は必要ありません。

オペランド

-a 注釈

注釈を64バイト以内で指定します。

業務ID

登録対象となるジョブの業務IDを指定します。SymfoNETで登録した業務IDと同じIDを指定してください。

業務IDは、長さ8バイト以内の英数字です。

注意事項

- 本コマンドはフルパスで使用してください。本コマンドは“/opt/FJSVsjblk/sbin”配下に格納されています。
- 本コマンドは、シェルからシステム管理者が実行してください。
- 各項目中に使用できる文字は英数字だけです。
- 業務IDとmfjobpasswdコマンドで設定する業務パスワードの対応は、SymfoNET側で登録した業務IDと業務パスワードの対応と同じものを指定してください。

14.9 mfjobdel 業務情報削除コマンド

記述形式

```
mfjobdel 業務ID
```

機能説明

mfjobaddコマンドで登録した業務情報を削除するコマンドです。

オペランド

業務ID

削除対象となる業務IDを指定します。

業務IDは、長さ8バイト以内の英数字です。

注意事項

- 本コマンドはフルパスで使用してください。本コマンドは“/opt/FJSVsjblk/sbin”配下に格納されています。
- 本コマンドは、シェルからシステム管理者が実行してください。
- 各項目中に使用できる文字は英数字だけです。

14.10 mfjobmod 業務情報変更コマンド

記述形式

```
mfjobmod [-a 注釈] 業務ID
```

機能説明

mfjobaddコマンドで登録したSymfoNET業務の業務情報を変更するコマンドです。

オペランド

-a 注釈

注釈を64バイト以内で指定します。

業務ID

変更対象となるジョブの業務IDを指定します。

業務IDは、長さ8バイト以内の英数字です。

注意事項

- 本コマンドはフルパスで使用してください。本コマンドは“/opt/FJSVsjblk/sbin”配下に格納されています。
- 本コマンドは、シェルからシステム管理者が実行してください。
- 各項目中に使用できる文字は英数字だけです。
- 本コマンドで指定した業務IDとmfjobpasswdコマンドで設定する業務パスワードの対応は、SymfoNET側で登録した業務IDと業務パスワードの対応と同じものを指定してください。

14.11 mfjobpasswd 業務パスワード登録コマンド

記述形式

```
mfjobpasswd 業務ID
```

機能説明

mfjobaddコマンドで登録した業務にパスワードを設定します。

本コマンドを起動すると、以下のようにパスワード入力画面が表示されるので、8文字以内の文字列を入力してください。

```
New JOB passwd:           ← 8文字以内の文字列を入力します。
Retype new JOB passwd:    ← 先に入力した文字列を再度入力します。
```

SymfoNETで業務パスワードを設定していない業務を実行する場合は、本コマンドによるパスワードの設定は必要ありません。

オペランド

業務ID

業務パスワードを登録するジョブの業務IDを指定します。

業務IDは、長さ8バイト以内の英数字です。

注意事項

- 本コマンドはフルパスで使用してください。本コマンドは“/opt/FJSVsjblk/sbin”配下に格納されています。
- 本コマンドは、シェルからシステム管理者が実行してください。
- 各項目中に使用できる文字は英数字だけです。
- 業務IDと設定する業務パスワードの対応は、SymfoNET側で登録した業務IDと業務パスワードの対応と同じものを指定してください。

第15章　JCL(業務手続き言語)

本章では、Systemwalker Operation Managerが提供するJCL(業務手続き言語)について説明します。

JCL(業務手続き言語)とは、複雑な業務の手順を容易に記述できる、ジョブの実行制御に適した言語です。ジョブステップの概念を取り入れ、ジョブステップの構成で業務の手続きを表現します。

15.1　JCLの概要

本節では、JCLの概要について説明します。

15.1.1　JCLの機能

JCLはジョブステップの概念を取り入れ、ジョブステップの構成でジョブの手続きを表現します。

ジョブステップにおいて、プログラム(実行プログラム、バッチファイル、シェルスクリプトなど)の起動を行います。

ジョブ投入時に指定したパラメタをプログラムまで引き継ぎます。

ジョブステップは、ジョブステップの終了コードにより次に実行するジョブステップを選択的に実行することができます。

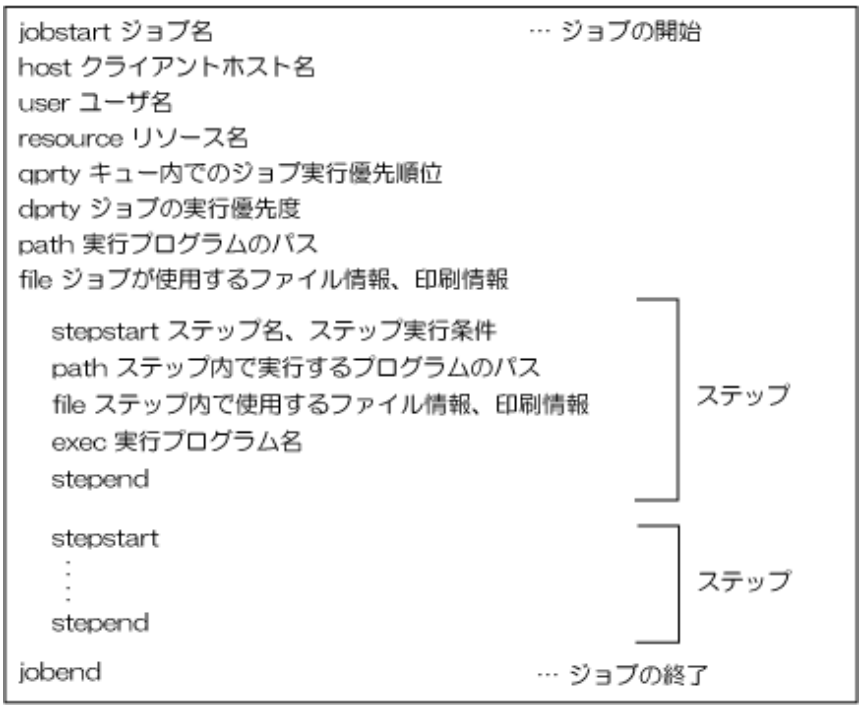
バッチジョブ間の排他を行うことができます。

JCLファイルをネットワークジョブとして実行することはできません。

15.1.2　JCLの構成

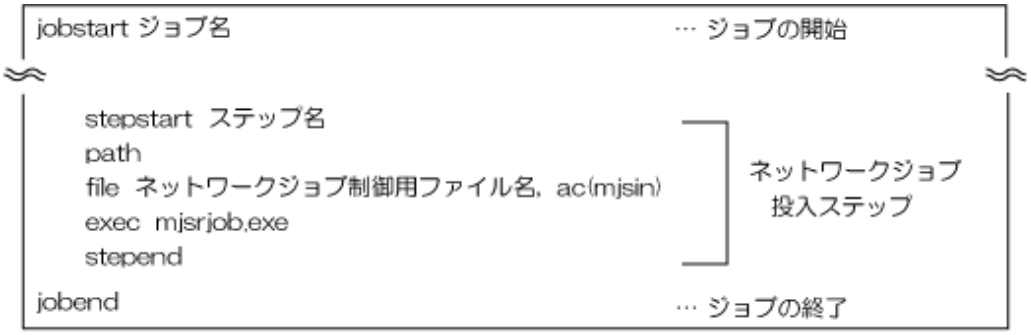
JCLはジョブ全体に関する制御文と、ジョブステップに関する制御文の群から構成されます(構成や形式を説明するために、制御文のパラメタは簡略化しています)。

以下にJCLの構成を示します。



ネットワークジョブを投入する場合のJCLの形式を以下に示します。

図15.1 ネットワークジョブのジョブステップ構成



ネットワークジョブ制御用ファイル

connect 実行先サーバ	… 実行先サーバとの回線を接続
send	… ジョブ投入元サーバ上のファイルを実行先サーバへ送信
execute 実行プログラム名	
receive	… 実行先サーバ上のファイルをジョブ投入元サーバに受信
disconnect	… 実行先サーバとの回線を切断

15.1.3 JCLの規約

以下にJCLの規約を説明します。また、制御文の一覧を示します。

制御文 オペランド

制御文

- 1カラム目以降に記述します。

  制御文によっては必ず1カラム目から記述しなければならないものがあります(下表参照)。
- 1行に複数の制御文を記述することはできません。
- 行を継続して記述することはできません。

制御文	意味	記述位置
jobstart	ジョブの開始	先頭行のみ記述できます。 必ず第1カラムから記述します。
jobend	ジョブの終了	最終行のみ記述できます。 必ず第1カラムから記述します。
stepstart	ジョブステップの開始	必ず第1カラムから記述します。
stepend	ジョブステップの終了	必ず第1カラムから記述します。
host	クライアントホスト名	第1カラム以降から記述します。
user	クライアントユーザ名	第1カラム以降から記述します。
resource	リソース名と排他属性	第1カラム以降から記述します。
qprty	キュー内のジョブ実行優先順位	第1カラム以降から記述します。
dprty	ジョブ実行優先度	第1カラム以降から記述します。
path	プログラムが存在するパス	第1カラム以降から記述します。

制御文	意味	記述位置
file	ジョブ全体で使用するファイル	第1カラム以降から記述します。
exec	プログラムの実行	第1カラム以降から記述します。

オペランド

- 制御文との間に1つ以上の空白を空けて記述します。
- 複数のオペランドを記述する場合、各オペランドを“,”で区切ります。
- 可変記号を記述することができます。

  可変記号の表記は“&n”であり、nはジョブ投入時に指定したパラメタの何個目かを指定します。

コメント

- 1カラム目に“*”または“#”を記述するとその行はコメント行となり、行コメントを記述することができます。
- 制御文の行中にコメントは記述できません。

15.2 ジョブ全体に関する制御文

本節では、ジョブ全体に関する制御文について説明します。

15.2.1 jobstart文

記述形式

```
jobstart [ジョブ名] [,prt(プリンタ名)  [,op] [,oh] [,nop] ]
        [,joblst（{0｜1}）]
```

機能説明

ジョブの開始を意味する制御文であり、他のJCL制御文よりも先に記述しなければなりません。

ジョブの一括出力をするか否かの指定およびジョブ出力を保留するか否かを指定します。

オペランド

ジョブ名

ジョブ名を指定します。ジョブ名は、最大64バイトまで指定できます。

qsubコマンドで-jオプションが指定された場合、-jオプションで指定したジョブ名が有効になります。

[登録－ジョブ]ウィンドウ－[基本情報]シートで[ジョブ名]が指定されたスケジュールジョブの場合、[ジョブ名]で指定したジョブ名が有効になります。

[ジョブ情報編集／投入]ダイアログボックス/ウィンドウ－[基本情報]シートで[ジョブ名]が指定されたデマンドジョブの場合、[ジョブ名]で指定したジョブ名が有効になります。

本オペランドを省略した場合で、かつ、以下が指定されていない場合は、ジョブファイル名が有効になります。

- qsubコマンドにおける-jオプション
- [登録－ジョブ]ウィンドウ－[基本情報]シートにおける[ジョブ名]
- [ジョブ情報編集／投入]ダイアログボックス/ウィンドウ－[基本情報]シートにおける[ジョブ名]

prt(プリンタ名)

JCLの標準出力、標準エラー出力およびジョブリストを出力するプリンタ名を指定します。opオペランド、ohオペランド、またはnopオペランドを指定する場合は、本オペランドが必須です。

また、本オペランドで指定したプリンタ名が、file制御文におけるprtオペランドの省略値として採用されます。このため、各file制御文においてプリンタ名を省略したい場合は、本オペランドを指定する必要があります。

指定するプリンタ名は、以下の条件を満たしている必要があります。

- 32バイト以下の半角英数字である
- “&”、“(”および“)”を含まない

プリンタ名は本名の他にジョブ実行制御の[運用情報の定義]ウィンドウの[印刷形式]シートで定義した別名も指定することができます。【Windows版】

op

ジョブ一括出力時に、標準出力ファイル、標準エラー出力ファイルおよびジョブリストファイルを出力する場合に指定します。

oh

ジョブ終了後、ジョブ出力を保留状態とします。本オペランドを指定しない場合は、ジョブ終了後、自動的に出力依頼が行われ、ジョブは削除されます。

nop

ジョブ出力がある場合(JCLで、prtオペランドの指定がある場合)、ジョブ出力完了後、ジョブを削除せずに、出力保留ジョブとしてSystemwalker Operation Managerを導入しているサーバ内に保存します。

保存されたジョブは、クライアント画面または、サーバ上のコマンドにおいて「保留解除」の操作をすることにより、再出力することができます。再出力後もジョブはSystemwalker Operation Managerを導入しているサーバ内に保存されます。

本オプションの指定によりSystemwalker Operation Managerを導入しているサーバ内に保存されたジョブは、削除操作しない限りSystemwalker Operation Managerを導入しているサーバから削除されません。不要になったジョブは、クライアント画面または、サーバ上のコマンドにおいて「ジョブ削除」を行ってください。

joblst({0|1})

joblstオペランドで(1)を指定するとジョブリストが出力されます。joblstオペランドで(0)を指定した場合および、joblstオペランドを省略した場合は、ジョブリストは出力されません。

ジョブリストファイルは、標準出力ファイルと同じパスに、以下の名前で作成されます。

```
…¥ジョブ名.lジョブ番号
```

ジョブリストには以下の情報が出力されます。

- ジョブの実行開始日時
- ジョブステップの実行開始日時
- ジョブステップの実行終了日時
- ジョブステップの終了コード
- ジョブの実行終了日時
- ジョブの終了コード
- ネットワークジョブの開始/終了日時
- ネットワークジョブのファイル転送量

ジョブリストの出力例を以下に示します。

```
MJS800I MJSRJOB JOB CONTROL START 10/14.11:11: 1
MJS802I CONNECT TO nomonomo
MJS804I send1 SENT 8BYTES
MJS805I JOB START ON nomonomo 10/14.11:11: 3
MJS806I JOB STOP ON nomonomo 10/14.11:11:18 CODE(0)
MJS804I STDOUT RECEIVED 18KB
MJS804I STDERR RECEIVED 0BYTES
MJS804I recv1 RECEIVED 9BYTES
MJS803I DISCONNECT TO nomonomo
MJS801I MJSRJOB JOB CONTROL STOP 10/14.11:11:19
```

記述例

ジョブ名が“updatejob”、出力先が“printer1”で標準出力/標準エラーを出力する指定、その出力はジョブが終了しても即時印刷せずに出力保留状態とし、さらに保留を解除して印刷をしてもジョブをパージせずに再び出力保留状態とする、また、ジョブリストを出力させる指定の場合の記述例を以下に示します。

```
jobstart updatejob,prt(printer1),op,oh,nop,joblst(1)
```

15.2.2 host文

記述形式

```
host ホスト名
```

機能説明

クライアントホスト名を指定します。

オペランド

ホスト名

クライアントホスト名を指定します。

クライアントホスト名は、最大64バイトまでの英数字で指定できます。

qsubコマンドで-chオプションが指定された場合、-chオプションで指定したクライアントホスト名が有効になります。

記述例

```
host konzert
```

konzert：ホスト名

15.2.3 user文

記述形式

```
user ユーザ名
```

機能説明

クライアントユーザ名を指定します。

オペランド

ユーザ名

クライアントユーザ名を指定します。

クライアントユーザ名は、最大64バイトまでの英数字で指定できます。

qsubコマンドで-cuオプションが指定された場合、-cuオプションで指定したクライアントユーザ名が有効になります。

スケジュールジョブの場合、プロジェクトの所有者がクライアントユーザ名になります。ただし、[登録－ジョブ]ウィンドウ－[詳細情報]シートで[実行ユーザ名]が指定されたスケジュールジョブの場合、[実行ユーザ名]で指定したユーザがクライアントユーザ名になります。

記述例

```
user s7025
```

s7025：ユーザ名

注意事項

この制御文はシステム管理者(“Administrators”グループに属するユーザ/スーパーユーザ)のみ指定できます。

15.2.4 resource文

記述形式

```
resource リソース名 [,  {sh | ex} ]
```

機能説明

当該ジョブが使用するリソース名と排他属性を指定します。

ジョブの使用するリソースが複数ある場合は、本制御文を複数行記述します。

記述できる本制御文の数は以下のとおりです。

【Windows版】

- 最大16個

【UNIX版】

- 最大256個

オペランド

リソース名

ジョブが使用するリソース名を指定します。

リソース名は、最大64バイトまで指定できます。半角英数字、“_”(半角アンダーバー)、“-”(半角ハイフン)を利用することが可能です。

qsubコマンドで-Rオプションが指定された場合、-Rオプションで指定したリソース情報が有効になります。

[登録－ジョブ]ウィンドウ－[詳細情報]シートで[リソース名]が指定されたスケジュールジョブの場合、[リソース名]および[リソース割当]で指定したリソース情報が有効になります。

[ジョブ情報編集／投入]ダイアログボックス/ウィンドウ－[拡張情報]シートで[使用するリソース名]が指定されたデマンドジョブの場合、[使用するリソース名]および[属性]で指定したリソース情報が有効になります。

{sh|ex}

排他属性を指定します。shは共有属性、exは占有属性を示します。

省略した場合、shとなります。

記述例

```
resource RDB,ex
```

RDB：リソース名
ex：排他属性

15.2.5 qprty文

記述形式

```
qprty キュー優先順位
```

機能説明

キュー内のジョブ実行優先順位を指定します。

オペランド

キュー優先順位

キュー内のジョブ実行優先順位を0～63の数値で指定します。

値が大きいほど実行優先順位が高いことを示します。

qsubコマンドで-pオプションが指定された場合、-pオプションで指定した実行優先順位が有効になります。

[ジョブ情報編集／投入]ダイアログボックス／ウィンドウ－[拡張情報]シートで[実行優先順位を設定する]が指定されたデマンドジョブの場合、[実行優先順位を設定する]で指定した実行優先順位が有効になります。

記述例

```
qprty 32
```

32：キュー優先順位

15.2.6 dprty文

記述形式

```
dprty exec文に指定したプログラムの実行優先度
```

機能説明

ジョブ実行優先度を指定します。

オペランド

exec文に指定したプログラムの実行優先度

【Windows版】

exec文に指定したプログラムのスレッドの優先度を、0～4の数値で指定します。

値が大きいほどスレッドの優先度が高くなります。

指定できる0～4の値は、以下に示す意味を持ちます。

指定値	優先順位クラス	スレッドの優先順位
4	NORMAL	HIGHEST
3	NORMAL	ABOVE_NORMAL
2	NORMAL	NORMAL
1	NORMAL	BELOW_NORMAL
0	NORMAL	LOWEST

qsubコマンドで-dpオプションが指定された場合、-dpオプションで指定した実行優先度が有効になります。

[登録－ジョブ]ウィンドウ－[詳細情報]シートで[実行優先度]が指定されたスケジュールジョブの場合、[実行優先度]で指定した実行優先度が有効になります。

[ジョブ情報編集／投入]ダイアログボックス/ウィンドウ－[拡張情報]シートで[実行優先度を設定する]が指定されたデマンドジョブの場合、[実行優先度を設定する]で指定した実行優先度が有効になります。

【UNIX版】

exec文に指定したプログラムの実行優先度を、-20～19の数値で指定します。

値が小さいほどCPU割当て順位が高いことを示します。

qsubコマンドで-dpオプションが指定された場合、-dpオプションで指定した実行優先度が有効になります。

[登録－ジョブ]ウィンドウ－[詳細情報]シートで[実行優先度]が指定されたスケジュールジョブの場合、[実行優先度]で指定した実行優先度が有効になります。

[ジョブ情報編集／投入]ダイアログボックス/ウィンドウ－[拡張情報]シートで[実行優先度を設定する]が指定されたデマンドジョブの場合、[実行優先度を設定する]で指定した実行優先度が有効になります。

本オペランドで指定できる値は、ナイス値に相当します。

“0”が標準の優先度であり、負の値は優先度を上げ、正の値は優先度を下げます。したがって、“-20”が最も優先度が高く、“19”が最も優先度が低い値となります。

記述例

```
dprty 4
```

4：ジョブ実行優先度

15.2.7 path文

記述形式

```
path パス名
```

機能説明

PATH環境変数に追加したいパスを指定します。

パスを複数指定する場合は、path制御文を複数指定してください。1つのpath制御文に複数のパスは指定できません。

path制御文で指定されたパスはPATH環境変数の先頭に追加されます。

path制御文はjobstart制御文から最初のstepstart文までの間に記述します。

オペランド

パス名

環境変数に追加したいパス名を指定します。

1つのpath制御文で指定するパス名の長さは、以下が259バイト以内になるように記述します。

・“path制御文に指定するパス名の長さ” ＋ “exec文に指定するプログラム名の長さ”

パスを複数指定する場合は、すべてのpath制御文を合わせたパス名の長さは以下が4096バイト以内になるように記述します。

・“ジョブ全体に関するpath制御文に指定するパス名の長さの合計” + “ジョブ全体に関するpath制御文の数”

なお、上記の長さは可変記号“&n”の情報を展開した後の長さになります。

記述例

【Windows版】

```
path C:¥win32app¥local
```

C:¥win32app¥local：実行プログラムが存在するパス

【UNIX版】

```
path /home/user1/bin
```

/home/user1/bin ： シェルスクリプト、またはコマンドが存在するパス

15.2.8 file文

記述形式

【Windows版】

```
file [ファイル名],ac（アクセス名） [,prt({プリンタ名｜*})
      [,prtform([fontname=フォント名] [,fontsize=フォントサイズ]
            [,orient= {port｜land}][,form=用紙サイズ])]]
```

【UNIX版】

```
file [ファイル名],ac（アクセス名） [,prt（{プリンタ名｜*}）]
```

機能説明

ジョブ全体で使用するファイル情報を指定します。

prtオペランドを指定すると印刷することができます。

オペランド

ファイル名

実行プログラムまたはバッチファイル【Windows版】/シェルスクリプト【UNIX版】で使用する入力ファイル/出力ファイルのファイル名を指定します。

指定したファイル名は、アクセス名と同名の環境変数に設定されます。ファイル名の一部に他の環境変数“%文字列%”【Windows版】/“$文字列”【UNIX版】が記述されていても置き換わりません。exec制御文に指定したシェルスクリプトは、環境変数からファイル名を取得できます。

本オペランドは、prtオペランドを指定した場合のみ省略できます。

本オペランドを省略した場合、ジョブ実行制御は、作業用一時ファイルを生成します。作業用一時ファイルには、システムで一意なファイル名が設定されます。なお、この作業用一時ファイルは、ファイル内容がプリンタに出力された後、ジョブがキューから削除される時に削除されます。

ac(アクセス名)

任意のアクセス名を指定します。指定したアクセス名と同名の環境変数にファイル名が設定されます。

アクセス名は、以下の条件を満たしている必要があります。

- 64バイト以下である
- 半角英数字および“_”(半角アンダーバー)のみで構成されている
- 先頭の1文字目は半角数字でない

prt({プリンタ名|*})

ジョブの終了後、本制御文の第1引数に指定したファイルの内容をプリンタに出力します。ただし、出力できる内容は、テキストファイルのみです。

プリンタ名を指定した場合は、指定したプリンタへ出力されます。指定するプリンタ名は以下の条件を満たしている必要があります。

- 32バイト以下の半角英数字である
- “&”、“(”および“)”を含まない

(*)を指定した場合は、jobstart文のprtオペランドで指定したプリンタに出力されます。

プリンタ名は本名の他にジョブ実行制御の[運用情報の定義]ウィンドウの[印刷形式]シートで定義した別名も指定することができます。【Windows版】

prtform【Windows版】

印刷形式に関する属性を指定します。サブオペランドは必ず1つ以上指定してください。サブオペランドの指定値には、Windowsのプリントマネージャで許されている値を指定します。

印刷形式に関する属性を指定した場合、以下のように印刷形式が決定されます。

- ジョブ実行制御の[運用情報の定義]ウィンドウの[印刷形式]シートに、file文のprtオペランドで指定したプリンタが定義されている場合

  [運用情報の定義]ウィンドウの[印刷形式]シートに定義した印刷形式と本オペランドのprtオペランドで指定した印刷形式がマージされて印刷されます。

- ジョブ実行制御の[運用情報の定義]ウィンドウの[印刷形式]シートに、file文のprtオペランドで指定したプリンタが定義されていない場合

  本オペランドで指定した印刷形式で印刷されます。

本オペランドを省略した場合は、以下のように印刷形式が決定されます。

- ジョブ実行制御の[運用情報の定義]ウィンドウの[印刷形式]シートに、file文のprtオペランドで指定したプリンタが定義されている場合

  [運用情報の定義]ウィンドウの[印刷形式]シートで定義した印刷形式で印刷されます。

- ジョブ実行制御の[運用情報の定義]ウィンドウの[印刷形式]シートに、file文のprtオペランドで指定したプリンタが定義されていない場合

  プリンタのデフォルト(Windowsでの定義)で印刷されます。

fontname=フォント名

フォント名を指定します。

本オペランドを省略した場合は、初期化パラメタ“prt”で指定したフォント名が有効となります。

fontsize=フォントサイズ

フォントサイズを指定します。単位は“ポイント”です。

本オペランドを省略した場合は、初期化パラメタ“prt”で指定したフォントサイズが有効となります。

orient={port|land}

印刷方向を指定します。縦に印刷する場合はport、横に印刷する場合はlandを指定します。

本オペランドを省略した場合は、初期化パラメタ“prt”で指定した印刷方向が有効となります。

form=用紙サイズ

用紙サイズを指定します。

本オペランドを省略した場合は、初期化パラメタ“prt”で指定した用紙サイズが有効となります。

記述例

【Windows版】

以下の記述例では、ファイル“d:¥result¥data1”をアクセス名“out1”という名前で割り当て(ジョブプロセスからは、“out1”という名前の環境変数を参照することで、ファイル名を獲得することができます)、ジョブ終了後“prt1”に出力することを指定しています。また、出力属性として、フォント名は「system」、フォントサイズは「10ポイント」、印刷方向は「縦」を指定しています。

```
file d:¥result¥data1,ac(out1),prt(prt1),
prtform(fontname=system,fontsize=10,orient=port)
```

【UNIX版】

ファイル“/result/data1”をアクセス名“output1”という名前で割り当て(ジョブプロセスからは、“output1”という名前の環境変数を参照することで、ファイル名を獲得することができます)、ジョブ終了後“printer1”に出力するように指定する場合の記述例を以下に示します。

```
file /result/data1,ac(output1),prt(printer1)
```

また、上記の条件でファイル名を省略する場合は、以下のように記述します。

```
file ,ac(output1),prt(printer1)
```

15.2.9 jobend文

記述形式

```
jobend
```

機能説明

ジョブの終了を意味する制御文です。この制御文以降の行に他のJCL制御文を記述することはできません。

本コマンドには、オペランドはありません。

記述例

```
jobend
```

15.3 ジョブステップに関する制御文

本節では、ジョブステップに関する制御文について説明します。

15.3.1 stepstart文

記述形式

```
stepstart ジョブステップ名 { [,skip （cc {＜｜＝ ｜＞} code
                    [, {jobend｜ジョブステップ名} ] ) ]
                    ｜[, {even｜only｜every} ] }
```

機能説明

ジョブステップの開始を意味し、本制御文から次に記述するstepend制御文までが1つのジョブステップになります。

当該ジョブステップの終了コードにより、次に実行するジョブステップを指定することができます。

1つのJCLファイルに記述できるジョブステップの数は最大186個です。

オペランド

ジョブステップ名

ジョブステップ名を指定します。

ジョブステップ名は最大16バイトまでの英数字で指定できます。

skip(cc{<|=|>}code[,{jobend|ジョブステップ名}])

当該ジョブステップの終了コードにより、次に実行するジョブステップを指定します。指定した条件(cc{<|=|>}code)が成立すると、分岐先ジョブステップ({jobend|ジョブステップ名})で指定したジョブステップへ制御が移り、JCL上当該ジョブステップの次に位置するジョブステップは実行されません。

skip文は、32個まで同時に指定できます。

条件(cc{<|=|>}code)

ccは、固定の文字列で当該ジョブステップの終了コードを示します。

codeには、終了コードの値を0～256の整数で指定します。

ccがcodeより小さい値のときに終了させたい場合は“cc<code”、ccとcodeが同じ値のときに終了させたい場合は“cc=code”、ccがcodeより大きい値のときに終了させたい場合は“cc>code”という形式で指定します。

分岐先ジョブステップ({jobend|ジョブステップ名})

指定した条件が成立した場合、どのジョブステップに制御を移すかをステップ名または、固定の文字列“jobend”で指定します。

ジョブを終了させる場合、“jobend”を指定するか、または本指定を省略します。

当該ジョブステップよりも後ろに存在するステップ名以外を指定した場合は、エラーとなり、ジョブは実行されません。

指定したジョブステップに制御が移った後の制御は、通常どおり上から下へジョブステップを実行します。この際、codeの値は継承されます。

even|only|every

even、onlyまたは、everyのいずれかを指定します。

本オペランドを指定したジョブステップは、必ずジョブの最終ジョブステップとして指定します。

また、ジョブステップが1ステップのみの場合に指定された場合は、本オペランドは無視されます。

- onlyを指定した場合：

  先行したジョブステップのどれかでアプリケーションエラーが発生した場合だけ、当該ジョブステップが実行されます。

- evenを指定した場合：

  先行したジョブステップの終了状態にかかわらず、当該ジョブステップは必ず実行されます。

ただし、先行ジョブステップ実行中に強制終了操作された場合やサービス/デーモンの停止依頼があった場合は、当該ジョブステップは実行されません。

- everyを指定した場合：

  ジョブの終了時に(正常終了、異常終了、強制終了のいずれの場合も)、当該ジョブステップは必ず実行されます。ただし、サービス/デーモンの停止時には、当該ジョブステップは呼び出されません。なお、当該ジョブステップの実行中は、強制終了を受け付けません。また、当該ジョブステップの実行中にサービス/デーモンの停止依頼があった場合、当該ジョブステップは中断します。

注意事項

例外コードで異常終了した場合をアプリケーションエラーとしています。

記述例

当該ジョブステップ“step1”が、0より大きい値で終了した場合は、ジョブステップ“recovery”へ制御を移す場合の記述例を以下に示します。

```
stepstart step1,skip(cc>0,recovery)
```

15.3.2 path文

記述形式

```
path パス名
```

機能説明

PATH環境変数に追加したいパスを指定します。

パスを複数指定する場合は、path制御文を複数指定してください。1つのpath制御文に複数のパスは指定できません。

path制御文で指定されたパスはPATH環境変数の先頭に追加されます。

path制御文はstepstart制御文からstepend文までの間に記述します。

ジョブ全体に関する制御文においてpath文を指定した場合は、本path文で指定したパスの後ろに追加されます。

オペランド

パス名

環境変数に追加したいパス名を指定します。

１つのpath制御文で指定するパス名の長さは、以下が259バイト以内になるように記述します。

- “path制御文に指定するパス名の長さ” ＋ “exec文に指定するプログラム名の長さ”

パスを複数指定する場合は、すべてのpath制御文を合わせたパス名の長さは以下が4096バイト以内になるように記述します。

- “ジョブステップに関するpath制御文に指定するパス名の長さの合計” ＋ “ジョブステップに関するpath制御文の数”

なお、上記の長さは可変記号“&n”の情報を展開した後の長さになります。

記述例

【Windows版】

```
path c:¥users¥user1¥bin
```

c:¥users¥user1¥bin：実行プログラムが存在するパス

【UNIX版】

```
path /users/user1/bin
```

/users/user1/bin：シェルスクリプト、またはコマンドが存在するパス

15.3.3 file文

記述形式

【Windows版】

```
file {[ファイル名] ,ac(アクセス名)｜* ,ac(アクセス名),dlm(区切り文字)}
      [,prt({プリンタ名｜* })
     [,prtform([fontname=フォント名] [,fontsize=フォントサイズ]
           [,orient= {port｜land} ][,form=用紙サイズ])]]
```

【UNIX版】

```
file {[ファイル名] ,ac(アクセス名)｜*,ac(アクセス名),dlm(区切り文字)}
   [,prt( {プリンタ名｜*} )]
```

機能説明

該当ステップで実行されるプログラムが使用するファイルを指定します。

prtオペランドを指定すると印刷を行うことができます。

オペランド

[ファイル名],ac(アクセス名)

ファイル名

実行プログラムまたはバッチファイル【Windows版】/シェルスクリプト【UNIX版】で使用する入力ファイル/出力ファイルのファイル名を指定します。

指定したファイル名は、アクセス名と同名の環境変数に設定されます。ファイル名の一部に他の環境変数“%文字列%”【Windows版】/“$文字列”【UNIX版】が記述されていても置き換わりません。exec制御文に指定した実行プログラムまたはバッチファイル【Windows版】/シェルスクリプト【UNIX版】は、環境変数からファイル名を取得できます。

本オペランドは、prtオペランドを指定した場合のみ省略できます。

本オペランドを省略した場合、ジョブ実行制御は、作業用一時ファイルを生成します。作業用一時ファイルには、システムで一意なファイル名が設定されます。なお、この作業用一時ファイルは、ファイル内容がプリンタに出力された後、ジョブがキューから削除される時に削除されます。

ac(アクセス名)

任意のアクセス名を指定します。アクセス名は、以下の条件を満たしている必要があります。

- 64バイト以下である
- 半角英数字および“_”(半角アンダーバー)のみで構成されている
- 先頭の1文字目は半角数字でない

ネットワークジョブを投入する場合は、前述の“ファイル名”にネットワークジョブを制御するコマンド列を格納したファイル名を指定し、アクセス名として“mjsin”を指定します。

ネットワークジョブを投入するジョブステップの構成については、“図15.1 ネットワークジョブのジョブステップ構成”を参照してください。

*,ac(アクセス名),dlm(区切り文字)

*

“*”を指定した場合、本制御文の次の行から区切り文字が現れる直前の行までが、ジョブ実行制御によって生成される一時ファイルに保存されます。

“*”を指定した本制御文は、実行プログラムまたはバッチファイル【Windows版】/シェルスクリプト【UNIX版】が入力パラメタをファイルから取得する場合に使用します。

入力パラメタをJCL中に記述することができるため、入力パラメタを記述したファイルを別途作成する必要がありません。

dlm(区切り文字)

JCL中に展開するデータの終了行を意味する任意の文字列を指定します。本オペランドで指定した区切り文字列が現れる直前の行までが、ジョブ実行制御によって生成される一時ファイルに保存されます。

区切り文字は、行の先頭から指定してください。空白などの区切り文字以外から始まる行は、データの最終行として扱われません。

区切り文字は、最大8バイトまでの英数字で指定します。英字の大文字と小文字は区別されます。

prt{プリンタ名|*}

ジョブの終了後、本制御文の第1引数に指定したファイルの内容をプリンタに出力します。ただし、出力できる内容は、テキストファイルのみです。

プリンタ名を指定した場合は、指定したプリンタへ出力されます。指定するプリンタ名は以下の条件を満たしている必要があります。

- 32バイト以下の半角英数字である
- “&”、“(”および“)”を含まない

(*)を指定した場合は、jobstart文のprtオペランドで指定したプリンタに出力されます。

プリンタ名は本名の他にジョブ実行制御の[運用情報の定義]ウィンドウの[印刷形式]シートで定義した別名も指定することができます【Windows版】

prtform【Windows版】

印刷形式に関する属性を指定します。サブオペランドは必ず1つ以上指定してください。サブオペランドの指定値には、プリントマネージャで許されている値を指定します。

印刷形式に関する属性を指定した場合、以下のように印刷形式が決定されます。

- ジョブ実行制御機能の運用情報ファイルのprtオペランドに、file文のprtオペランドで指定したプリンタが定義されている場合

  運用情報ファイルの該当するprtオペランドに定義した印刷形式と本オペランドのprtオペランドで指定した印刷形式がマージされて印刷されます。

- ジョブ実行制御機能の運用情報ファイルのprtオペランドに、file文のprtオペランドで指定したプリンタが定義されていない場合

  本オペランドで指定した印刷形式で印刷されます。

本オペランドを省略した場合は、以下のように印刷形式が決定されます。

- ジョブ実行制御機能の運用情報ファイルのprtオペランドに、file文のprtオペランドで指定したプリンタが定義されている場合

  運用情報ファイルの該当するprtオペランドに定義した印刷形式で印刷されます。

・ ジョブ実行制御機能の運用情報ファイルのprtオペランドに、file文のprtオペランドで指定したプリンタが定義されていない場合

プリンタのデフォルト(Windowsでの定義)で印刷されます。

fontname=フォント名

フォント名を指定します。

本オペランドを省略した場合は、初期化パラメタ“prt”で指定したフォント名が有効となります。

fontsize=フォントサイズ

フォントサイズを指定します。単位は“ポイント”です。

本オペランドを省略した場合は、初期化パラメタ“prt”で指定したフォントサイズが有効となります。

orient={port|land}

印刷方向を指定します。縦に印刷する場合はport、横に印刷する場合はland を指定します。

本オペランドを省略した場合は、初期化パラメタ“prt”で指定した印刷方向が有効となります。

form=用紙サイズ

用紙サイズを指定します。

本オペランドを省略した場合は、初期化パラメタ“prt”で指定した用紙サイズが有効となります。

記述例

【Windows版】

以下の記述例では、ファイル“d:¥result¥data1”をアクセス名“out1”という名前で割り当て(ジョブプロセスからは、“out1”という名前の環境変数を参照することで、ファイル名を獲得することができます)、ジョブ終了後“prt1”に出力することを指定しています。また、出力属性として、フォント名は「system」、フォントサイズは「10ポイント」、印刷方向は「縦」を指定しています。

```
file d:¥result¥data1,ac(out1),prt(prt1),
prtform(fontname=system,fontsize=10,orient=port)
```

【UNIX版】

ファイル“/result/data1”をアクセス名“output1”という名前で割り当て(ジョブプロセスからは、“output1”という名前の環境変数を参照することで、ファイル名を獲得することができます)、ジョブ終了後“printer1”に出力することを指定する場合の記述例を以下に示します。

```
file /result/data1,ac(output1),prt(printer1)
```

また、上記の条件でファイル名を省略する場合は、以下のように記述します。

```
file ,ac(output1),prt(printer1)
```

ファイル名に“*”を指定することで、JCL中に任意のデータをインライン展開することを指定しています。インライン展開の終了を示す区切り文字列は“ZZZ”を使用している場合の記述例を以下に示します。

```
file *,ac(indata1),dlm(ZZZ)
・・・・
・・・・
・・・・
ZZZ
```

・・・・・： 任意のデータ

15.3.3.1 connectコマンド

記述形式

```
connect host（ホスト名）
```

機能説明

connectコマンドは、ジョブの実行を依頼する他サーバとの回線を接続します。

オペランド

ホスト名

ジョブの実行を依頼する他サーバのホスト名を指定します。ホスト名は、ネットワークセットアップのTCP/IP構成で定義したホスト名です。

構成の変更を特に行っていない場合は、コンピュータ名を小文字に変換した文字列がホスト名です。【Windows版】

記述例

```
connect host(host1)
```

host1 ：回線を接続する他サーバのホスト名

15.3.3.2 sendコマンド

記述形式

```
send アクセス名 [,file(送信先ファイル名) ]
```

機能説明

sendコマンドは、ジョブを投入したサーバ上のファイルを、connectコマンドで指定した他サーバに送信します。

オペランド

アクセス名

送信するファイルを指定したfile文のアクセス名を指定します。JCLに記述されていないアクセス名を指定した場合は、エラーとなり、ジョブは終了します。

file(送信先ファイル名)

送信先のファイル名をフルパスで指定します。本オペランドを指定すると指定された名前でファイルが作成され、ジョブの実行が終了した後もファイルは削除されません。

本オペランドを省略した場合は,一時ファイルとして送信され、ジョブからのアクセスは通常と同様(アクセス名と環境変数に一時ファイル名が設定される)に扱えます。ジョブの実行が終了すると一時ファイルは削除されます。

記述例

【Windows版】

アクセス名“data1”が定義されているfile文で指定されたファイルを、fileオペランドで指定した“d:¥userinfo¥data1”へ送信する場合の記述例を以下に示します。

```
send data1,file(d:¥userinfo¥data1)
```

【UNIX版】

アクセス名“data1”が定義されているfile文で指定されたファイルを、fileオペランドで指定した“/userinfo/data1”へ送信する場合の記述例を以下に示します。

```
send data1,file(/userinfo/data1)
```

15.3.3.3 receiveコマンド

記述形式

```
receive アクセス名 [,file(受信元ファイル名) ]
```

機能説明

receiveコマンドは、connectコマンドで指定した他サーバ上のファイルを、ジョブを投入した自サーバで受信します。

オペランド

アクセス名

受信するファイルを指定したfile文のアクセス名を指定します。JCLに記述されていないアクセス名を指定した場合は、エラーとなり、ジョブは終了します。

file(受信元ファイル名)

connectコマンドで接続したホスト上に存在するファイル名をフルパスで指定します。指定したファイルが受信されます。

本オペランドを省略した場合は,アクセス名に対応したファイルが受信されます。

記述例

【Windows版】

fileオペランドで指定した“d:¥userinfo¥list1”から、アクセス名“list1”が定義されているfile文で指定されたファイルを受信する場合の記述例を以下に示します。

```
receive list1,file(d:¥userinfo¥list1)
```

【UNIX版】

fileオペランドで指定した“/userinfo/list1”から、アクセス名“list1”が定義されているfile文で指定されたファイルを受信する場合の記述例を以下に示します。

```
receive list1,file(/userinfo/list1)
```

15.3.3.4 executeコマンド

記述形式

```
execute (実行ファイル名 実行オプション) [,queue（キュー名)]
```

機能説明

executeコマンドは、オペランドで指定された内容に従って、connectコマンドで指定した他サーバ上でジョブを実行します。

オペランド

実行ファイル名 実行オプション

他サーバで実行する実行ファイル名および実行オプションを指定します。

ジョブファイル名にqsubコマンドを指定することで、突放し形態のネットワークジョブの依頼ができます。なお、実行ファイル名と実行オプションの間に1つ以上の空白を置きます。

queue(キュー名)

他サーバへ依頼するジョブを、どのキューで実行するかを指定します。

本オペランドを省略した場合は、以下の手順でキューが決定されます。

1. 投入元ジョブの実行キュー(投入時に指定したキュー名、または省略値)と同一名のキュー名。
2. 同一のキュー名が存在しない場合は、ジョブを依頼する他サーバで[運用情報の定義]ウィンドウの“ジョブ投入先のキューの省略値を指定する”(systemパラメタのdfltqueueオペランド)に定義されたキュー名。
3. [運用情報の定義]ウィンドウで“ジョブ投入先のキューの省略値を指定する”(systemパラメタのdfltqueueオペランド)が指定されていない場合は、ジョブの投入に失敗します。

記述例

アプリケーション“appl1.exe”を、キュー“queue1”で実行する場合の記述例を以下に示します。

```
execute (appl1.exe),queue(queue1)
```

15.3.3.5 disconnectコマンド

記述形式

```
disconnect
```

機能説明

disconnectコマンドは、connectコマンドで接続した他サーバとの回線を切断します。

本コマンドは、省略できます。省略した場合、他サーバとの回線は、ジョブステップ終了時または、次のconnectコマンド実行時に自動切断されます。

なお、接続されていない状態でdisconnectコマンドを実行した場合、本コマンドは無視され処理が続行されます。

本コマンドには、オペランドはありません。

15.3.4 exec文

記述形式

```
exec プログラム名 [引数…]
```

機能説明

実行するプログラムやバッチファイル名【Windows版】/シェルスクリプト名【UNIX版】と引数を指定します。

オペランド

プログラム名

実行するプログラム名を指定します。バッチファイル名【Windows版】/シェルスクリプト名【UNIX版】を指定することもできます。

引数…

実行するプログラムに引数を渡す場合、プログラム名と引数の間に1つ以上の空白を置きます。

引数は空白で区切ることにより、複数個指定することができます。

引数に空白を含む場合などは、引数を一対の引用符(''または"")で囲んでください。

引数として&nを指定した場合、qsubコマンドで指定した引数がプログラムに渡ります。この時のnは、qsubコマンドで指定した引数の何個目かを指定します。

ネットワークジョブを実行する場合

ネットワークジョブを投入する場合は、プログラム名および引数を以下の形式で指定します。

```
mjsrjob.exe [-r リトライ回数]  (注)
```

注)

mjsrjob.exe は、ファイル名のみ記述してください(フルパスで記述する必要はありません)。

-r リトライ回数

ネットワークジョブの実行依頼中(コマンドconnect,sendおよびexecuteの実行中)に回線異常が発生した場合にリトライを行います。この時のリトライ回数を、1～20の範囲で指定します。

なお、executeコマンド処理中については、ジョブの実行依頼処理が終了した後の、ジョブの実行完了を待っている状態において回線異常が発生した場合は、リトライは行われません。

ジョブ投入時に、[Systemwalker Operation Manager]ウィンドウ([ジョブ情報編集/投入]ウィンドウ)で拡張情報としてリトライ処理を指定した場合、または、qsubコマンドでリトライオプションを指定した場合は、それらの指定が優先されます。

ポイント

mjsrjob.exeの復帰値は、ネットワークジョブ制御用ファイル内においてexecuteコマンドで指定した実行ファイルの復帰値になります。

記述例

【Windows版】

```
exec appl01.exe gyoumu1 &1 &2 &3
```

appl01.exe：実行プログラム名

【UNIX版】

```
exec script1 gyoumu1 &1 &2 &3
```

script1：シェルスクリプト名

gyoumu1：第1引数
&1：qsubコマンドで指定した第1引数
&2：qsubコマンドで指定した第2引数
&3：qsubコマンドで指定した第3引数

15.3.5　stepend文

記述形式

```
stepend
```

機能説明

ジョブステップの終了を指定します。

本コマンドには、オペランドはありません。

記述例

```
stepend
```

15.4 JCLとプログラムの関係

本節では、JCLとプログラムの関係について説明します。

15.4.1 path文とプログラムの関係

path文は、ジョブ全体に関する制御文および、ジョブステップに関する制御文に存在し、プログラムが存在するパスを指定することができます。

プログラムの検索は、ジョブステップにおけるpath文から先に検索されます。ジョブステップおよび、ジョブにおけるpath文に複数のパスが指定された場合、指定順に検索されます。

以下に検索順の例を示します。

【Windows版】

```
jobstart s7025job
path c:¥system¥appl -------- 3
path c:¥system¥applx -------- 4

stepstart step1
path c:¥user¥appl -------- 1
path c:¥user¥applx -------- 2
```

【UNIX版】

```
jobstart    s7025job
path /usr/local/bin      ----------  3
path /home/s7025/bin     ----------  4

stepstart step1
path /home/s7025/bin1    ----------  1
path /home/s7025/bin2    ----------  2
```

15.4.2 file文とプログラムの関係

プログラムにおいてファイルを扱う際、JCLで定義したアクセス名を利用することができます。

アクセス名と同名の環境変数にJCLで定義したファイル名が格納されます。すなわち、任意の環境変数にファイル名を設定したことになります。



注意

file文のアクセス名に指定したものと同名の環境変数を、シェルのスタートアップファイル(“.profile”、“.cshrc”、“.login”など)に設定した場合、シェルのスタートアップファイルの設定が有効になるので注意してください(file文→シェルのスタートアップファイルの順に設定されるため)。

15.4.3　exec文とプログラムの関係

exec文に指定された名前のプログラムを起動します。プログラムの後ろに引数が指定されている場合、プログラムに引数を渡します。

JCLにおいて“exec appl.exe host1 s7025”とした場合、プログラムの引数は、以下のようになります。

第1引数:

“host1”

第2引数:

“s7025”

15.5　JCL・ジョブステップの終了コード

本節では、JCL・ジョブステップの終了コードについて説明します。

15.5.1　JCLの終了コード

JCLに記述したジョブステップの中で最も大きいジョブステップの終了コードが、JCLの終了コードになります。ただし、ジョブの実行が中断された場合、JCLの終了コードは、特別な値になります。

また、[運用情報の定義]ウィンドウの[利用機能]シートで、最後に実行されたジョブステップの終了コードをJCLの終了コードとするよう指定できます。

ジョブの実行が中断された場合の終了コードについては、“15.5.2 ジョブステップの終了コード”を参照してください。

15.5.2　ジョブステップの終了コード

ジョブステップ内のexec制御文に指定したプログラムの復帰コードが、そのジョブステップの終了コードになります。stepstart制御文の実行条件によって実行されなかったジョブステップには、終了コードがありません。

また、[運用情報の定義]ウィンドウの[利用機能]シートで、ジョブリストにジョブステップの区切り(ジョブステップのステップ名および終了コード)を標準出力するように設定できます。

第16章　Systemwalkerスクリプト

本章では、Systemwalker Operation Managerが提供するSystemwalkerスクリプト(ジョブ制御言語)について説明します。

16.1　Systemwalkerスクリプトの概要

本節では、Systemwalkerスクリプトの概要について説明します。

16.1.1　Systemwalkerスクリプトの機能

Systemwalkerスクリプトは、Systemwalker Operation Managerが提供する新たなジョブ制御言語です。

Systemwalkerスクリプトは、Tcl(Tool Command Language)/Tk(Tool kit)をベースとしているため、動作するプラットフォームに依存しません。

Systemwalkerスクリプトは、ジョブとして扱われ実行されます。

本章で説明される文法・コマンドを使用して、任意のジョブをSystemwalkerスクリプトで作成することができます。また、Systemwalkerスクリプトのデバッグを行うためのスクリプト実行用コマンドとトレース用コマンドが用意されています。

16.1.2　Systemwalkerスクリプトの構成

ファイル名

拡張子が“swt”となっているファイルを、Systemwalkerスクリプトとして扱います。

```
～.swt
```

構成

Systemwalkerスクリプトは、コマンド文の集まりで構成されています。スクリプトの記述内容は、本章の文法およびコマンドの説明の範囲内で任意です。コマンド文の記述形式は、以下のとおりです。

```
コマンド オプション引数
```

スクリプトファイル内は、以下のようにコマンド文の集まりになっています。

```
コマンド文
…
…
…
…
```

起動引数

スクリプトの起動時に引数として、スクリプト内部で処理させたい情報を与えることができます。スクリプトファイル名とともに指定した起動引数は、以下の予約変数を使用することでスクリプト内において扱うことができます。

argc:　引数の個数が格納されます。

argv:　引数の内容が格納されます。

引数の参照は、作業用変数に引数を分解して取り出した上で行います。

例：

第1引数を作業用変数“prm1”に取り出します。

```
set prm1 [lindex $argv 0]
```

説明：

set:　変数に値を格納するコマンド

prm1:　値を格納する変数名

lindex:　起動引数を取り出すコマンド

argv:　起動引数が格納された変数($を先頭に付けることで変数の中身を表します)

0:　(取り出す引数の位置－1)の数値を指定

環境変数

環境変数は、予約変数“env”にあらかじめ格納されています。

スクリプト内では、“env”の直後に個々の環境変数名を“()”(小括弧)で指定した変数名として扱い、環境変数の参照・更新は、env(環境変数名)に対して行います。

例：

環境変数“PATH”の参照・更新に使用する変数名

```
env(PATH)
```

文字コード

スクリプトファイルの文字コードは、サーバと同じ文字コードにします。なお、Windows版の場合、SJISコードで格納してください。

注意事項

Windows版のSystemwalkerスクリプトで、JIS X 0213:2004で新規に追加された文字を使用することはできません。

16.1.3　Systemwalkerスクリプトの文法

Systemwalkerスクリプトの文法について説明します。

変数

変数名は、アルファベットで始まる任意の長さの英数字が使用できます。大文字と小文字は区別され、また“_”(アンダーバー)を含むことができます。

構文

Systemwalkerスクリプト言語の構文とその解釈は以下のように定義されています。

コマンドの区切り

スクリプト行は、“;”(セミコロン)または改行がコマンドの区切りとなります。

例：変数abc,defにそれぞれ値を格納する場合

```
set abc ABC
set def DEF
```

または

```
set abc ABC ; set def DEF
```

“$”(ダラー記号)による変数置換

変数置換は、“$”(ダラー記号)を使用します。変数置換とは、変数に格納した値の参照をすることです。値が未設定の変数に対して変数置換を行うとスクリプトエラーとなり、異常終了します。

例：変数abcに格納した値を別の変数defに格納する場合

```
set abc ABC
set def $abc
```

“[ ]”(大括弧)によるコマンド置換

スクリプトのコマンドには、復帰値を返すものがあります。コマンド置換とは、呼出し元におけるコマンド呼出しを行った場合の復帰値の参照に当たります。コマンド置換は、“[ ]”(大括弧)を使用します。

例：算術演算(3+2)の結果を変数abcに格納する場合

```
set abc [ expr 3 + 2 ]
```

コマンド内のワードの区切り

コマンド内のワードの区切りは、スペースまたはタブを使用します。ワードの区切りとは、「コマンド名、第1引数、第2引数…」の区切りのことを指します。

ワード内の変数置換・コマンド置換

ワード内に“$”(ダラー記号)が含まれる場合、変数置換が行われ変数の内容に置き換えられます。ワード内に“[]”(大括弧)が含まれる場合は、コマンド置換が行われ、コマンド実行結果の復帰値に置き換えられます。

コマンド置換は、再帰的に行われますが、変数置換は一度しか行われません。

例：再帰的にコマンド置換を行い、計算結果を格納する場合

```
set abc [ expr [expr 1 + 1] + 2 ]
```

注意：

exprコマンドは、入れ子の計算式も解釈可能なため、算術演算だけの場合は通常は、例で示したような書き方はせず、以下の記述をします。

```
set abc [ expr (1 + 1) + 2 ]
```

「"」(ダブルクォーテーション)によるワードの指定

ワードの最初の文字が「"」(ダブルクォーテーション)の場合は、次のダブルクォーテーションまでが1つのワードとして扱われます。ワードの最初の文字がダブルクォーテーションでそれに対応する終了ダブルクォーテーションがない場合は、文法エラーとなります。また、開始と終了のダブルクォーテーション自体はワードの内容としては解釈されません。

例：文字列123を変数abcに格納する場合

```
set abc "123"
```

ダブルクォーテーションは、直前に“¥”(円記号)がある場合、ワード囲み記号として解釈されず、通常の文字として扱われます。円記号置換については、“¥”(円記号)によるエスケープシーケンス(円記号置換)”を参照してください。

例：文字列123"を変数abcへ格納する場合

```
set abc "123¥""
```

ワード指定のダブルクォーテーションの中に含まれる“;”(セミコロン)、“]”(閉じ括弧)、スペース(改行も含む)は、普通の文字として扱われますが、コマンド置換・変数置換は処理されます。

例：空白・変数置換を含む文字列を格納する場合

```
set abc "123"
set str "abc $abc def"
```

この場合、変数strへ格納される文字列は abc 123 def となります。

“{ }”(中括弧)によるワード指定

ワードの最初の文字が“{”(開き中括弧)の場合は、対応する“}”(閉じ中括弧)までがワードになります。ワードの最初の文字が開き中括弧でそれに対応する閉じ中括弧がない場合は、文法エラーとなります。また、開始と終了の中括弧自体はワードの内容としては解釈されません。

例：文字列1{2}3を変数abcへ格納する場合

```
set abc {1{2}3}
```

中括弧は、直前に“¥”(円記号)がある場合、ワード囲み記号として解釈されません。ただし、中括弧によるワード囲み内では、“¥{”，“¥}”を通常文字の“{”，“}”に置き換えることはありません。詳細については、“ワード囲み記号と円記号置換”を参照してください。

例：中括弧を含む文字列を格納する場合

```
set abc {123¥}}
```

この場合、変数abcへは文字列 123¥}が格納されます。

ワード指定の中括弧囲みの中に含まれる“;”(セミコロン)、“]”(閉じ括弧)、スペース(改行も含む)は、普通の文字として扱われ、コマンド置換・変数置換は処理されません。

例：空白を含む文字列を格納する場合

```
set abc ”123”
set str {abc $abc def}
```

この場合、変数strへ格納される文字列は abc $abc def となります。

“¥”(円記号)によるエスケープシーケンス(円記号置換)

ワードに“¥”(円記号)が含まれている場合は、その直後の文字との組み合わせをエスケープシーケンスとして解釈され、円記号置換が行われます。

円記号置換の種類を以下に示します。

シーケンス	置換内容
¥a	ベル
¥b	バックスペース
¥f	改頁
¥n	改行
¥r	キャリッジ・リターン
¥t	タブ
¥v	垂直タブ
¥<改行>	空白文字(継続行)
¥”	ワード囲み記号として解釈されない単独ダブルクォーテーション
¥{	ワード囲み記号として解釈されない単独中括弧開き記号
¥}	ワード囲み記号として解釈されない単独中括弧閉じ記号
¥¥	単独円記号“¥”

注意：円記号の直後に上記以外の文字を指定することはできません。

ワード囲み記号と円記号置換

ダブルクォーテーションで囲まれたワードに対しては、すべての円記号置換が行われますが、中括弧で囲まれたワードに対しては“¥<改行>”を除き、円記号置換は行われません。

例：中括弧内で円記号置換されない例

```
set abc ” C:¥¥tmp”
set def {C:¥tmp}
```

この場合、変数abc,defともに文字列 C:¥tmp が格納されます。

例：中括弧内でも円記号置換される例

```
set abc ” 123¥
456”
set def {123¥
456}
```

この場合、変数abc,defともに文字列 123 456 が格納されます。

中括弧によるワード囲み内では、“¥{”，“¥}”は円記号置換されません。また、“¥¥”も円記号置換されませんが、後の¥は“¥<改行>”、“¥{”、“¥}”でのエスケープ文字としての意味を失います。

例：“¥”がエスケープ文字としての意味を失う場合の例

```
set abc {123¥¥}
```

この場合、変数abcには文字列 123¥¥ が格納されます。



注意

これまでの説明から、以下の文字列を中括弧で囲んだワードとして記述することはできません。

例：末尾に円記号がある文字列

```
C:¥tmp¥
```

例：対応のとれていない中括弧が存在する文字列

```
123}
```

例：改行の直前に円記号がある文字列

```
C:¥tmp¥
D:¥tmp¥
```

条件式における文字列の扱い

ifやwhileなどの判定文で指定する条件式は、通常は数値式として解釈されます。このため、条件式を文字列式として解釈させたい場合は文字列を「”」(ダブルクォーテーション)で囲む必要があります。真偽値の意味を持つ特別な文字列(yes,no,true,false)を指定する場合においても同様です。

例1：変数aの内容が文字列xyzであるかどうか判定する場合

```
if {$a == ” xyz”} {
    puts ok
}
```

例2：真である間処理を繰り返す場合

```
set i 0
while {”true”} {
    incr i
    if {$i > 5} {
        break
    }
}
```

文字列操作コマンド

文字列をダブルクォーテーションで囲んでも、文字列全体が数値として解釈可能な場合は、数値式として判定されます。厳密な文字列比較を行うためには、文字列操作用の16.2.19 string(文字列を操作する)コマンドを使用して判定を行うことを推奨します。

例：変数aの内容が文字列 01 であるかどうか判定する場合

```
if {[string compare $a 01] == 0} {
    puts ok
}
```

上記で、判定文を以下のようにすると、変数aの内容が1や001の場合も真となります。

```
if {$a == ”01”}
```

コメント行

#で始まる行およびコマンドの最初が#の場合は、改行までをコメントと解釈します。

例：

```
# 変数を初期化する
set abc 0
```

注意

コメントを記述する場合、以下の点に注意してください。

- コメント行内には、“{”(開き中括弧)、“}”(閉じ中括弧)を使用しないでください。
- コメント行、およびコメント行の直前の行で、継続行(¥<改行>)は使用しないでください。
- ifやwhileなどの判定文で指定する条件式内には、コメントを記述しないでください。

16.1.4 正規表現

スクリプト内では、正規表現を使用することができます。以下にスクリプトにおける正規表現について説明します。

正規表現の記号

.	任意の1文字に一致する。 例：regexp {ABC.} $text 解説：文字中に“ABC”とそれに続く任意の1文字からなる4文字の文字列があるか検査する。
*	直前のパターン要素の0回以上の繰り返しに一致する。 例：regexp {<.*>} $text 解説：文字列中に“<”と“>”で囲まれた部分があるか検査する。(囲まれた中は空文字でも可)

+	直前のパターン要素の1回以上の繰り返しに一致する。 例：regexp {<.+>} $text 解説：文字列中に“<”と“>”で囲まれた文字があるか検査する。(囲まれた中には任意の文字が1文字以上あること)
?	直前のパターン要素の0回または1回の出現に一致する。 例：regexp {0?1} $text 解説：文字列中に“1”または“01”があるか検査する。
( )	サブパターンをグループ化する。繰り返しや選択は、サブパターン全体に適用される。 例：regexp {(.*),(.*)} $text all sub1 sub2 解説：文字列中にコンマがあるか検査し、ある場合は一致部分すべてをallに、コンマの前後をそれぞれsub1,sub2に切り出す。(コンマの前後は空文字列でも可)
|	選択 例：regexp {Info|Warning} $text 解説：文字列中に“Info”または“Warning”が含まれるか検査する。
[ ]	文字集合に含まれる文字に一致する。 ただし、最初の文字が“^”(ハット)である場合は、文字集合に含まれない文字に一致する。範囲は[a-z]のように“-”(ハイフン)を入れる。 例：regexp {ABC[0-9][0-9][^]} $text 解説：文字列中に“ABC”に続く2桁の数字があり、さらにその直後が空白以外であるか検査する。
^	パターンの先頭に指定された場合、文字列の先頭に一致する。 例：regexp{^ABC} $text 解説：文字列の先頭が“ABC”であるか検査する。
$	パターンの末尾に指定された場合、文字列の末尾に一致する。 例：regexp{ABC$} $text 解説：文字列の末尾が“ABC”であるか検査する。

パターン検索の規則

- パターンと一致した最初で最長の文字列が選択されます。
- “|”(選択)が含まれる場合は、左側のサブパターンから検査されます。
- “*”，“+”，“?”を使用した表現では、一致する部分が長い文字列が優先されます。
- 複数の部分表現が並んでいる場合は、左側の部分表現から検索します。

特殊文字と正規表現記号の扱い

正規表現記号のうち、スクリプト上の特殊文字に当たるものがあるため、それらを使用する場合には以下のように記述してください。

スクリプト上の特殊文字

`[, ], $, (, ), {, }`

特殊文字の使用方法

以下のどちらかの方法で記述します。

- 前に“¥”(円記号)を付けてエスケープする
- 正規表現全体を“{ }”(中括弧)で囲む

例：

regexpの引数に正規表現の“[0-9]”を指定します。

```
regexp ”¥[0-9¥]” $text
```

または、

```
regexp {[0-9]} $text
```

また、正規表現記号を正規表現の中で単なる文字として扱う場合は、“¥”(円記号)を直前に付加してください。ただし、円記号はスクリプトとしての特殊文字でもあるため、記述する際には“¥¥”とするか正規表現全体を中括弧で囲む必要があります。

例1：

文字“*”を含んでいるかを検査します。

```
regexp ”¥¥*” $text
```

または、

```
regexp {¥*} $text
```

例2：

文字“[”を含んでいるかを検査します。

```
regexp “¥¥¥[” $text
```

または、

```
regexp {¥[} $text
```

また、これにともない正規表現の中で円記号自身を単なる文字として扱わせる場合は、“¥¥”と記述する必要があります。この場合も円記号がスクリプトとしての特殊文字であることを考慮し、2つの円記号それぞれを“¥¥”とするか、正規表現全体を中括弧で囲む必要があります。

例3：

文字“¥”を含んでいるかを検査します。

```
regexp ”¥¥¥¥” $text
```

または、

```
regexp {¥¥} $text
```

参考

スクリプトの特殊文字を円記号でエスケープすると、例2,3のように複数の円記号が必要になり間違いやすいため、正規表現の記述は常に中括弧で囲むことを推奨します。

ポイント

文字列を切り出すには[^]を組み合わせて記述します。

文字列"A<BC>D"から"< >"で括られている"BC"部分を変数Var0に切り出す場合は、以下のように記述します。

```
regexp {<([^>]*)} "A<BC>D" All Var0
```

文字列"A[BC]D"から"[ ]"で括られている"BC"部分を切り出す場合、特殊記号の"[ ]"の直前に円記号"¥"を付加してエスケープします。

```
regexp {¥[([^¥]]*)} "A¥[BC¥]D" All Var0
```

正規表現の一般的な文法エラーについて

正規表現の一般的な文法エラーとその対処法の例を示します。

- couldn't compile regular expression pattern: parentheses () not balanced

  正規表現の記述内で括弧“( )”の対応がとれていません。閉じ括弧の記述などが漏れていないか確認してください。括弧“( )”を通常文字として扱う場合は、“¥”(円記号)を付けてエスケープしてください。

- couldn't compile regular expression pattern: brackets [] not balanced

  正規表現の記述内で大括弧“[ ]”の対応がとれていません。閉じ大括弧の記述などが漏れていないか確認してください。大括弧“[ ]”を通常文字として扱う場合は、“¥”(円記号)を付けてエスケープしてください。

- couldn't compile regular expression pattern: invalid character range

  文字集合の範囲の指定に誤りがあります。大括弧“[ ]”で囲まれた部分の記述に誤りがないか確認してください。文字集合の範囲を指定する場合は、[0-9]、[a-z]などのように順序が先の文字から後の文字を指定するようにしてください。

- couldn't compile regular expression pattern: quantifier operand invalid

  正規表現の記号の使用方法に誤りがないか確認してください。

注意事項

正規表現では空白も1文字として認識するため、正規表現中に不要な空白があると、正しく検査できません。また、全角と半角は区別します。

16.2 Systemwalkerスクリプトで使用するコマンド・制御文

本節では、Systemwalkerスクリプトで使用するコマンドと制御文について説明します。

省略可能オプションの記述形式について

“[ ]”がSystemwalkerスクリプト上、特別な意味を持つため、本節では省略可能オプションを示すのに“【 】”を使用しています。

16.2.1 after(一定時間処理を休止する)

機能説明

一定時間処理を休止します。

記述形式

```
after msec
```

オプション

msec

休止時間をミリ秒(1/1000秒)単位で指定します。

復帰値

なし

使用例

1秒間隔で10回ループする

```
set i 0
while {$i < 10} {
    incr i
    after 1000
}
```

実行結果／出力形式

なし

16.2.2 break(ループを中止する)

機能説明

ループ処理を中止します。

記述形式

```
break
```

オプション

なし

復帰値

なし

使用例

5回ループする。

```
set i 0
while {"true"} {
    incr i
    if {$i > 5} {
        break
    }
}
```

実行結果／出力形式

なし

16.2.3 catch(スクリプト行を実行し処理例外をトラップする)

機能説明

スクリプト行を実行して処理例外をトラップします。

スクリプト実行時の処理例外とは、スクリプトの処理がそこで打ち切られ、プロセス自体が強制終了されるエラーです。このようなエラーが発生する例として、実行中のスクリプトに文法エラーや値未設定の変数に対する変数置換の他に、ファイルオープン時のエラー(openコマンド)や、実行プロセスの終了コードが0以外(execコマンド)など、実行時の条件によって発生するものがあります。

catchコマンドはこのような処理例外の発生時に、独自のリカバリ処理を行い、スクリプトのプロセスが終了するのを防ぐために使用します。

記述形式

```
catch {script} 【varname】
```

オプション

{script}

実行するスクリプトコマンド部分です。改行またはセミコロンで区切ることにより複数のコマンドが記述できます。

varname

処理例外が発生した場合にその原因を示すエラーテキストを格納する変数名です。省略時は、エラーテキストは格納されません。

復帰値

0：

処理例外は発生しませんでした

0以外：

処理例外が発生しました。

この場合、グローバル変数errorCodeに、詳細情報として以下の値が格納されます。

ー スクリプトコマンドexecで実行したコマンドの終了コードが“0”以外で終了した場合

```
CHILDSTATUS pid exitcode
```

Pid

【Windows版】

ここに入る数字は意味を持ちません。

【UNIX版】

実行したコマンドのプロセスID

exitcode

実行したコマンドの終了コード

ー スクリプトコマンドexecで実行したコマンドがシグナルにより強制終了された場合

```
CHILDKILLED pid sigName msg
```

Pid

【Windows版】

ここに入る数字は意味を持ちません。

【UNIX版】

実行したコマンドのプロセスID

SigName

実行したコマンドに送られたシグナル名

Msg

詳細メッセージ

- スクリプトコマンドexecで実行したコマンドがシグナルにより一時停止された場合

```
CHILDSUSP pid sigName msg
```

Pid

【Windows版】

ここに入る数字は意味を持ちません。

【UNIX版】

実行したコマンドのプロセスID

SigName

プロセスに送られたシグナル名

Msg

詳細メッセージ

- スクリプトコマンドexprで算術エラーが発生した場合

```
ARITH errName msg
```

ErrName

エラー内容

Msg

詳細メッセージ

- システムコールでエラーが起きた場合

```
POSIX errName msg
```

ErrName

エラー内容

Msg

詳細メッセージ

- エラーとともに返されるメッセージ以外の情報がない場合

```
NONE
```

使用例

ファイルを読み込み用にオープンし、失敗した場合は標準入力からの読み込みをする。

```
if {[catch {
    set file [open data.txt r]
}]} {
    set file ”stdin”
}
```

```
gets $file buf
if {$file != ”stdin”} {
    close $file
}
```

実行結果／出力形式

なし

16.2.4 cd(作業ディレクトリを変更する)

機能説明

作業ディレクトリを変更します。

記述形式

```
cd dirname
```

オプション

dirname

変更先ディレクトリ名を指定します。Windows上のディレクトリ名は、ディレクトリの区切りを“¥”(円記号)の他に“/”(スラッシュ)で記述することが可能です。円記号を使用する場合は、エスケープ文字に扱われないようディレクトリ名全体を“{ }”(中括弧)で囲む必要があります。

復帰値

なし

注意事項

存在しないディレクトリ名を指定した場合、スクリプトは処理例外の発生を検知して強制終了します。

使用例

【Windows版】

カレントディレクトリをc:¥tmpに移動します。

```
cd {c:¥tmp}
```

【UNIX版】

カレントディレクトリを/var/tmpに移動します。

```
cd /var/tmp
```

実行結果／出力形式

なし

16.2.5 close(オープンされたファイルをクローズする)

機能説明

オープンされたファイルをクローズします。

記述形式

```
close channelID
```

オプション

channelID

openコマンドの復帰値として取得したチャンネルIDを指定します。

復帰値

なし

参照

open

注意事項

openコマンドでオープンしたファイルは、最終的に必ずcloseコマンドでクローズする必要があります。

使用例

変数fcに格納されているチャンネルIDをクローズします。

```
close $fc
```

実行結果／出力形式

なし

16.2.6　continue(ループを次の繰り返しまでスキップする)

機能説明

ループ内の以降の処理をスキップして、ループ文の条件判定へ処理をとばします。

記述形式

```
continue
```

オプション

なし

復帰値

なし

使用例

5回ループするうち、最後の3回だけカウント値を表示します。

```
set i 0
while {$i < 5} {
    incr i
    if {$i < 3} {
        continue
    }
    puts $i
}
```

実行結果／出力形式

```
3
4
5
```

16.2.7　exec(サブプロセスを起動する)

機能説明

サブプロセスを起動し、プログラムのコマンドを実行します。

記述形式

【Windows版】

```
exec cmd /c cmdname 【args …】
```

注）execコマンドは、cmd /cと併せて使用してください。

【UNIX版】

```
exec cmdname 【args …】
```

オプション

cmdname

起動するコマンドファイル名を指定します。Windows上のファイル名の場合は、ディレクトリの区切りを“¥”(円記号)の他に“/”(スラッシュ)で記述することが可能です。円記号を使用する場合は、エスケープ文字に扱われないようファイル名全体を“{ }”(中括弧)で囲む必要があります。

args …

起動するコマンドに渡す引数を指定します。

最後の*args*が“&”の場合、コマンドはバックグラウンドで実行されます。コマンドは通常、フォアグラウンドで実行され、execコマンドは*cmdname*のプロセス終了を待ちますが、バックグラウンドで実行した場合は、終了を待たずに復帰します。

また、*args*にはフロー制御記号を指定することが可能です。

Windowsの場合、ファイルをパス指定で与える時は、ディレクトリの区切りに“¥”(円記号)を使用してファイル名全体を“{ }”(中括弧)で囲む必要があります。

フロー制御

*args*に以下のフロー制御記号を記述することで、起動コマンドの標準入出力への操作や複数コマンドを起動しての情報のやりとりが可能となります。

記号・書式	意味	例
|	パイプライン中の個々のコマンドを区切ります。前のコマンドの標準出力を後のコマンドの標準入力へつなげます。同様にして3つ以上のコマンドをつないでいくことが可能です。 “|”の代わりに“|&”を使用すると標準出力の他に標準エラー出力も後のコマンドの標準入力につながれます。	【Windows版】 exec cmd /c dir |cmd /c sort 【UNIX版】 exec ls -l | grep root

記号・書式	意味	例
< *filename*	*cmdname*の標準入力として、*filename*で指定したファイルの内容が読み込まれます。(注1)	【Windows版】 exec cmd /c sort < data.txt 【UNIX版】 exec sort < data.txt
<< *value*	*cmdname*の標準入力として、*value*で指定した値(文字列)が読み込まれます。(注1)	【Windows版】 exec cmd /c sort << $data 【UNIX版】 exec sort << $data
> *filename*	*cmdname*の標準出力が*filename*で指定したファイルに上書きで書き込まれます。(注1) “>”の代わりに“>>”を使用すると上書きでなく追加で書き込まれます。	【Windows版】 exec cmd /c dir > data 【UNIX版】 exec ls > data
>& *filename*	*cmdname*の標準出力と標準エラー出力が*filename*で指定したファイルに上書きで書き込まれます。(注1) “>&”の代わりに“>>&”を使用すると上書きでなく追加で書き込まれます。	【Windows版】 exec cmd /c dir data >& out 【UNIX版】 exec rm -f data >& out
2> *filename*	*cmdname*の標準エラー出力が*filename*で指定したファイルに上書きで書き込まれます。(注2) “2>”の代わりに“2>>”を使用すると上書きでなく追加で書き込まれます。	【Windows版】 exec cmd /c dir data 2> out 【UNIX版】 exec rm -f data 2> out

注1)

*cmdname*の部分が“|”により複数のコマンドからなる場合、読み込みは先頭のコマンドが、書き込みは末尾のコマンドがその対象となります。

注2)

*cmdname*の部分が“|”により複数のコマンドからなる場合、すべてのコマンドの標準エラー出力が*filename*で指定したファイルに格納されます。

復帰値

コマンドがフォアグラウンドで実行されている場合：

cmdnameコマンドの標準出力が返されます。このとき、出力テキストの末尾にある改行コードは削除されます。

フロー制御により複数のコマンドが実行される場合は、末尾のコマンドがその対象となります。

また、フロー制御により標準出力がリダイレクトされている場合は、空文字列を返します。

コマンドがバックグラウンドで実行されている場合：

起動コマンドのプロセスIDが返されます。

フロー制御により複数のコマンドが実行される場合は、全プロセスIDが空白で区切って返されます。そのため、アプリケーションが出力する標準出力をexecコマンドの復帰値として獲得することはできません。

その他入出力情報

- 以下の条件の場合、スクリプトは処理例外の発生を検知し強制終了されます。必要に応じてcatchコマンドで処理例外のトラップを行ってください。
  - コマンドがフォアグラウンドで実行されている、かつ、

－ コマンドの標準エラー出力がリダイレクトされていない状態でコマンドが標準エラー出力へのメッセージを出力する、または、

－ コマンドの終了コードが“0”以外である。

例：

catchによる処理例外のトラップをしない場合

【Windows版】

```
#アプリケーションを起動します。標準出力を変数“outmsg”に格納します。
set outmsg [exec cmd /c {c:¥usr¥bin¥appl001.exe}]

#アプリケーションが標準出力へ出力したデータをジョブの標準出力へ出力します。
puts stdout $outmsg

#アプリケーションの実行まで正常に完了した場合は、終了コード“0”で終了します。
exit 0
```

【UNIX版】

```
#アプリケーションを起動します。標準出力を変数“outmsg”に格納します。
set outmsg [exec /usr/bin/appl001]

#アプリケーションが標準出力へ出力したデータをジョブの標準出力へ出力します。
puts stdout $outmsg

#アプリケーションの実行まで正常に完了した場合は、終了コード“0”で終了します。
exit 0
```

備考.

catchによる処理例外設定がないため、execで起動したコマンドが、“0”以外の終了コードで終了したり、異常終了した場合に以下の問題が発生します。

・ジョブの終了コードは“1”となり、実行コマンドの終了コードが採用されない。
・実行コマンドが標準エラー出力に出力したエラーメッセージなどを獲得することができない。

これらを回避するために、コマンドを起動する際にはcatchコマンドによる処理例外のトラップをすることを推奨します。

・ 処理例外をcatchコマンドでトラップした場合、以下の情報を参照することができます。

－ 標準出力/標準エラー出力情報(errorInfo)

－ 終了コード(errorCode)

参考

標準出力/標準エラー出力情報(errorInfo)について

コマンドの標準出力/標準エラー出力がリダイレクトされていない場合、グローバル変数errorInfoの先頭にその出力文字列が格納されます。errorInfoには、exec文自身の実行エラーを示す文字列も格納されます。

例：

【Windows版】

以下の仕様のアプリケーション(c:¥usr¥bin¥appl001.exe)がエラー終了した場合

仕様：

指定された日付のデータを標準出力に書き込む。
日付の指定が不当な場合は以下のメッセージを標準エラー出力に出力する。
“日付の指定に誤りがあります。yyyymmdd で指定してください。”

```
if {[catch {
        exec cmd /c {c:¥usr¥bin¥appl001.exe} 010520
    }]} {
     puts ”エラー情報 START”
     puts $errorInfo
     puts ”エラー情報 END”
}
```

上記を実行すると、標準出力には以下のように表示されます。

```
エラー情報START
日付の指定に誤りがあります。yyyymmddで指定してください。←
appl001コマンドの出力
   while executing                    ← exec文自身の実行エラーを
"exec cmd /c {c:¥usr¥bin¥appl001.exe} 010520"      ←      示す文字列
エラー情報END
```

【UNIX版】

存在しないファイル(/var/tmp/data)を指定し、lsコマンドを実行する場合

```
if {[catch {
        exec ls /var/tmp/data
    }]} {
     puts ”エラー情報 START”
     puts $errorInfo
     puts ”エラー情報 END”
}
```

上記を実行すると、標準出力には以下のように表示されます。

```
エラー情報 START
/var/tmp/data: ファイルもディレクトリもありません。 ←lsコマンドの出
力
   while executing                    ← exec文自身の実行エラーを
”exec ls /var/tmp/data”                    ←              示す文字列
エラー情報 END
```

参考

終了コード(errorCode)について

コマンドの終了コードが“0”以外の場合は、グローバル変数errorCodeに以下の書式で終了コードが格納されています。また、その他の詳細情報も格納されます。 errorCodeの書式については、“catch（スクリプト行を実行し処理例外をトラップする）”の復帰値を参照してください。

```
CHILDSTATUS pid exitcode
```

pid

【Windows版】

ここに入る数字は意味を持ちません。

【UNIX版】

プロセスID

exitcode

プロセスの終了コード

・ 実行コマンドの入出力について

実行コマンドのリダイレクトされていない入出力は以下のとおりに扱われます。

入出力	フォアグラウンド実行	バックグラウンド実行
標準入力	スクリプトプロセス自身の標準入力から読み込まれる。	
標準出力	通常は、execコマンドの戻り値となる。 処理例外をcatchでトラップした場合には、errorInfoに格納される。	スクリプトプロセス自身の標準出力に書き込まれる。
標準エラー出力	execコマンドの処理例外として扱われ、catchでトラップした場合にはerrorInfoに格納される。	スクリプトプロセス自身の標準エラー出力に書き込まれる。

参照

catch

注意事項

・ 子孫プロセスが常駐するコマンドやシェルの起動について

以下の条件の場合、直接起動したコマンドが終了しても復帰しません。

－ フォアグラウンドでコマンドを起動している、かつ、

－ 起動したコマンドがさらに子プロセスを起動している、かつ、

－ 標準出力と標準エラー出力をリダイレクトしていない。

したがって、デーモン起動シェルなど常駐するプロセス起動を行う場合は、以下のようにバックグラウンドで起動するか、標準出力と標準エラー出力をファイルなどにリダイレクトするようにしてください。

－ バックグラウンド起動

```
exec rdbstart &
```

－ 標準出力と標準エラー出力をファイルにリダイレクト

```
exec rdbstart >& /var/tmp/log.txt
```

－ 標準出力と標準エラー出力を無視

```
exec rdbstart >& /dev/null
```

・ Windowsのコマンドでパラメタの解釈が特殊なコマンドについて【Windows版】

Windowsのプログラムには、コマンドのオプションのダブルクォーテーションなどを特別な文字として解釈する一般的なコマンドと、そういった解釈をしない特殊なコマンドがあります。そのような特殊なコマンドには、スクリプト自身もダブルクォーテーションの解釈を行っているためパラメタを正しく渡せない場合があります。

ー 一般的なコマンドのパラメタ解釈例

コマンドラインの記述	コマンド内での解釈
"abc"	abc
¥"abc¥"	"abc"

特殊なコマンドは、コマンドラインの記述をそのまま解釈します。

特別な文字を解釈しないコマンドを実行させる場合は、コマンドを実行するバッチファイルを作成し、そのバッチファイルをexecで実行するように記述してください。

・ フロー制御記号から始まる引数について

フロー制御記号（< > 2>)から始まる文字列は、引数に指定した場合フロー制御記号として解釈されるため、文字列として渡すことはできません。以下のどちらかの方法で指定してください。

ー 文字列の先頭に半角空白を付加して引数に指定する

文字列として解釈させるために、先頭に半角空白を付加した文字列に変えて引数に指定します。

文字列が変数(msgtxt)に格納されている場合、以下のように記述することで、先頭に半角空白を付加することができます。

```
if {[regexp {^<|^>|^2>} $msgtxt] == 1} {
   set msgtxt " $msgtxt"
}
```

ー シェルスクリプトまたはバッチファイルを利用する

目的のコマンドを起動するシェルスクリプト【UNIX版】またはバッチファイル【Windows版】をスクリプト内で作成し、それをexecコマンドで起動させるようにします。ただし、引数文字列内にダブルクォーテーション(")が含まれていると、シェルスクリプトやバッチファイル内での文字列解釈が正しく行われなくなるため、引数文字列中に含まれるダブルクォーテーションをシングルクォーテーション(')にあらかじめ置き換えます。

例:コマンド(usercmd)の引数に、任意の文字列が格納されている変数(msgtxt)を指定して起動させます。

【UNIX版】

“/var/tmp/usercmd_tmp.sh”は、一時的に作成するシェルスクリプトで、実行後に削除します。

```
if {[regexp {^<|^>|^2>} $msgtxt all] == 1} {
   regsub {"} $msgtxt {'} msgtxt
   set fd [open /var/tmp/usercmd_tmp.sh w]
   puts $fd "usercmd ¥"$msgtxt¥""
   close $fd
   exec sh /var/tmp/usercmd_tmp.sh
   exec /usr/bin/rm -f /var/tmp/usercmd_tmp.sh
} else {
   exec usercmd $msgtxt
}
```

【Windows版】

“c:¥temp¥usercmd_tmp.bat”は、一時的に作成するバッチファイルで、実行後に削除します。

```
if {[regexp {^<|^>|^2>} $msgtxt all] == 1} {
   regsub {"} $msgtxt {'} msgtxt
   set fd [open {c:¥temp¥usercmd_tmp.bat} w]
   puts $fd "usercmd ¥"$msgtxt¥""
   close $fd
   exec {c:¥temp¥usercmd_tmp.bat}
   exec cmd /c del {c:¥temp¥usercmd_tmp.bat}
```

```
} else {
    exec usercmd $msgtxt
}
```

使用例

【Windows版】

dirコマンドを実行し、コマンドの標準出力/標準エラー出力をすべてスクリプトの標準出力へ出力させます。

```
if {[catch {
        set outmsg [exec cmd /c dir {c:¥tmp¥data}]

        # dirの標準出力(exec戻り値)をスクリプトの標準出力へ
        puts stdout $outmsg
    }]} {
   # 処理例外発生
   # outmsgには初期値として$errorInfoを格納しておきます。
   set outmsg $errorInfo

  # dirの出力情報(errorInfoの末尾2行を除いたもの)をスクリプトの標準出力へ
   regexp {(.*)¥n.*¥n.*} $errorInfo all outmsg
   puts stdout $outmsg
}
```

【UNIX版】

lsコマンドを実行し、コマンドの標準出力/標準エラー出力をすべてスクリプトの標準出力へ出力させます。

```
if {[catch {
        set outmsg [exec ls -l /var/tmp/data]

        # lsの標準出力(exec戻り値)をスクリプトの標準出力へ
        puts stdout $outmsg
    }]} {
   # 処理例外発生
   # outmsgには初期値として$errorInfoを格納しておきます。
   set outmsg $errorInfo

  # lsの出力情報(errorInfoの末尾2行を除いたもの)をスクリプトの標準出力へ
   regexp {(.*)¥n.*¥n.*} $errorInfo all outmsg
   puts stdout $outmsg
}
```

実行結果／出力形式

標準出力には、以下のように表示されます。

【Windows版】

・ ファイルc:¥tmp¥dataが存在する場合

```
ドライブ C のボリューム ラベルはありません。
ボリューム シリアル番号は 2420-12F3 です

c:¥tmp のディレクトリ
00/12/01 11:37a                 972 data
               1 個のファイル             972 バイト
                    463,110,144 バイトの空き領域
```

・ ディレクトリ(ファイル)c:¥tmp¥dataが存在しない場合

```
ドライブ C のボリューム ラベルはありません。
ボリューム シリアル番号は 2420-12F3 です

c:¥tmp のディレクトリ

ファイルが見つかりません
```

【UNIX版】

・ ファイル/var/tmp/dataが存在する場合

```
-rw-rw-r--   1 root     other       0  9月 25日  15:05  /var/tmp/data
```

・ ファイル(ディレクトリ)/var/tmp/dataが存在しない場合

```
/var/tmp/data: ファイルもディレクトリもありません。
```

16.2.8 exit(スクリプトを終了する)

機能説明

スクリプトのプロセスを終了します。

記述形式

exit 【*returncode*】

オプション

returncode

プログラムの終了コード。0～128を指定します。省略時は、0になります。

復帰値

なし

使用例

変数rcの内容が0以外の場合、その値を終了コードにして異常終了します。

```
if {$rc != 0} {
    exit $rc
}
```

実行結果／出力形式

なし

16.2.9 expr(算術演算をする)

機能説明

算術計算をします。

記述形式

expr *arg* 【*arg arg …*】

オプション

arg

式を指定します。

使用可能な演算子(優先度の高い順)

- + ~ !	単項マイナス、単項プラス、補数、論理否定
* / %	乗算、除算、剰余
+ -	加算、減算

式中に“()”(小括弧)で囲まれた部分がある場合は、その部分の優先度が上がり、先に演算されます。

復帰値

演算結果を返します。

使用例

3×(10－(4－2))の結果を表示します。

```
set result [expr 3*(10-(4-2))]
puts $result
```

実行結果／出力形式

```
24
```

16.2.10　flush(バッファリングされた出力をフラッシュする)

機能説明

バッファリングされた出力をフラッシュします。



参考

フラッシュとは、書き込みバッファ内に残っているデータを排出させる操作です。例えば、パイプにより標準出力が他のコマンドとつながっている場合など、出力したデータをすぐに相手側に受け取らせるには、フラッシュ操作が必要になります。

記述形式

```
flush channelID
```

オプション

channelID

openコマンドの復帰値として取得したチャンネルIDを指定します。また、“stdout”，“stderr”を指定することにより、標準出力、標準エラー出力へのフラッシュを行います。

復帰値

なし

使用例

標準出力へデータを書き込み、フラッシュします。

```
puts stdout $Data
flush stdout
```

実行結果／出力形式

なし

16.2.11　gets(ファイルから1行読む)

機能説明

ファイルまたは標準入力から1行読み込みます。

記述形式

```
gets channelID 【varname】
```

オプション

channelID

openコマンドの復帰値として取得したチャンネルIDを指定します。また、“stdin”を指定することにより、標準入力からの読み込みを行います。

varname

読み込んだ行の内容を格納する変数名を指定します。省略時は、読み込んだ行の内容は復帰値として返されます。

復帰値

***varname*を指定した場合**

読み込んだ文字列の文字数を返します。1文字も読み込まずファイルの末尾に達した場合は、-1を返します。

***varname*を省略した場合**

読み込んだ文字列を返します。

注意事項

読み込みされる“1行”とは、ファイル内の現在の読み込み位置(open直後は先頭、以後読み込みを行うごとに読み込んだ分だけ移動)から改行またはファイルの末尾までのことを指します。なお、*varname*で指定した変数には、改行コードは格納されません。

使用例

標準入力から1行(改行まで)のデータを読み込み、変数bufに格納します。

```
gets stdin buf
```

実行結果／出力形式

なし

16.2.12　if～else(条件分岐をする)

機能説明

スクリプトを条件実行します。

記述形式

```
if {expr1} 【then】 {body1} 【elseif {expr2} 【then】 {body2}】
【elseif …】 【else {bodyN}】
```

オプション

expr

条件式を記述します。数値は、0が偽でそれ以外が真です。文字では、true,yesが真でfalse,noが偽になります。
条件式内には、コメントを記述しないでください。

条件式で使用できる条件演算子

!	否定
< , > , <= , >= , == , !=	比較
&& , ||	論理積、論理和

body

条件が真の場合に実行されるスクリプトコマンド部分です。改行またはセミコロンで区切ることにより複数のコマンドが記述できます。

復帰値

なし

注意事項

各キーワード(if,then,elseif,else)と“{”(中括弧)の間、および“then”を省略した場合の“}”と“{”の間は、必ず空白またはTAB文字が必要です。

“}”と“{”の間など、body以外のところで改行する場合は、行の最後に“¥”(円記号)を付けて継続行となるようにしてください。

- body以外のところで改行している誤った例:

```
if {$cnt == 1 }
{
    puts "start"
}
```

- 正しい例:

  “}”と“{”の間で改行しない

```
if {$cnt == 1 } {
    puts "start"
}
```

  または、改行した場合、“¥”(円記号)を付けて継続行とする

```
if {$cnt == 1 } ¥
{
    puts "start"
}
```

使用例

変数aの値が0より大きいか0以下であるかを表示します。

```
if {$a > 0} {
    puts ” a > 0”
} else {
    puts ” a <= 0”
}
```

実行結果／出力形式

aの値に応じて、以下のどちらかが出力されます。

```
a > 0
```

または、

```
a <= 0
```

16.2.13　incr(変数に値を加算する)

機能説明

変数に値を加算します。

記述形式

```
incr varname 【increment】
```

オプション

varname

加算対象の変数名を指定します。この変数には10進数の整数が格納されていなければなりません。また、加算結果もこの変数へ格納されます。

increment

加算する値(10進数の整数)を指定します。省略時は、1が加算されます。

復帰値

加算された結果の値（=incrコマンド呼出し前の$*varname* + *increment*)

注意事項

*varname*で指定した変数の設定値、および*increment*の指定値が10進数の整数でない場合、スクリプトは処理例外の発生を検知して、強制終了されます。本コマンド使用時は、必要に応じてcatchコマンドで処理例外のトラップを行ってください。

使用例

5回ループするうち、最後の3回だけカウント値を表示します。

```
set i 0
while {$i < 5} {
    incr i
    if {$i < 3} {
        continue
    }
```

```
    puts $i
}
```

実行結果／出力形式

```
3
4
5
```

16.2.14　open(ファイルをオープンする)

機能説明

ファイルをオープンします。

記述形式

```
open filename 【access】
```

オプション

filename

オープンするファイル名を指定します。Windows上のファイル名の場合は、ディレクトリの区切りを“¥”(円記号)の他に“/”(スラッシュ)で記述することが可能です。円記号を使用する場合は、エスケープ文字に扱われないようファイル名全体を“{}”(中括弧)で囲む必要があります。また、ファイル名にスペースまたはタブが含まれる場合は、「”」(ダブルクォーテーション)で囲む必要があります。

access

アクセスモードの指定をします。省略時は、読み込み専用(r)です。指定内容とその意味は以下のとおりです。

アクセスモードの種類

コード	説明
r	読み込み専用モードでオープンします(省略時)。ファイルは存在している必要があります。
r+	読み書きモードでオープンします。ファイルは存在している必要があります。
w	書き込み専用モードでオープンします。ファイルが存在している時は上書きします。なければ新規作成します。
w+	読み書きモードでオープンします。ファイルが存在している時は上書きします。なければ新規作成します。
a	書き込み専用モードでオープンします。ファイルは存在している必要があります。書き込みデータは、ファイルの最後に追加されます。
a+	読み書きモードでオープンします。ファイルが存在していない時は、新規に作成します。書き込みデータはファイルの最後に追加されます。

復帰値

チャンネルID

オープンしたファイルへのアクセス(closeを含む)には、このチャンネルIDを指定するため、復帰値は必ずコマンド置換により変数に格納する必要があります。

参照

close

注意事項

アクセス権のないファイルを指定した場合などファイルオープンに失敗した場合、スクリプトは処理例外の発生を検知して強制終了されます。本コマンド使用時は、必要に応じてcatchコマンドで処理例外のトラップを行ってください。

使用例

【Windows版】

ファイルc:¥tmp¥dataを書き込みモードでオープンし、そのチャンネルIDを変数fcに格納します。

```
set fc [open {c:¥tmp¥data} w]
```

【UNIX版】

ファイル/var/tmp/dataを書き込みモードでオープンし、そのチャンネルIDを変数fcに格納します。

```
set fc [open /var/tmp/data w]
```

実行結果／出力形式

なし

16.2.15　puts(ファイルに書き込む)

機能説明

ファイルまたは標準出力・標準エラー出力へ書き込みます。

記述形式

```
puts 【channelID】 string
```

オプション

channelID

openコマンドの復帰値として取得したチャンネルIDを指定します。また、“stdout”,“stderr”を指定することにより、標準出力、標準エラー出力への書き込みを行います。省略時は、標準出力への書き込みをします。

string

書き込むテキストデータを指定します。

復帰値

なし

注意事項

書き込み時には、必ず末尾に改行コードが付加されます。

使用例

【Windows版】

ログファイルと標準エラー出力へエラーメッセージを出力します。

```
set errlog [open {c:¥var¥tmp¥log.txt} a]
puts $errlog $ErrMsg
puts stderr $ErrMsg
close $errlog
```

【UNIX版】

ログファイルと標準エラー出力へエラーメッセージを出力します。

```
set errlog [open /var/tmp/log.txt a]
puts $errlog $ErrMsg
puts stderr $ErrMsg
close $errlog
```

実行結果／出力形式

なし

16.2.16 regexp(文字列と正規表現のマッチング検査と一致部分の切り出しをする)

機能説明

文字列と正規表現のマッチング検査と一致部分の切り出しをします。

記述形式

```
regexp exp string 【matchvar】 【submatchvar1 submatchvar2 …】
```

オプション

exp

正規表現を指定します。

string

検査対象となる文字列を指定します。

matchvar

一致した場合に、*string*中の正規表現に一致している部分を切り出して格納する変数の名前を指定します。省略時は、一致した部分の格納が行われません。

submatchvar

一致した場合に、*string*中で正規表現内の“()”(小括弧)で囲まれた部分に一致している部分を切り出して格納する変数の名前を指定します。省略時は、一致した部分の格納が行われません。

ポイント

正規表現とは、文字列の構造を記号で表したもので、本コマンドは*string*で指定した文字列中に正規表現*exp*で表した構造に一致する部分があるか検査するものです。
一致した場合は、*exp*で表した構造に一致した部分を*string*から切り出し、*matchvar*の変数に格納します。また、正規表現中には任意の箇所に任意の個数だけ文字列の切り出し箇所を指定可能で、その指定がある場合は前から順に*submatchvar*の変数に格納されます。

正規表現の書き方については、“16.1.4 正規表現”を参照してください。

復帰値

0： 一致しませんでした。

1： 一致しました。

使用例

- 指定文字の前後を切り出す

  文字列中に“,”(コンマ)があるか検査し、ある場合は一致部分すべてをallに、コンマの前の部分と後の部分をそれぞれ変数sub1,sub2に切り出します。(コンマの前後は空文字でも可)

```
regexp {(.*),(.*)} $text all sub1 sub2
```

- 切り出し部分を指定する

  以下のメッセージから、“< >”で囲まれた部分を切り出して、変数Var0に格納します。

  ネットワークで"LinkUp"が発生しました。<DBサーバに異常が発生しました。>

```
regexp {<([^>]*)} $EventText All Var0
```

  変数Var0には“.DBサーバに異常が発生しました。”が格納されます。

- 特殊文字は、"¥"円記号でエスケープする

  以下のメッセージから"JobName="以降から空白までの部分を変数Var0に、RC=以降の"[ ]"(大括弧)で囲まれた部分を変数Var1に、"device="以降の"( )"(小括弧)で囲まれた部分を変数Var2に、それぞれ切り出して格納します。

  JOBNET Error. JobName=SCRJB_1 RC=[0022] xxx-device=(DX038)

```
regexp {JobName=([^ ]*) RC=¥[([^¥]]*)¥] .*device=¥(([^
¥)]*)¥)} $EventText All Var0 Var1 Var2
```

  それぞれ変数Var0には“SCRJB_1”、変数Var1には“0022”、変数Var2には“DX038”が格納されます。

実行結果／出力形式

なし

16.2.17 regsub(正規表現のパターンマッチングに基づいて置換する)

機能説明

正規表現のパターンマッチングに基づいて置換を行います。

記述形式

```
regsub 【-all】 exp string subspec varname
```

オプション

-all

正規表現に一致した箇所すべてを置換する場合に指定します。省略時は、最初に一致した箇所だけ置換します。

exp

置換箇所を特定する正規表現を指定します。

string

置換対象となる文字列を指定します。

subspec

置き換え文字列を指定します。置き換え文字列は、必ず“{}”(中括弧)で囲んでください。また、置き換え文字列中に“¥”または“&”を入れる場合は、それぞれ“¥¥”,“¥&”と表記してください。

varname

置換の結果、できた文字列を格納する変数の名前を指定します。

ポイント

正規表現とは、文字列の構造を記号で表したもので、本コマンドは*string*で指定した文字列中から正規表現*exp*で表した構造と一致する部分を探し、一致する部分を*subspec*で指定した文字列に置き換えて*varname*で指定した変数に格納します。

正規表現の書き方については、“16.1.4 正規表現”を参照してください。

復帰値

0： 一致しませんでした。

1： 一致しました。

使用例

文字列の間の空白をすべてコンマに置換します。

```
set text ” a b c d”
regsub -all ” ” $text {,} result
puts $result
```

実行結果／出力形式

```
a,b,c,d
```

16.2.18 set(変数に値を格納する)

機能説明

変数に値を格納します。

記述形式

```
set varname 【value】
```

オプション

varname

値を格納する変数名を指定します。

value

格納する値です。数値または文字列を指定します。省略時は、*varname*で指定した変数の値は変更されません。

復帰値

処理後に*varname*変数に格納されている値(*value*を指定した場合は*value*と同じ値、*value*が省略された場合は*varname*変数にあらかじめ格納されていた値)

注意事項

*value*を省略した場合、復帰値として*varname*変数に格納されていた値が返されます。このため、*varname*で指定する変数が未設定(新規)であるとエラーになります。

使用例

変数rcに数値を変数msgにテキストを格納します。

```
set rc 1
set msg ”File open error”
```

実行結果／出力形式

なし

16.2.19 string(文字列を操作する)

機能説明

文字列に対して、比較、取り出し、変換、切断などの操作を行います。

記述形式

```
string option arg 【arg …】
```

オプション

option

操作内容を指定するオプションキーワードを指定します。指定可能なオプションの指定方法と説明を以下に示します。

arg …

操作に必要な引数を指定します。

オプションごとの説明

string compare *string1 string2*	文字列を比較します。 *string1*が辞書順で*string2*より小さいか等しいか大きいかによってそれぞれ-1、0、1を返します。
string first *string1 string2*	*string2*の中に*string1*が出現するかを調べます。 見つかれば*string2*の中の最初に*string1*が出現した場所(先頭文字のインデックス(注))を返します。見つからなければ -1を返します。
string index *string charIndex*	*string*の中の*charIndex*で示される場所(インデックス(注))の文字を返します。
string last *string1 string2*	*string2*の中に*string1*が出現するかを調べます。 見つかれば*string2*の中で最後に*string1*が出現した場所(先頭文字のインデックス(注))を返します。見つからなければ-1を返します。
string length *string*	*string*の文字列の長さを返します。
string match *pattern string*	*string*に*pattern*が一致するか調べ、一致する文字列があれば1を、なければ0を返します。 *pattern*には、UNIXのCシェルでファイル名指定などに用いる特殊文字も使用できます。*pattern*中で特殊文字として扱われるものとその意味を以下に説明します。

	：任意の文字の並び(空文字列を含む)と解釈されます。 　例) パターン ab は、abで始まる任意の文字列に一致します。 ?：任意の1文字と解釈されます。 　例) パターン ab? は、abで始まる3文字の文字列に一致します。 []：大括弧内に指定された任意の1文字と解釈されます。また、大括弧内にa-zのように指定された場合は、a から z までの任意の文字として解釈されます。 　例) パターン ab[0-9] は、abで始まりその後ろが数字である3文字の文字列に一致します。 ¥：パターン記述上の特殊文字(*?[]¥)の直前に記述することにより、その文字の特殊な意味を消し、単なる1文字として扱わせます。 　例) パターン ab¥* は、文字列 ab* に一致します。 pattern中に特殊文字が含まれていない場合は、*pattern*と完全に同一である場合に一致となります。
string range *string i j*	文字列の*i*から*j*で示される場所(インデックス(注))の文字列を返します。
string tolower *string*	*string*を小文字に変換した文字列を返します。
string totitle *string*	*string*の先頭を大文字、残りを小文字に変換した文字列を返します。
string toupper *string*	*string*を大文字に変換した文字列を返します。
string trim *string* 【*chars*】	*string*の最初か最後に、*chars*で指定された文字セットに含まれる文字があれば、その文字を削除した文字列を返します。*chars*省略時は、空白文字が削除されます。
string trimleft *string* 【*chars*】	*string*の最初に、*chars*で指定された文字の並びがあれば、その並びを削除した文字列を返します。*chars*省略時は、空白文字が削除されます。
string trimright *string* 【*chars*】	*string*の最後に、*chars*で指定された文字セットに含まれる文字があれば、その文字を削除した文字列を返します。*chars*省略時は、空白文字が削除されます。
string wordend *string index*	*string*の中で、インデックス(注)で示される文字を含んだ単語の次の文字の位置を返します。
string wordstart *string index*	*string*の中で、インデックス(注)で示される文字を含んだ単語の先頭の文字の位置を返します。

注)

インデックスとは、先頭文字を0とした場合の文字列中の位置(n文字目)を表します。
日本語文字(2バイト文字)も1文字として数えられます。

ポイント

文字列判定にstringコマンドを使用する場合、stringコマンドだけで複雑な条件は指定できないので論理式を利用します。

例:

変数EventTextに“AAA”または“BBB” という文字列が格納されていた場合、標準出力に表示する。

```
if {[string first "AAA" $EventText] > 0 || [string first "BBB" $EventText] > 0} {
  puts $EventText
}
```

復帰値

“オプションごとの説明”内を参照してください。

使用例

変数bufに格納されている文字列の先頭3文字を抜き出し、変数topに格納します。

```
set top [string range $buf 0 2]
```

実行結果／出力形式

なし

16.2.20 while(ループする)

機能説明

条件が成立する間、処理を繰り返します。

記述形式

```
while {test} {body}
```

オプション

test

ループ(*body*)を繰り返す条件式を指定します。数値は、0が偽でそれ以外が真です。文字では、true,yesが真でfalse,noが偽になります。条件式内には、コメントを記述しないでください。

条件式で使用できる条件演算子

!	否定
< , > , <= , >= , == , !=	比較
&& , ||	論理積、論理和

body

条件が真の間実行されるスクリプトコマンド部分です。改行またはセミコロンで区切ることにより複数のコマンドが記述できます。

復帰値

なし

注意事項

- “while”と“{}”(中括弧)の間、および条件式の“}”(閉じ中括弧)と実行部分の“{”(開き中括弧)の間は、必ず空白またはTAB文字が必要です。

  “}”と“{”の間など、body以外のところで改行する場合は、行の最後に“¥”(円記号)を付けて継続行となるようにしてください。

・ while文の条件判定の論理に誤りがあるとスクリプトは無限ループをする場合があります。それを回避するために、以下のような仕組みで、ある回数より多くループすることがないよう対処をすることを推奨します。

  ー while文の中で他で使用しないカウンタ用の変数を設け、その値をwhile文の処理の早い段階で毎回チェックさせる。

使用例

5回ループします。

```
set i 0
while {$i < 5} {
    incr i
}
```

実行結果／出力形式

なし

16.3 デバッグで使用するコマンド

Systemwalkerスクリプトは、C言語のようなプログラミング言語と異なり、コンパイル手続きなしで動作させることができます。しかしその反面、単純な記述ミスや論理的な誤り、実行環境に対する考慮不足などは動作させるまで検出されにくいという欠点を持っています。

そこで、スクリプト作成後の段階において、動作確認・デバッグを実施することが必要です。

本節では、Systemwalkerスクリプトのデバッグ時に使用するコマンドについて説明します。

16.3.1 トレース拡張コマンド

Systemwalkerスクリプト内に以下のコマンドを記述して、トレースを採取することにより、スクリプトのデバッグを可能にします。

・ sw_TcOpenTrace(トレースファイルオープンコマンド)

・ sw_TcWriteTrace(トレース情報出力コマンド)

・ sw_TcCloseTrace(トレースファイルクローズコマンド)

16.3.1.1 sw_TcOpenTrace (トレースファイルオープンコマンド)

機能説明

トレースファイルをオープンし、トレース採取のための初期処理を行います。スクリプトの実行トレース採取を開始する場合は、本コマンドが最初に呼び出されなければなりません。

記述形式

```
sw_TcOpenTrace 【-size fsize】【-level tlevel】
【-mode group|other】 filename
```

オプション

-size *fsize*

*fsize*にトレースファイル1ファイルあたりの最大サイズを1～100000[KB]で指定します。省略時のデフォルト値は、50KBです。

トレース採取は、2つのファイルが作成されサイクリックに採取されるため、ファイルサイズの最大値の見積もりはここで指定するサイズの2倍になります。

-level *tlevel*

*tlevel*にトレースレベルを0～2で指定します。省略時のデフォルト値は、1です。以下にレベルごとの採取内容と推奨例を示します。

0：	それ以降のsw_TcWriteTraceコマンドは無効となりトレースは採取されません。
1：	それ以降のsw_TcWriteTraceでトレースレベル1のものだけが採取されます。 例：実運用時にも採取する情報の出力に使用 ・起動時や処理結果など動作ログとして残す情報の出力 ・異常を検知した場合の詳細情報など、エラーログとして残す情報の出力
2：	それ以降のsw_TcWriteTraceでトレースレベル1と2のものが採取されます。 例：動作確認テスト時やデバッグ時に採取する情報の出力に使用 ・処理分岐点や外部プログラム起動の前後など、スクリプトのルートチェックとして残す情報の出力 ・読み込みファイルの内容やループ処理内での変数値など、スクリプト内情報として残す情報の出力

-mode group|other

作成するトレースファイルの書き込みモードを指定します。省略時は、所有者のみに書き込み許可属性を設定します。本オプションは、Windows上では無視されます。

group

所有者およびグループに書き込み許可属性を設定します。

other

所有者、グループおよびその他に書き込み許可属性を設定します。

filename

トレースファイル名を指定します。パス指定なしでファイル名のみ指定した場合はデフォルトのトレース格納ディレクトリへ採取され、パス指定をした場合はそのパスに従って採取されます。

Windows上のファイル名の場合は、ディレクトリの区切りを“¥”(円記号)の他に“/”(スラッシュ)で記述することが可能です。円記号を使用する場合は、エスケープ文字に扱われないようファイル名全体を“{}”(中括弧)で囲む必要があります。

トレースは、デフォルトで以下のディレクトリに格納されます。

【Windows版】

Systemwalker Operation Managerインストールディレクトリ ¥MpWalker.JM¥mpfwtco¥trc

【UNIX版】

/var/opt/FJSVfwtco/trc

注意

トレースファイルが作成されるディレクトリは、Systemwalkerスクリプトを実行するユーザが更新権を持つディレクトリにしてください。

参考

トレースは、2つのファイルが作成されそれらに対してサイクリックに採取していきます。そのため、実際のファイル名は、ここで指定するファイル名の後ろに“#1.txt”または“#2.txt”が付加された名前となります。

例えば、{c:¥tmp¥trc}と指定した場合、c:¥tmp配下にtrc#1.txtとtrc#2.txtが作成されます。【Windows版】

“/var/tmp/trc”と指定した場合、/var/tmp配下にtrc#1.txtとtrc#2.txtが作成されます。【UNIX版】

復帰値

正の値： 正常終了。トレースコマンドで使用するハンドル

-9： メモリ不足

-15： コード変換エラー

-17： ファイルオープンエラー

参照

sw_TcCloseTrace

実行環境による差異

OSやエディションによる差異はなし。

使用例

【Windows版】

ファイルc:¥var¥tmp¥usrtrcをトレースファイルに指定してトレースをオープンし、ハンドルを変数hwndに格納します。

```
set hwnd [sw_TcOpenTrace {c:¥var¥tmp¥usrtrc}]
```

【UNIX版】

ファイル/var/tmp/usrtrcをトレースファイルに指定してトレースをオープンし、ハンドルを変数hwndに格納します。

```
set hwnd [sw_TcOpenTrace /var/tmp/usrtrc]
```

実行結果／出力形式

なし

16.3.1.2 sw_TcWriteTrace (トレース情報出力コマンド)

機能説明

トレースファイルにトレース情報を出力します。

記述形式

```
sw_TcWriteTrace 【-level tlevel】 thandle text
```

オプション

-level *tlevel*

*tlevel*にトレースレベル1または2を指定します。省略時のデフォルト値は、1です。

thandle

sw_TcOpenTraceで取得したハンドルを指定します。空文字列 “” を指定した場合は、トレースの採取はされません。

text

トレースファイルに出力する任意の文字列を900バイト以内で指定します。900バイトを超える文字列は、900バイトで切り捨てられます。

復帰値

TCOK：	正常終了
TCERR_WRITE：	ファイル書き込みエラー
TCERR_CODECONV：	コード変換エラー

参照

sw_TcOpenTrace

実行環境による差異

OSやエディションによる差異はなし。

使用例

変数hwndにsw_TcOpenTraceの復帰値が格納されている場合、文字列“File open error” をhwndが示すトレースファイルに書きます。

```
sw_TcWriteTrace $hwnd "File open error"
```

実行結果／出力形式

```
Fri Apr  4 10:37:00 2014    18879 File open error
          (1)               (2)       (3)
```

出力項目の説明

出力項目について説明します。

(1)時刻

出力時の時刻が“曜日 月 日 時:分:秒 年”の形式で出力されます。

(2)プロセスID

systemwalkerスクリプトのプロセスIDが出力されます。

(3)文字列

sw_TcWriteTrace で指定した文字列が出力されます。

16.3.1.3 sw_TcCloseTrace (トレースファイルクローズコマンド)

機能説明

トレースファイルをクローズし、トレースの採取を終了します。

記述形式

```
sw_TcCloseTrace thandle
```

オプション

thandle

sw_TcOpenTraceで取得したハンドルを指定します。

復帰値

TCOK　正常終了
：
TCERR　異常終了
：

参照

sw_TcOpenTrace

実行環境による差異

OSやエディションによる差異はなし。

使用例

変数hwndにsw_TcOpenTraceの復帰値が格納されている場合、hwndが示すトレースファイルをクローズする。

```
sw_TcCloseTrace $hwnd
```

実行結果／出力形式

なし

16.3.2　swotclsh(スクリプト実行コマンド)

作成したSystemwalkerスクリプトを単体で実行させることができます。スクリプトの事前動作確認のために使用します。

機能説明

Systemwalkerスクリプトを実行します。実行できるスクリプトは、Systemwalkerが提供するTcl/Tk標準機能の制御文、コマンドおよびSystemwalker拡張コマンドです。作成したスクリプトのデバッグをするために使用します。

記述形式

```
swotclsh scriptfile 【parameter1 parameter2 …】
```

オプション

scriptfile

実行するスクリプトファイル名をフルパスで指定します。



注意

Systemwalker Operation Managerで提供される拡張コマンドは実行可能ですが、Systemwalker Centric Managerで提供される拡張コマンドは実行できません。

parameter1 parameter2 …

*scriptfile*に渡す引数を指定します。

復帰値

復帰値は、実行するスクリプトファイルの記述で操作することが可能です。ただし、スクリプトが返してもよい復帰値は、0～128までとします。

0～128：	実行スクリプトの結果コード
129～255：	スクリプト実行コマンドによるエラー 返す値は、同時に標準エラー出力に出力されるエラーメッセージ(“swotclsh:”で始まるメッセージ)のIDと同じ値になります。

コマンド格納場所

Windows	Systemwalker Operation Managerインストールディレクトリ¥MpWalker.JM¥bin
Solaris	/usr/bin

実行に必要な権限

なし

注意事項

実行するスクリプトファイルが原因でエラーが発生した場合は、ラベルなどがなくスクリプトファイル名や行番号が明示されたメッセージが、標準エラー出力に出力されます。以下に出力エラーメッセージの例を示します。

- 文法エラーの場合

```
wrong # args: no expression after "if" argument
    while executing
"if"
    (file "err.swt" line 5)
```

- コマンド実行時のエラーの場合

```
invalid command name "noentry"
    while executing
"noentry"
    (file "err.swt" line 2)
```

- コマンド実行結果がエラーの場合【Windows版】

```
couldn't open "c:¥aaa¥bbb": no such file or directory
    while executing
"open {c:¥aaa¥bbb}"
    (file "err.swt" line 17)
```

- コマンド実行結果がエラーの場合【UNIX版】

```
couldn't open "/aaa/bbb": no such file or directory
    while executing
"open /aaa/bbb"
    (file "err.swt" line 17)
```

実行環境による差異

OSやエディションによる差異はなし。

使用例

【Windows版】

スクリプトファイルc:¥home¥scripts¥a.swt をテスト実行します。

```
swotclsh c:¥home¥scripts¥a.swt
```

【UNIX版】

スクリプトファイル/home/scripts/a.swt をテスト実行します。

```
swotclsh /home/scripts/a.swt
```

実行結果／出力形式

a.swtの処理に依存します。

a.swtが、スクリプト内で標準出力/標準エラー出力に情報を書く処理をしている(puts stdout xxxxx, puts stderr xxxxx)場合は、出力結果が端末上に表示されます。

16.4　スクリプトの例

本節では、Systemwalkerスクリプトの記述例を2つ挙げて説明します。

例1

以下の例では、あるファイルの存在を確認し、存在していればアプリケーションを起動し、ファイルが存在しない場合は、アプリケーションを起動しない制御を行います。

- 入力

  引数として存在確認するファイル名を指定します

- 出力

終了コード	意味
0	正常終了
4	引数が指定されていません。
8	指定されたファイルが存在しません。
16	アプリケーションの終了コードが“0”以外で終了、またはアプリケーションが標準エラー出力にメッセージを出力しました。

【Windows版】

```
#
# This script is sample.
#
# check argument
# 引数の個数を判定
if { $argc == 0 } {
        # 引数の個数が0の場合は、エラーメッセージを出力
        puts stderr "file name was not specified."
        # この場合、終了コード"4"で終了
        exit 4
```

```
}
# set file name
# 引数に指定された文字列を変数"filename"に設定
set filename $argv

# check file
# 処理例外をトラップする設定
if {[catch {
      # ファイルをオープン
      set file1 [open $filename r]
  }]} {
      # 処理例外がトラップされたらエラーメッセージを出力
      puts stderr "file not found."
      # この場合、終了コード"8"で終了
      exit 8
}
# オープンしたファイルをクローズ
close $file1

# 処理例外をトラップする設定
if {[catch {
      # アプリケーションを起動し、標準出力を変数"outmsg"に格納
      set outmsg [exec cmd /c {c:¥usr¥bin¥appl001.exe}]
     # アプリケーションが標準出力へ出力したデータをジョブの標準出力へ出力
      puts stdout $outmsg
  }]} {
      # 処理例外がトラップされたらプロセスのＩＤや終了コードを出力
      puts stderr $errorCode
      # 処理例外がトラップされたらエラーメッセージを出力
      puts stderr $errorInfo
      # この場合、終了コード"16"で終了
      exit 16
}
# アプリケーションの実行まで正常に完了した場合は、終了コード"0"で終了
exit 0
```

【UNIX版】

```
#
# This script is sample.
#
# check argument
# 引数の個数を判定
if { $argc == 0 } {
      # 引数の個数が0の場合は、エラーメッセージを出力
      puts stderr "file name was not specified."
      # この場合、終了コード"4"で終了
      exit 4
}
# set file name
# 引数に指定された文字列を変数"filename"に設定
set filename $argv

# check file
# 処理例外をトラップする設定
if {[catch {
     # ファイルをオープン
```

```
    set file1 [open $filename r]
  }]} {
    # 処理例外がトラップされたらエラーメッセージを出力
    puts stderr "file not found."
    # この場合、終了コード"8"で終了
    exit 8
}
# オープンしたファイルをクローズ
close $file1

# 処理例外をトラップする設定
if {[catch {
    # アプリケーションを起動し、標準出力を変数"outmsg"に格納
    set outmsg [exec /usr/bin/appl001]
   # アプリケーションが標準出力へ出力したデータをジョブの標準出力へ出力
    puts stdout $outmsg
  }]} {
    # 処理例外がトラップされたらプロセスのＩＤや終了コードを出力
    puts stderr $errorCode
    # 処理例外がトラップされたらエラーメッセージを出力
    puts stderr $errorInfo
    # この場合、終了コード"16"で終了
    exit 16
}
# アプリケーションの実行まで正常に完了した場合は、終了コード"0"で終了
exit 0
```



注意

スクリプト内で実行するコマンドの標準出力について

スクリプト内で実行されるコマンドに対して、標準出力は引き継がれません。そのため、例1内のアプリケーションの起動箇所を以下のようにすると、アプリケーションの標準出力がジョブの標準出力として出力されません。

つまり、Systemwalkerスクリプトをスケジュールジョブとして実行した場合に、スクリプトから実行されるアプリケーションが出力する処理結果メッセージなどが、“前回履歴”画面で確認することができませんので、注意が必要です。

【Windows版】

```
if {[catch {
    # アプリケーションを起動します。
    exec cmd /c {c:¥usr¥bin¥appl001.exe}
}]} {
puts stderr $errorCode
puts stderr $errorInfo
exit 16
}
exit 0
```

【UNIX版】

```
if {[catch {
    # アプリケーションを起動します。
    exec /usr/bin/appl001
}]} {
    puts stderr $errorCode
    puts stderr $errorInfo
```

```
    exit 16
}
exit 0
```

例2

以下の例では、アプリケーションが処理するファイル名一覧が格納されているファイルを読み込み、ファイル内に記述されたすべてのファイル名に対して、アプリケーションを起動します。アプリケーションを起動する際には、ファイル名を引数として渡します。

- 入力

  引数として処理対象であるファイル名一覧を確認したファイル名を指定します

- 出力

終了コード	意味
0	正常終了
4	引数が指定されていません。
8	指定されたファイルが存在しません。
16	アプリケーションの終了コードが“0”以外で終了、またはアプリケーションが標準エラー出力にメッセージを出力しました。

【Windows版】

```
#
# This script is sample.
#
# check argument
# 引数の個数を判定
if { $argc == 0 } {
        # 引数の個数が0の場合は、エラーメッセージを出力
        puts stderr "file name was not specified."
        # この場合、終了コード"4"で終了
        exit 4
}

# set file name
# 引数に指定された文字列を変数"filename"に設定
set filename $argv
# check file
# 処理例外をトラップする設定
if {[catch {
      # ファイルをオープン
      set file1 [open $filename r]
   }]} {
      # 処理例外がトラップされたらエラーメッセージを出力
      puts stderr "file not found."
      # この場合、終了コード"8"で終了
      exit 8
}
# 1行読み込み、ファイルの末尾に達するまでループ
while { [gets $file1 param] != -1 } {
      # 処理例外をトラップする設定
      if {[catch {
          # アプリケーションを起動し、この際読み込んだ文字列を渡す
```

```
        # 標準出力を変数"outmsg"に格納
        set outmsg [exec cmd /c {c:¥usr¥bin¥appl002.exe} $param]
        # アプリケーションが標準出力へ出力したデータをジョブの標準出力へ出力
        puts stdout $outmsg
    }]} {
        # 処理例外がトラップされたらプロセスのＩＤや終了コードを出力
        puts stderr $errorCode
        # 処理例外がトラップされたらエラーメッセージを出力
        puts stderr $errorInfo
        # オープン中のファイルをクローズ
        close $file1
        # 終了コード"16"で終了
        exit 16
    }
}
# ファイルをクローズ
close $file1
# すべてのアプリケーションの実行が正常に完了した場合は、終了コード"0"で終了
exit 0
```

【UNIX版】

```
#
# This script is sample.
#
# check argument
# 引数の個数を判定
if { $argc == 0 } {
      # 引数の個数が0の場合は、エラーメッセージを出力
      puts stderr "file name was not specified."
      # この場合、終了コード"4"で終了
      exit 4
}

# set file name
# 引数に指定された文字列を変数"filename"に設定
set filename $argv
# check file
# 処理例外をトラップする設定
if {[catch {
    # ファイルをオープン
    set file1 [open $filename r]
  }]} {
    # 処理例外がトラップされたらエラーメッセージを出力
    puts stderr "file not found."
    # この場合、終了コード"8"で終了
    exit 8
}
# 1行読み込み、ファイルの末尾に達するまでループ
while { [gets $file1 param] != -1 } {
    # 処理例外をトラップする設定
    if {[catch {
        # アプリケーションを起動し、この際読み込んだ文字列を渡す
        # 標準出力を変数"outmsg"に格納
        set outmsg [exec /usr/bin/appl002 $param]
        # アプリケーションが標準出力へ出力したデータをジョブの標準出力へ出力
        puts stdout $outmsg
    }]} {
```

```
        # 処理例外がトラップされたらプロセスのＩＤや終了コードを出力
        puts stderr $errorCode
        # 処理例外がトラップされたらエラーメッセージを出力
        puts stderr $errorInfo
        # オープン中のファイルをクローズ
        close $file1
        # 終了コード"16"で終了
        exit 16
    }
}
# ファイルをクローズ
close $file1
# すべてのアプリケーションの実行が正常に完了した場合は、終了コード"0"で終了
exit 0
```

16.5　スクリプトのデバッグ例

本節では、Systemwalkerスクリプトのデバッグ例として以下の2つの例を説明します。

- デバッグで使用するコマンドの組み込み例
- トレースレベルの変更の仕方の例

デバッグで使用するコマンドの組み込み例

“16.3 デバッグで使用するコマンド”をスクリプト内に組み込んだ例を以下に示します。

例：引数で渡されたファイルを読み込むスクリプト

スクリプトファイル	level (注)	出力内容

# トレースオープン 	
# 実運用時は"level 2"を削除のこと 	
set Trchwnd [sw_TcOpenTrace -level 2 /var/log/G001]
if {$Trchwnd < 0} {
puts stderr "Trace open err"
exit 1
}
# 起動ログ出力 	
sw_TcWriteTrace $Trchwnd "業務001 START"
# ファイル名取得 	
set filename [lindex $argv 0]
sw_TcWriteTrace -level 2 $Trchwnd "Input file name: $filename"
# 動作制御ファイルオープン 	
if {[catch {
set cid [open $filename r]
}]} {
# open のエラーをキャッチした場合 	
# エラーログ出力 	
sw_TcWriteTrace $Trchwnd "File open err (filename :$filename)"
sw_TcCloseTrace $Trchwnd
exit 1
}
sw_TcWriteTrace -level 2 $Trchwnd "File opened "
# ファイル読み込み 	
set rc [gets $cid buf]
if {$rc == -1} {
# ファイルが空のため読み込みバッファにデフォルト値を設定する 	
	1
	2
	1
	2
	2
		動作ログ
		スクリプト内情報（ルートチェック）
		エラーログ
		ルートチェック
		ルートチェック

スクリプトファイル	level (注)	出力内容
sw_TcWriteTrace -level 2 $Trchwnd "File empty " set buf 100 } sw_TcWriteTrace -level 2 $Trchwnd "Gets data: $buf" close $cid … # 終了ログ出力 sw_TcWriteTrace $Trchwnd "業務001 END" sw_TcCloseTrace $Trchwnd	2 1	スクリプト内情報 (ルートチェック) 動作ログ

注)

levelの欄は、トレースレベルを表します。
トレースレベルのデフォルトは1のため、1の場合はコマンド内での指定を省略しています。

トレースレベルの変更の仕方の例

トレースレベルの指定は、Systemwalkerスクリプト内に記述するトレース拡張コマンドのオプションで行います。Systemwalker Operation Managerでの運用中にトレースレベルの変更が必要になった場合、スクリプト内を直接修正するのは困難なため、以下に示すような方法により、スクリプトを変更せずにトレースレベルを変更可能にしておくことができます。

- スクリプトの引数による指定

  スクリプト起動時の引数によりトレースレベルを受け渡します。

  例：第1引数がトレースレベルを表す場合【Windows版】

```
# 第1引数からトレースレベル取得
set trclevel [lindex $argv 0]
# 指定値チェック
if {$trclevel < 0 || $trclevel > 2} {
      # 指定値エラー
      puts stderr "trace level err :$trclevel"
      exit 1
}
# トレースオープン
set Trchwnd [sw_TcOpenTrace -level $trclevel {c:¥var¥log
¥G001.trc}]
…
sw_TcCloseTrace $Trchwnd
exit 0
```

  例：第1引数がトレースレベルを表す場合【UNIX版】

```
# 第1引数からトレースレベル取得
set trclevel [lindex $argv 0]
# 指定値チェック
if {$trclevel < 0 || $trclevel > 2} {
      # 指定値エラー
      puts stderr "trace level err :$trclevel"
      exit 1
}
# トレースオープン
set Trchwnd [sw_TcOpenTrace -level $trclevel /var/log/
G001.trc]
…
```

```
sw_TcCloseTrace $Trchwnd
exit 0
```

・ 外部ファイルによる指定

トレースレベルを記述したファイルからトレースレベルを読み込むようにします。

例：

トレースレベル定義ファイルc:¥home¥etc¥trclevelにレベルが記述されている場合【Windows版】

```
# トレースレベル記述ファイルのオープン,読み込み,クローズ
if {[catch {
        set cid [open {c:¥home¥etc¥trclevel} r]
        gets $cid trclevel
        close $cid
        }]} {
# トレースレベル記述ファイルのエラー
puts stderr "Can not read {c:¥home¥etc¥trclevel}"
exit 1
}
# 指定値チェック
if {$trclevel < 0 || $trclevel > 2} {
        # 指定値エラー
        puts stderr "trace level err :$trclevel"
        exit 1
}
# トレースオープン
set Trchwnd [sw_TcOpenTrace -level $trclevel {c:¥var¥log
¥G001.trc}]
…
sw_TcCloseTrace $Trchwnd
exit 0
```

例：

トレースレベル定義ファイル“/home/etc/trclevel”にレベルが記述されている場合【UNIX版】

```
# トレースレベル記述ファイルのオープン,読み込み,クローズ
if {[catch {
        set cid [open /home/etc/trclevel r]
        gets $cid trclevel
        close $cid
        }]} {
# トレースレベル記述ファイルのエラー
puts stderr "Can not read /home/etc/trclevel"
exit 1
}
# 指定値チェック
if {$trclevel < 0 || $trclevel > 2} {
        # 指定値エラー
        puts stderr "trace level err :$trclevel"
        exit 1
}
# トレースオープン
set Trchwnd [sw_TcOpenTrace -level $trclevel /var/log/
G001.trc]
…
sw_TcCloseTrace $Trchwnd
exit 0
```

第17章　API情報

本章では、Systemwalker Operation Managerが提供するAPI(Application Programming Interface)情報について説明します。

注意

64bitのAPI使用時の留意事項

本章のプログラム例は、32bitのプログラム例です。64bitのAPIを利用する場合は、64bit API用のファイルに置き換えてください。

17.1　カレンダAPI

本節では、カレンダAPIについて説明します。

なお、旧バージョンのSystemwalker Operation Managerで提供された以下のAPIについては、本バージョンでもそのままご使用になれます。動作環境、注意事項および必要ファイルについては、本バージョンと同じです。

- カレンダ登録API(Mp_AddCalendar)
- カレンダ更新API(Mp_SetCalendar)
- カレンダ情報取得API(Mp_GetCalendar)

旧バージョンのSystemwalker Operation Managerとは、ここでは以下のバージョンを指しています。

Windows版の場合:

SystemWalker/OperationMGR V4.0L10/V4.0L20

Solaris版の場合:

SystemWalker/OperationMGR 4.0

カレンダAPI共通の動作環境、注意事項、および必要ファイルについての説明を以下に示します。

動作環境

当APIは、Systemwalker Operation Managerサーバ導入マシン上でのみ動作します。Systemwalker Operation Managerクライアントのみが導入されているマシン上では動作しません。

当APIは、Windows版、Solaris版、およびLinux版で使用できます。

注意事項

- コンパイル環境として、最適化は行わないでください。
- コンパイル環境として、コンパイラは、Visual C++(バージョン2022以降)を、ランタイムライブラリは“マルチスレッド(DLL)”を使用してください。【Windows版】
- コンパイルしたVisual C++バージョン以降のランタイムをAPI実行環境にインストールしてください。【Windows版】
- LIBファイルおよびINCLUDEファイルは、同じバージョン/レベルで提供されたものを使用してください。

必要ファイル

上記カレンダAPIを使用するには、以下のファイルが必要となります。

Windows 32bit用

LIB:

f3crhcap.lib

INCLUDE:

f3crhcap.h

Windows 64bit用

LIB:

f3crhcap_x64.lib

INCLUDE:

f3crhcap_x64.h

Solaris 32bit用

LIB：

libcalapi.so

INCLUDE：

f3crhcap.h

Solaris 64bit用

LIB：

libcalapi_64.so

INCLUDE：

f3crhcap_64.h

Linux 64bit用

LIB：

libcalapi_x64.so

INCLUDE：

f3crhcap_x64.h

カレンダの各APIはLIBファイルに格納し、各APIで使用する定数および構造体はINCLUDEファイルで宣言されています。

これらのファイルは、以下のディレクトリ配下に格納されています。

Windows版

LIB：

Systemwalker Operation Managerインストールディレクトリ
MpWalker.JM¥lib

INCLUDE：

Systemwalker Operation Managerインストールディレクトリ
MpWalker.JM¥include

Solaris 32bit用、Solaris 64bit用、Linux 64bit用共通

LIB：

/opt/FJSVjmcal/lib

INCLUDE：

/opt/FJSVjmcal/include

17.1.1　カレンダ名一覧取得API

カレンダ名一覧取得API(Mp_ListCalendar)について説明します。

記述形式

```
int Mp_ListCalendar (char *APL_name,
                     CAL_LIST *calendar_list) ;
```

機能説明

Mp_ListCalendar関数は、サーバに登録されているカレンダ名の一覧を取得します。

パラメタの説明

APL_name

要求元アプリケーション名(64バイト以内)の格納領域アドレスを指定します。

終わりはNULLを指定してください。“Mp_”および“_CAL_”の文字列で始まる名前は、使用しないでください。

calendar_list

カレンダ名取得情報構造体(CAL_LIST)のアドレスを指定します。

復帰値

復帰値	意 味	対 処
0	正常終了しました。	-
-7	カレンダー一覧領域が不足です。	“備考”に記載されている手順を参考に、カレンダー一覧領域のサイズを見直してください。
-10	カレンダサービスが動作していません。	カレンダサービスを起動してください。
-11	メモリ不足です。	以下の方法などで、使用できるメモリを増やしてください。 ・ほかのアプリケーションを終了する ・メモリを増やす

備考

カレンダ名格納域構造体を配列としてあらかじめ獲得します。そのアドレスと配列数(カレンダ名格納可能個数)をカレンダ名取得情報構造体に設定して、本関数を呼び出します。

カレンダ数が格納可能個数を超える場合は、カレンダ名取得情報構造体の“cal_num”に実際のカレンダ数が設定されるので、それ以上の配列数でカレンダ名格納域構造体を再獲得して、カレンダ名格納域構造体アドレス(pcal_name)およびカレンダ名格納可能個数(cal_area_num)を更新して、再び本関数を呼び出します。

カレンダ名取得情報構造体(CAL_LIST)の形式

```
typedef struct CAL_LIST_TAG {
      short         cal_area_num ;       // カレンダ名格納可能個数
      short         cal_num ;            // 実際のカレンダ数
      CAL_NAME_AREA *pcal_name ;         // カレンダ名格納域構造体アド
                                         // レス
} CAL_LIST ;
```

cal_area_num

カレンダ名格納域の領域数(カレンダの数で指定する)を指定します。

cal_num

サーバに格納されている実際のカレンダ数(不足の場合も設定される)が設定されます。

pcal_name

カレンダ名格納域構造体(CAL_NAME_AREA)のアドレスを指定します。

カレンダ名格納域構造体(CAL_NAME_AREA)の形式

```
typedef struct CAL_NAME_AREA_TAG {
        char   cal_name[25] ;            // カレンダ名格納域
} CAL_NAME_AREA ;
```

cal_name

カレンダ名が設定されます。

プログラム例

カレンダ名一覧取得APIのプログラム例を以下に示します。

```
#include <stdio.h>
#include <stdlib.h>
#include <malloc.h>
#include <string.h>

#include <f3crhcap.h>

#define CAL_NUM 20

extern void put_calendar_list( CAL_LIST * ) ;

int main() {

        int ret ;
        CAL_LIST calendar_list ;

        /* 初期化 */
        memset( &calendar_list, 0x00, sizeof( CAL_LIST) ) ;

        /* カレンダ名領域の獲得 */
        calendar_list.pcal_name = (CAL_NAME_AREA *) malloc (sizeof (CAL_NAME_AREA) * CAL_NUM) ;

        if ( calendar_list.pcal_name == NULL ) {
                printf( "Lack of memory !!¥n") ;
                exit ( 1 ) ;
        }

        /* カレンダ名領域数の設定 */
        calendar_list.cal_area_num = CAL_NUM ;

        /* カレンダー一覧の取得 */
        ret = Mp_ListCalendar( "Test Pro", &calendar_list) ;

        printf( "ret= %d  calendar num = %d¥n", ret, calendar_list.cal_num ) ;

        /* カレンダー一覧取得後の処理 */
```

```
        if ( ret == 0 )
                put_calendar_list( &calendar_list ) ;

        /* カレンダ名領域の返却 */
        free( calendar_list.pcal_name ) ;

        exit ( 0 ) ;
}
```

17.1.2 カレンダ登録API(Mp_AddCalendar2)

カレンダ登録API(Mp_AddCalendar2)について説明します。

記述形式

```
int Mp_AddCalendar2 (char *APL_name,
                char *cal_name,
                CALENDAR_WDAT2 *cal_dat2) ;
```

機能説明

Mp_AddCalendar2関数は、新規にカレンダを登録します。

第何週目の何曜日といった、年ごとに日付が変動する休日および振替休日も登録できます。

過去の日付は、設定できません。

パラメタの説明

APL_name

要求元アプリケーション名(64バイト以内)の格納領域アドレスを指定します。

終わりはNULLを指定してください。“Mp_”および“_CAL_”の文字列で始まる名前は、使用しないでください。

cal_name

登録するカレンダ名の格納領域アドレスを指定します。カレンダ名は24バイト以内で終わりはNULLを指定してください。

cal_dat2

カレンダ更新情報構造体(CALENDAR_WDAT2)のアドレスを指定します。

復帰値

復帰値	意 味	対 処
0	正常終了しました。	-
-2	同一カレンダ名がすでに存在しています。	指定するカレンダ名を変更してください。
-5	SYSTEM_CALENDARが指定されました。	SYSTEM_CALENDAR以外を指定してください。
-6	システム管理者(Administratorsグループに属するユーザまたはスーパーユーザ)ではありません。	システム管理者(Administratorsグループに属するユーザまたはスーパーユーザ)で実行してください。
-8	カレンダ情報に誤りがあります。	指定したカレンダ情報(カレンダ更新情報構造体(CALENDAR_WDAT2))を見直してください。
-10	カレンダサービス/カレンダデーモンが動作していません。	カレンダサービス/カレンダデーモンを起動してください。

復帰値	意 味	対 処
-11	メモリ不足です。	以下の方法などで、使用できるメモリを増やしてください。 ・ほかのアプリケーションを終了する ・メモリを増やす
-15	カレンダ名に指定できない文字を指定しました。	カレンダ名に使用できない文字を指定しないでください。 カレンダ名には、空白、半角カナ、ユーザ定義文字の他に以下の文字を使用することはできません。 ？￥＊"＜＞｜／：，；
-16	ファイル操作に失敗しました(writeエラー)。	ハードディスクの空き容量があるか確認してください。 ハードディスクに異常がないか確認してください。
-17	カレンダ名長が不当です。	カレンダ名を24バイト以内で指定してください。

備考

過去の日付は、すべて運用日として設定されます。

ジョブスケジューラでカレンダを引用する場合は、その日は運用日として計算されます。

カレンダ更新情報構造体(CALENDAR_WDAT2)の形式

32bitの場合

```
typedef struct CALENDAR_WDAT2_TAG {
  short year ;                          // 本年の年4桁
  short sub_holiday ;                   // 振替休日の有無を指定
  unsigned char holiday_w[12] ;         // 毎年の曜日情報(12カ月分)
  unsigned long holiday_d[12] ;         // 毎年の日付情報(12カ月分)
  unsigned char holiday_m[12][6] ;      // 毎年の週情報(12カ月分×6週
                                        // 分)
  short cal_dat[3][12][31] ;            // 3年分(year年から)のカレン
                                        // ダ情報

}CALENDAR_WDAT2 ;
```

64bitの場合

```
typedef struct CALENDAR_WDAT2_TAG {
  short year ;                          // 本年の年4桁
  short sub_holiday ;                   // 振替休日の有無を指定
  unsigned char holiday_w[12] ;         // 毎年の曜日情報(12カ月分)
  unsigned int holiday_d[12] ;          // 毎年の日付情報(12カ月分)
  unsigned char holiday_m[12][6] ;      // 毎年の週情報(12カ月分×6週
                                        // 分)
  short cal_dat[3][12][31] ;            // 3年分(year年から)のカレン
                                        // ダ情報

}CALENDAR_WDAT2 ;
```

year

3年分のカレンダ情報を設定する基準の年(3年分の先頭の年)を西暦4桁で設定します。

sub_holiday

振替休日が有効かどうかを指定します。以下の値を指定してください。

値	振替休日の有効／無効
0	無効
1	有効

holiday_w

毎年の曜日情報(12カ月分の配列)です。毎年の休日にする曜日を指定します。先頭から1ビット目を日曜日、2ビット目を月曜日とし、以降3ビット目から7ビット目まで順に、それぞれ火曜日から土曜日までに対応します。毎年、配列に指定した月のビットをオンにした曜日が休日になります。

holiday_d

毎年の日付情報(12カ月分の配列)です。毎年の休日にする日付を指定します。先頭から1ビット目を1日、2ビット目を2日とし、以降3ビット目から31ビット目まで順に、それぞれ3日から31日までに対応します。毎年、配列に指定した月のビットをオンにした日付が休日になります。

holiday_m

毎年の週情報(12カ月分×6週分の配列)です。第何週目の何曜日といった、毎年日付が変動する休日を指定します。1番目の配列を何月(1～12)、2番目の配列を何週目(1～5は第1～第5週目、6は最終週)として対応します。先頭から1ビット目を日曜日、2ビット目を月曜日とし、以降3ビット目から7ビット目まで順に、それぞれ火曜日から土曜日までに対応します。毎年、1番目の配列に指定した月の2番目の配列に指定した週のビットをオンにした曜日が休日になります。

cal_dat

yearで指定した年から3年分のカレンダ情報です。毎年の休日の情報を反映したものを指定します。以下の値を指定してください。

値	カレンダ情報
0	あり得ない日(2月30日など)
1	休日とする日
2	運用日とする日

プログラム例

カレンダ登録APIのプログラム例を以下に示します。

```
#include <stdio.h>
#include <stdlib.h>
#include <malloc.h>
#include <string.h>

#include <f3crhcap.h>

extern  void set_cal_dat2( CALENDAR_WDAT2 *) ;

int main() {

        int ret ;
        int count ;
        CALENDAR_WDAT2 cal_dat2 ;

        /*.......... 初期化 ..........*/
        memset( &cal_dat2, 0x00, sizeof( CALENDAR_WDAT2) ) ;
```

```
        /*..... カレンダ情報の設定 .....*/
        /* 今年を設定 */
        cal_dat2.year = 2000 ;

        /* 振替休日を有効にする */
        cal_dat2.sub_holiday = 1 ;

        /* 毎週土日を休日にする */
        for ( count = 0 ; count < 12 ; count++ ) {
                cal_dat2.holiday_w[count] = (unsigned char)0x82 ;
        }

        /* 3年分のカレンダ情報を設定する */
        set_cal_dat2( &cal_dat2 ) ;


        /* カレンダの登録 */
        ret = Mp_AddCalendar2( "Test Pro", "calendar_1", &cal_dat2 ) ;

        printf( "ret= %d¥n", ret ) ;

        exit ( 0 ) ;
}
void set_cal_dat2(CALENDAR_WDAT2* cal_dat2)
{
   /* 3年分のカレンダ情報を
       cal_dat2->cal_dat
       に設定する */
}
```

17.1.3　カレンダ更新API(Mp_SetCalendar2)

カレンダ更新API(Mp_SetCalendar2)について説明します。

記述形式

```
int Mp_SetCalendar2 (char *APL_name,
                     char *cal_name,
                     CALENDAR_WDAT2 *cal_dat2) ;
```

機能説明

Mp_SetCalendar2関数は、カレンダ情報の更新(指定カレンダが存在しない場合は、登録として扱う)を行います。第何週目の何曜日といった、年ごとに日付が変動する休日および振替休日の情報も更新できます。

SYSTEM_CALENDARについては、更新できません。

また、過去の日付は、設定できません。

パラメタの説明

APL_name

要求元アプリケーション名(64バイト以内)の格納領域アドレスを指定します。

終わりはNULLを指定してください。“Mp_”および“_CAL_”の文字列で始まる名前は、使用しないでください。

cal_name

更新するカレンダ名の格納領域アドレスを指定します。カレンダ名は、24バイト以内で終わりにNULLを指定してください。

cal_dat2

カレンダ更新情報構造体(CALENDAR_WDAT2)のアドレスを指定します。

カレンダ更新情報構造体については、“17.1.2 カレンダ登録API(Mp_AddCalendar2)”を参照してください。

復帰値

復帰値	意味	対処
0	正常終了しました。	-
-5	SYSTEM_CALENDARが指定されました。	SYSTEM_CALENDAR以外を指定してください。
-6	システム管理者(Administratorsグループに属するユーザまたはスーパーユーザ)ではありません。	システム管理者(Administratorsグループに属するユーザまたはスーパーユーザ)で実行してください。
-8	カレンダ情報に誤りがあります。	指定したカレンダ情報(カレンダ更新情報構造体(CALENDAR_WDAT2))を見直してください。
-10	カレンダサービス/カレンダデーモンが動作していません。	カレンダサービス/カレンダデーモンを起動してください。
-11	メモリ不足です。	以下の方法などで、使用できるメモリを増やしてください。 ・ほかのアプリケーションを終了する ・メモリを増やす
-15	カレンダ名に指定できない文字を指定しました。	カレンダ名に使用できない文字を指定しないでください。 カレンダ名には、空白、半角カナ、ユーザ定義文字の他に以下の文字を使用することはできません。 ？￥＊"＜＞｜／：，；
-16	ファイル操作に失敗しました(writeエラー)。	ハードディスクの空き容量があるか確認してください。ハードディスクに異常がないか確認してください。
-17	カレンダ名長が不当です。	カレンダ名を24バイト以内で指定してください。

プログラム例

カレンダ更新APIのプログラム例を以下に示します。

```
#include <stdio.h>
#include <stdlib.h>
#include <malloc.h>
#include <string.h>

#include <f3crhcap.h>

extern  void set_cal_dat2( CALENDAR_WDAT2 *) ;

int main() {

        int ret ;
        int count ;
        CALENDAR_WDAT2 cal_dat2 ;
```

```
        /*.........初期化 ..........*/
        memset( &cal_dat2, 0x00, sizeof( CALENDAR_WDAT2) ) ;

        /*.........カレンダ情報の設定 ..........*/
        /*今年を設定 */
        cal_dat2.year = 2000 ;

        /*  振替休日を有効にする */
        cal_dat2.sub_holiday = 1 ;

        /*毎週日曜を休日にする */
        for ( count = 0 ; count < 12 ; count++ ){
                cal_dat2.holiday_w[count] = (unsigned char)0x80 ;
        }

        /* 3年分のカレンダ情報を設定する */
        set_cal_dat2( &cal_dat2 ) ;


        /*カレンダの更新 */
        ret = Mp_SetCalendar2( "Test Pro", "calendar_1", &cal_dat2 ) ;

        printf( "ret= %d¥n", ret ) ;

        exit ( 0 ) ;
}
void set_cal_dat2(CALENDAR_WDAT2* cal_dat2)
{
   /* 3年分のカレンダ情報を
       cal_dat2->cal_dat
       に設定する */
}
```

17.1.4　カレンダ更新API(Mp_SetCalendar3)

カレンダ更新API(Mp_SetCalendar3)について説明します。

記述形式

```
int Mp_SetCalendar3 (char *APL_name,
                char *cal_name,
                CALENDAR_WDAT3 *cal_dat3) ;
```

機能説明

Mp_SetCalendar3関数は、カレンダ情報の更新(指定したカレンダが存在しない場合は、新規登録として扱う)を行います。第何週目の何曜日といった、年ごとに日付が変動する休日、振替休日の情報、およびジョブネットへの反映開始日も更新できます。

SYSTEM_CALENDARについては、更新できません。

反映開始日を指定しない場合は、すべての日付が反映対象となります。

また、過去の日付は、設定できません。

パラメタの説明

APL_name

要求元アプリケーション名(64バイト以内)の格納領域アドレスを指定します。

終わりはNULLを指定してください。“Mp_”および“_CAL_”の文字列で始まる名前は、使用しないでください。

cal_name

更新するカレンダ名(24バイト以内)の格納領域アドレスを指定します。終わりはNULLを指定してください。

cal_dat3

カレンダ更新情報構造体(CALENDAR_WDAT3)のアドレスを指定します。

復帰値

復帰値	意 味	対 処
0	正常終了しました。	-
-5	SYSTEM_CALENDARが指定されました。	SYSTEM_CALENDAR以外を指定してください。
-6	システム管理者(Administratorsグループに属するユーザまたはスーパーユーザ)ではありません。	システム管理者(Administratorsグループに属するユーザまたはスーパーユーザ)で実行してください。
-8	カレンダ情報に誤りがあります。	指定したカレンダ情報(カレンダ更新情報構造体(CALENDAR_WDAT3))を見直してください。
-10	カレンダサービス/カレンダデーモンが動作していません。	カレンダサービス/カレンダデーモンを起動してください。
-11	メモリ不足です。	以下の方法などで、使用できるメモリを増やしてください。 ・ほかのアプリケーションを終了する ・メモリを増やす
-15	カレンダ名に指定できない文字を指定しました。	カレンダ名に使用できない文字を指定しないでください。 カレンダ名には、空白、半角カナ、ユーザ定義文字の他に以下の文字を使用することはできません。 ？￥＊"＜＞｜／：，；
-16	ファイル操作に失敗しました(writeエラー)。	ハードディスクの空き容量があるか確認してください。 ハードディスクに異常がないか確認してください。
-17	カレンダ名長が不当です。	カレンダ名を24バイト以内で指定してください。

カレンダ更新情報構造体(CALENDAR_WDAT3)の形式

32bitの場合

```
typedef struct CALENDAR_WDAT3_TAG {
 short year ;                        // 本年の年4桁
 short sub_holiday ;                 // 振替休日の有無を指定
 unsigned char holiday_w[12] ;       // 毎年の曜日情報(12カ月分)
 unsigned long holiday_d[12] ;       // 毎年の日付情報(12カ月分)
 unsigned char holiday_m[12][6] ;    // 毎年の週情報(12カ月分×6週分)
 short cal_dat[3][12][31] ;          // 3年分(year年から)のカレンダ情報
 short start_y ;                     // 反映開始日の年4桁
 short start_m ;                     // 反映開始日の月(1～12)
 short start_d ;                     // 反映開始日の日(1～31)
}CALENDAR_WDAT3 ;
```

64bitの場合

```
typedef struct CALENDAR_WDAT3_TAG {
 short year ;                        // 本年の年4桁
```

```
 short sub_holiday ;                // 振替休日の有無を指定
 unsigned char holiday_w[12] ;      // 毎年の曜日情報(12カ月分)
 unsigned int holiday_d[12] ;       // 毎年の日付情報(12カ月分)
 unsigned char holiday_m[12][6] ;   // 毎年の週情報(12カ月分×6週分)
 short cal_dat[3][12][31] ;         // 3年分(year年から)のカレンダ情報
 short start_y ;                    // 反映開始日の年4桁
 short start_m ;                    // 反映開始日の月(1～12)
 short start_d ;                    // 反映開始日の日(1～31)
}CALENDAR_WDAT3 ;
```

year

3年分のカレンダ情報を設定する基準の年(3年分の先頭の年)を西暦4桁で設定します。

sub_holiday

振替休日が有効かどうかを指定します。以下の値を指定してください。

値	振替休日の有効／無効
0	無効
1	有効

holiday_w

毎年の曜日情報(12カ月分の配列)です。毎年の休日にする曜日を指定します。先頭から1ビット目を日曜日、2ビット目を月曜日とし、以降3ビット目から7ビット目まで順に、それぞれ火曜日から土曜日までに対応します。毎年、配列に指定した月のビットをオンにした曜日が休日になります。

holiday_d

毎年の日付情報(12カ月分の配列)です。毎年の休日にする日付を指定します。先頭から1ビット目を1日、2ビット目を2日とし、以降3ビット目から31ビット目まで順に、それぞれ3日から31日までに対応します。毎年、配列に指定した月のビットをオンにした日付が休日になります。

holiday_m

毎年の週情報(12カ月分×6週分の配列)です。第何週目の何曜日といった、毎年日付が変動する休日を指定します。1番目の配列を何月(1～12)、2番目の配列を何週目(1～5は第1～第5週目、6は最終週)として対応します。先頭から1ビット目を日曜日、2ビット目を月曜日とし、以降3ビット目から7ビット目まで順に、それぞれ火曜日から土曜日までに対応します。毎年、1番目の配列に指定した月の2番目の配列に指定した週のビットをオンにした曜日が休日になります。

cal_dat

yearで指定した年から3年分のカレンダ情報です。毎年の休日の情報を反映したものを指定します。以下の値を指定してください。

値	カレンダ情報
0	あり得ない日(2月30日など)
1	休日とする日
2	運用日とする日

start_y

反映開始日の年(現在から2年後まで)を西暦4桁で設定します。

反映開始日を指定しない場合は、0を指定してください。

start_m

反映開始日の月(1～12)を設定します。

反映開始日を指定しない場合は、0を指定してください。

start_d

反映開始日の日(1～31)を設定します。

反映開始日を指定しない場合は、0を指定してください。

プログラム例

カレンダ更新APIのプログラム例を以下に示します。

```
#include <stdio.h>
#include <string.h>
#include <f3crhcap.h>
extern void set_cal_dat3(CALENDAR_WDAT3*);

int main( int argc,char** argv)
{
/*.....変数宣言..... */
        int ret ;
        int count ;
        CALENDAR_WDAT3 cal_dat3 ;

        /*.....初期化 .....*/
        memset( &cal_dat3, 0x00, sizeof( CALENDAR_WDAT3) ) ;

        /*.....カレンダ情報の設定 .....*/
        /*今年を設定 */
        cal_dat3.year = 2006 ;

        /*  振替休日を有効にする */
        cal_dat3.sub_holiday = 1 ;

        /*毎週土日を休日にする */
        for ( count = 0 ; count < 12 ; count++ ){
                cal_dat3.holiday_w[count] = (unsigned char)0x82 ;
        }

        /* 3年分のカレンダ情報を設定する */
        set_cal_dat3(&cal_dat3);

        /* 反映開始日の設定(2006/08/09から反映を開始する。) */
        cal_dat3.start_y = 2006 ;
        cal_dat3.start_m = 8 ;
        cal_dat3.start_d = 9 ;

        /*カレンダの更新 */
        ret = Mp_SetCalendar3( "Test Pro", "calendar_1", &cal_dat3 ) ;

        printf( "ret= %d¥n", ret ) ;

        return 0 ;
}
void set_cal_dat3(CALENDAR_WDAT3* cal_dat3)
{
   /* 3年分のカレンダ情報を
       cal_dat3->cal_dat
```

```
        に設定する */
}
```

17.1.5　カレンダ情報取得API

カレンダ情報取得API(Mp_GetCalendar2)について説明します。

記述形式

```
int Mp_GetCalendar2 (char *APL_name,
                     char *cal_name,
                     short mode,
                     CALENDAR_RDAT2 *cal_dat2) ;
```

機能説明

Mp_GetCalendar2関数は、カレンダ情報を取得します。

第何週目の何曜日といった、年ごとに変動する休日および振替休日の情報も取得できます。

Windows版でSYSTEM_CALENDARの情報を取得する場合は、運用パターンの種別まで取得するのか、運用日または休日のカレンダ情報のみを取得するのかを指定できます。【Windows版】

パラメタの説明

APL_name

要求元アプリケーション名(64バイト以内)の格納領域アドレスを指定します。

終わりはNULLを指定してください。“Mp_”および“_CAL_”の文字列で始まる名前は、使用しないでください。

cal_name

カレンダ情報を取得するカレンダ名の格納領域アドレスを指定します。カレンダ名は、24バイト以内で終わりにNULLを指定してください。

mode

【Windows版】

パターン種別を取得するかの情報を指定します(SYSTEM_CALENDARの情報を取得する場合に有効)。

0:

パターン種別を取得しません。

1:

パターン種別を取得します。

【UNIX版】

0を指定します。

cal_dat2

カレンダ参照情報構造体(CALENDAR_RDAT2)のアドレスを指定します。

復帰値

復帰値	意 味	対 処
0	正常終了しました。	-
-1	カレンダが存在しません。	存在するカレンダのカレンダ名を指定してください。

復帰値	意 味	対 処
-10	カレンダサービス/カレンダデーモンが動作していません。	カレンダサービス/カレンダデーモンを起動してください。
-11	メモリ不足です。	以下の方法などで、使用できるメモリを増やしてください。 ・ほかのアプリケーションを終了する ・メモリを増やす
-16	ファイル操作に失敗しました(readエラー)。	ハードディスクの空き容量があるか確認してください。ハードディスクに異常がないか確認してください。
-17	カレンダ名長が不当です。	カレンダ名を24バイト以内で指定してください。

カレンダ参照情報構造体(CALENDAR_RDAT2)の形式

32bitの場合

```
typedef struct CALENDAR_RDAT2_TAG {
  short year ;                      // 本年の年4桁
  short sub_holiday ;               // 振替休日の有無を指定
  unsigned char holiday_w[12] ;     // 毎年の曜日情報
  unsigned long holiday_d[12] ;     // 毎年の日付情報
  unsigned char holiday_m[12][6] ;  // 毎年の週情報(12カ月分×6週分)
  short cal_dat[4][12][31] ;        // 4年分(year-1年から)のカレン
                                    //ダ情報
} CALENDAR_RDAT2 ;
```

64bitの場合

```
typedef struct CALENDAR_RDAT2_TAG {
  short year ;                      // 本年の年4桁
  short sub_holiday ;               // 振替休日の有無を指定
  unsigned char holiday_w[12] ;     // 毎年の曜日情報
  unsigned int holiday_d[12] ;      // 毎年の日付情報
  unsigned char holiday_m[12][6] ;  // 毎年の週情報(12カ月分×6週分)
  short cal_dat[4][12][31] ;        // 4年分(year-1年から)のカレン
                                    //ダ情報
} CALENDAR_RDAT2 ;
```

year

設定される4年分のカレンダ情報の基準の年(4年分の2年目)が西暦4桁で設定されます。通常は、今年が基準として設定されます。

sub_holiday

振替休日が有効かどうかが設定されます。以下の値が設定されます。

値	振替休日の有効／無効
0	無効
1	有効

holiday_w

毎年の曜日情報(12カ月分の配列)です。毎年の休日にする曜日が設定されます。先頭から1ビット目を日曜日、2ビット目を月曜日とし、以降3ビット目から7ビット目まで順に、それぞれ火曜日から土曜日までに対応します。毎年、配列に指定した月のビットをオンにした曜日が休日になります。

holiday_d

毎年の日付情報(12カ月分の配列)です。毎年の休日にする日付が設定されます。先頭から1ビット目を1日、2ビット目を2日とし、以降3ビット目から31ビット目まで順に、それぞれ3日から31日までに対応します。毎年、配列に指定した月のビットをオンにした日付が休日になります。

holiday_m

毎年の週情報(12カ月分×6週分の配列)です。第何週目の何曜日といった、毎年日付が変動する休日が設定されます。1番目の配列を何月(1～12)、2番目の配列を何週目(1～5は第1～第5週目、6は最終週)として対応します。先頭から1ビット目を日曜日、2ビット目を月曜日とし、以降3ビット目から7ビット目まで順に、それぞれ火曜日から土曜日までに対応します。毎年、1番目の配列に指定した月の2番目の配列に指定した週のビットをオンにした曜日が休日になります。

cal_dat

yearの前年から4年分のカレンダ情報が設定されます。設定される情報を以下に示します。

【Windows版】

値	カレンダ情報
0	あり得ない日(2月30日など)
1	休日
2	運用日(modeで0指定時)
3	標準パターン(modeで1指定時)
4	特異パターン1(modeで1指定時)
5	特異パターン2(modeで1指定時)
6	特異パターン3(modeで1指定時)
7	特異パターン4(modeで1指定時)
8	特異パターン5(modeで1指定時)
9	特異パターン6(modeで1指定時)
10	特異パターン7(modeで1指定時)
11	特異パターン8(modeで1指定時)
12	個別1(modeで1指定時)
11+n	個別ｎ(modeで1指定時)
2xxx	休日／個別(modeで1指定時) ｘｘｘは個別のＩＤに対応

【UNIX版】

値	カレンダ情報
0	あり得ない日(2月30日など)
1	休日とする日
2	運用日とする日

プログラム例

カレンダ情報取得APIのプログラム例を以下に示します。

```
#include <stdio.h>
#include <stdlib.h>
#include <malloc.h>
#include <string.h>

#include <f3crhcap.h>

extern  void put_cal_dat2( CALENDAR_RDAT2*) ;

int main() {

        int ret ;
        CALENDAR_RDAT2 cal_dat2 ;

        /*.......... 初期化 ..........*/
        memset( &cal_dat2, 0x00, sizeof( CALENDAR_RDAT2) ) ;

        /* カレンダ情報の取得 */
        ret = Mp_GetCalendar2( "Test Pro", "calendar_1", 0, &cal_dat2 ) ;

        printf( "ret= %d¥n", ret ) ;

        if ( ret == 0 )
                put_cal_dat2( &cal_dat2 ) ;

        exit ( 0 ) ;
}
```

17.1.6 日変わり時刻取得API

日変わり時刻取得API(Mp_GetDayStartTime)について説明します。

記述形式

```
int Mp_GetDayStartTime (char *APL_name,
                        char *start_time) ;
```

機能説明

Mp_GetDayStartTime関数は、システムの日変わり時刻を取得します。

パラメタの説明

APL_name

要求元アプリケーション名(64バイト以内)の格納領域アドレスを指定します。

終わりはNULLを指定してください。“Mp_”および“_CAL_”の文字列で始まる名前は、使用しないでください。

start_time

日変わり時刻の格納領域アドレス(HHMM形式の5バイト領域)を指定します。最後にNULLが設定されます。

復帰値

復帰値	意 味	対 処
0	正常終了しました。	-
-10	カレンダサービス/カレンダデーモンが動作していません。	カレンダサービス/カレンダデーモンを起動してください。

プログラム例

日変わり時刻取得APIのプログラム例を以下に示します。

```
#include <stdio.h>
#include <stdlib.h>
#include <malloc.h>
#include <string.h>

#include <f3crhcap.h>

int main() {

        int ret ;
        char start_time[5] ;

        /*.......... 初期化 ..........*/
        memset( start_time, 0x00, sizeof( start_time) ) ;

        /* 日変わり時刻の取得 */
        ret = Mp_GetDayStartTime( "Test Pro", start_time ) ;

        printf( "ret= %d  start_time=%s¥n", ret, start_time ) ;

        exit ( 0 ) ;
}
```

17.2 ジョブスケジューラAPI

本節では、ジョブスケジューラが提供するAPIについて説明します。

ジョブスケジューラAPI共通の動作環境および注意事項を以下に示します。

動作環境

当APIは、Systemwalker Operation Managerサーバ導入マシン上でのみ動作します。Systemwalker Operation Managerクライアントのみが導入されているマシン上では動作しません。

ジョブネット操作API、ジョブネット操作(起動パラメタ指定)API、グループ操作APIおよびジョブネット起動時刻変更APIは、Windows版、Solaris版、Linux版で使用できます。

EE GEE

ジョブネット操作API/EE、ジョブネット操作(起動パラメタ指定)API/EE、グループ操作API/EEおよびジョブネット起動時刻変更API/EEは、Windows版、Solaris版、Linux版で使用できます。

注意事項

- コンパイル環境として、コンパイラは、Visual C++(バージョン2022以降)を、ランタイムライブラリは“マルチスレッド(DLL)”を使用してください。【Windows版】
- コンパイルしたVisual C++バージョン以降のランタイムをAPI実行環境にインストールしてください。【Windows版】
- LIBファイルおよびINCLUDEファイルは、同じバージョン/レベルで提供されたものを使用してください。

- ジョブスケジューラAPIを、同一プロセス内の複数のスレッドから、同時に呼び出さないでください。
- 標準出力、標準エラー出力がクローズされている環境から、APIを実行しないでください。
- ジョブスケジューラAPIは、Operation Managerサーバの処理と同期を取っていません。そのため、APIの完了は、サーバでの処理の完了を保証するものではありません。
  また、APIが実行されてからサーバで処理が受け付けられるまで、若干のタイムラグが発生することがあります。
- 拡張ユーザ管理機能が有効な場合は、APIを実行したOSユーザにより、以下のようにアクセス権の認証が変わります。【UNIX版】

  ー 実行したOSユーザがシステム管理者の場合

  Operation Managerの管理者の権限(すべてのプロジェクトに対して更新権あり)で実行されます。

  ー 実行したOSユーザがプロジェクトの所有者の場合

  所有者となっているプロジェクトに対して更新権を持っているものとして実行されます。

  ー 実行したOSユーザがシステム管理者でなく、かつ、プロジェクトの所有者でない場合

  プロジェクトにアクセス権を持つすべてのOperation Managerユーザに対応するOSユーザの中に、APIを実行したOSユーザが含まれ、権限があれば、APIは実行されます。APIを実行したOSユーザが複数のOperation Managerユーザに対応づけられている場合、OSユーザは、対応づけられた複数のOperation Managerユーザのアクセス権の中で一番強い権限(更新権＞登録権・操作権＞参照権)を持ちます。なお、OSユーザに対応づけられたOperation Managerユーザに、登録権と操作権が設定されている場合、OSユーザは、両方の権限を持ちます。

17.2.1 ジョブネット操作API

ジョブネット操作API(Mp_JobschControl)について説明します。

記述形式

【Windows版】

```
#include "f3csbJobschControl.h"
int Mp_JobschControl (char *project,
                      char *netname,
                      int action,
                      int rerunopt) ;
```

【UNIX版】

```
#include "jobschapi.h"
int Mp_JobschControl (char *project,
                      char *netname,
                      int action,
                      int rerunopt) ;
```

機能説明

本APIは、jobschcontrolコマンドをC言語インタフェースから利用するためのインタフェースです。

パラメタの説明

project

操作するジョブネットの存在するプロジェクト名を指定します。なお、API実行ユーザの利用できるプロジェクトのプロジェクト名のみ指定できます。

API実行ユーザの利用できるプロジェクトのうちジョブネットが登録されているプロジェクトが1つしかない場合に限り省略できます。省略する場合はNULLを指定します。

netname

操作するジョブネット名を指定します。省略することはできません。

action

ジョブネットの操作方法を指定します。

1:

start(起動)

2:

cancel(強制終了)

3:

rerun(再起動)

4:

pause(停止)

5:

continue(停止解除)

6:

disable(無効)

7:

enable(無効解除)

8:

cope(確認)

9:

reinstate(終了取消)

10:

revoke(持越解除)

rerunopt

actionで3(rerun)を選択した場合のオプションを指定します。actionで3(rerun)を選択しなかった場合は、0を指定します。

0:

通常の再起動を行います。異常終了したジョブから起動します。異常終了したジョブより前のジョブに再起動ポイントが設定されている場合には、そのジョブから起動します。

1:

先頭ジョブからジョブネットを再起動します。

2:

異常終了したジョブに対して「次の再起動のみ無効」を設定してジョブネットを再起動します。

(異常終了したジョブを飛ばして、次のジョブから起動します。)

強制終了したジョブについても、異常終了したジョブと同様に扱います。

復帰値

復帰値	意 味	対 処
0	正常終了しました。	-
4	ジョブネットが操作できない状態です。 (ジョブネットが停止や無効状態の時に起動操作を実施した場合など)	ジョブネットの状態が変わってから再度実行してください。
8	ジョブスケジューラサービス/ジョブスケジューラデーモンが動作していません。	ジョブスケジューラサービス/ジョブスケジューラデーモンを起動して、再度実行してください。
12	パラメタの指定に誤りがあります。または、ユーザの権限に誤りがあります。(注)	以下のいずれかの処置を行ってください。 ・正しいパラメタ(プロジェクト名/ジョブネット名/操作方法/再起動オプション)を設定する ・実行ユーザに、プロジェクトの操作権以上のアクセス権を設定する
16	致命的なエラーが発生しました。	システムエラーが発生した可能性があります。イベントログ/SYSLOGを確認し、システムに異常がないか確認してください。異常があった場合は、保守情報収集ツールで“ジョブスケジューラ”の情報を採取し、富士通技術員に連絡してください。異常がない場合は一時的な問題と思われるため、再度実行してください。

注)
パラメタ指定の誤りの例：

- パラメタnetnameに指定したジョブネットが存在しない。
- パラメタactionに指定可能な値以外が指定されている。
- パラメタprojectがNULLで、かつ、コマンド実行ユーザがアクセスできるプロジェクトが複数ある。

ユーザ権限の誤りの例：

- コマンド実行ユーザにアクセス権(操作権)が不足している。
- API実行ユーザが、swadminグループに属していない。

必要ファイル

ジョブネット操作APIを使用するには、以下のファイルが必要となります。

Windows 32bit用

LIB:

f3csbJobschControl.lib

INCLUDE:

f3csbJobschControl.h

Windows 64bit用

LIB:

f3csbJobschControl_x64.lib

INCLUDE:

f3csbJobschControl_x64.h

Solaris 32bit用、Linux 32bit用共通

LIB:

libjobsch.so

INCLUDE:

jobschapi.h

Solaris 64bit用

LIB:

libjobsch_64.so

INCLUDE:

jobschapi_64.h

Linux 64bit用

LIB:

libjobsch_x64.so

INCLUDE:

jobschapi_x64.h

ジョブネット操作APIはLIBに格納されており、ジョブネット操作APIで使用する定数および構造体はINCLUDEファイルに宣言されています。

これらのファイルは以下のディレクトリに格納されています。

Windows版

LIB：

Systemwalker Operation Managerインストールディレクトリ
MpWalker.JM¥lib

INCLUDE：

Systemwalker Operation Managerインストールディレクトリ
MpWalker.JM¥include

Solaris版、Linux版共通

LIB:

/opt/FJSVJOBSC/usr/lib

INCLUDE:

/opt/FJSVJOBSC/usr/include

プログラム例

ジョブネット操作APIのプログラム例を以下に示します。

【Windows版】

```
#include <windows.h>
#include <stdio.h>
#include <stdlib.h>
#include "f3csbJobschControl.h"

int main()
{
    int ret;

    ret = Mp_JobschControl((char*)"user1",          // project name
                           (char*)"net001",         // jobnet name
                           1,               // action : start
```

```
                          0);              // rerun opt : none

    if (ret == 0){
        printf("jobnet(net001) is controlled normally.¥n");
    }
    else{
        printf("ERROR: return code = %d¥n", ret);
        exit(1);
    }
    exit(0);
}
```

【UNIX版】

```
#include <stdio.h>
#include <stdlib.h>
#include "jobschapi.h"

int main()
{
    int ret;

    ret = Mp_JobschControl("user1",          /* project name */
                           "net001",         /* jobnet name */
                           1,                /* action : start */
                           0);               /* rerun opt : none */

    if (ret == 0){
        printf("jobnet(net001) was normally controlled.¥n");
    }
    else{
        printf("ERROR: return code = %d¥n", ret);
        exit(1);
    }
    exit(0);
}
```

17.2.2 ジョブネット操作API/EE

ジョブネット操作API/EE(Mp_JobschControlEx)について説明します。

記述形式

【Windows版】

```
#include "f3csbJobschControlEx.h"
int Mp_JobschControlEx (char *project,
                        char *netname,
                        int action,
                        int rerunopt,
                        int system_num) ;
```

【UNIX版】

```
#include "jobschapiex.h"
int Mp_JobschControlEx (char *project,
                        char *netname,
                        int action,
                        int rerunopt,
                        int system_num) ;
```

機能説明

本APIは、Systemwalker Operation Manager EEおよびGEEで、jobschcontrolコマンドをC言語インタフェースから利用するためのインタフェースです。

パラメタの説明

project

操作するジョブネットの存在するプロジェクト名を指定します。なお、API実行ユーザの利用できるプロジェクトのプロジェクト名のみ指定できます。API実行ユーザの利用できるプロジェクトのうちジョブネットが登録されているプロジェクトが1つしかない場合に限り省略できます。省略する場合はNULLを指定します。

netname

操作するジョブネット名を指定します。省略することはできません。

action

ジョブネットの操作方法を指定します。

1:

start(起動)

2:

cancel(強制終了)

3:

rerun(再起動)

4:

pause(停止)

5:

continue(停止解除)

6:

disable(無効)

7:

enable(無効解除)

8:

cope(確認)

9:

reinstate(終了取消)

10:

revoke(持越解除)

rerunopt

actionで3(rerun)を選択した場合のオプションを指定します。actionで3(rerun)を選択しなかった場合は、0を指定します。

0:

通常の再起動を行います。異常終了したジョブから起動します。異常終了したジョブより前のジョブに再起動ポイントが設定されている場合には、そのジョブから起動します。

1:

先頭ジョブからジョブネットを再起動します。

2:

異常終了したジョブに対して「次の再起動のみ無効」を設定してジョブネットを再起動します。

(異常終了したジョブを飛ばして、次のジョブから起動します。)

強制終了したジョブについても、異常終了したジョブと同様に扱います。

system_num

操作対象のサブシステムを、0から9の範囲の番号で指定します。

復帰値

復帰値	意 味	対 処
0	正常終了しました。	-
4	ジョブネットが操作できない状態です。 (ジョブネットが停止や無効状態の時に起動操作を実施した場合など)	ジョブネットの状態が変わってから再度実行してください。
8	ジョブスケジューラサービス/ジョブスケジューラデーモンが動作していません。	ジョブスケジューラサービス/ジョブスケジューラデーモンを起動して、再度実行してください。
12	パラメタの指定に誤りがあります。または、ユーザの権限に誤りがあります。(注)	以下のいずれかの処置を行ってください。 ・正しいパラメタ(プロジェクト名/ジョブネット名/操作方法/再起動オプション/サブシステム番号)を設定する ・実行ユーザに、プロジェクトの操作権以上のアクセス権を設定する
16	致命的なエラーが発生しました。	システムエラーが発生した可能性があります。イベントログ/SYSLOGを確認し、システムに異常がないか確認してください。異常があった場合は、保守情報収集ツールで“ジョブスケジューラ”の情報を採取し、富士通技術員に連絡してください。異常がない場合は一時的な問題と思われるため、再度実行してください。

注)
パラメタ指定の誤りの例：

- パラメタnetnameに指定したジョブネットが存在しない。
- パラメタactionに指定可能な値以外が指定されている。
- パラメタprojectがNULLで、かつ、コマンド実行ユーザがアクセスできるプロジェクトが複数ある。

ユーザ権限の誤りの例：

- コマンド実行ユーザにアクセス権(操作権)が不足している。
- API実行ユーザが、swadminグループに属していない。

必要ファイル

ジョブネット操作APIを使用するには、以下のファイルが必要となります。

Windows 32bit用

LIB:

f3csbJobschControlEx.lib

INCLUDE:

f3csbJobschControlEx.h

Windows 64bit用

LIB:

f3csbJobschControlEx_x64.lib

INCLUDE:

f3csbJobschControlEx_x64.h

Solaris 32bit用、Linux 32bit用共通

LIB:

libjobschex.so

INCLUDE:

jobschapiex.h

Solaris 64bit用

LIB:

libjobschex_64.so

INCLUDE:

jobschapiex_64.h

Linux 64bit用

LIB:

libjobschex_x64.so

INCLUDE:

jobschapiex_x64.h

ジョブネット操作APIはLIBに格納されており、ジョブネット操作APIで使用する定数および構造体はINCLUDEファイルに宣言されています。

これらのファイルは以下のディレクトリに格納されています。

Windows版

LIB：

Systemwalker Operation Managerインストールディレクトリ
MpWalker.JM¥lib

INCLUDE：

Systemwalker Operation Managerインストールディレクトリ
MpWalker.JM¥include

Solaris版、Linux版共通

LIB:

/opt/FJSVJOBSC/usr/lib

INCLUDE:

/opt/FJSVJOBSC/usr/include

プログラム例

ジョブネット操作API/EEのプログラム例を以下に示します。

【Windows版】

```
#include <windows.h>
#include <stdio.h>
#include <stdlib.h>
#include "f3csbJobschControlEx.h"

int main()
{
    int ret;

    ret = Mp_JobschControlEx((char*)"user1",         // project name
                             (char*)"net001",        // jobnet name
                             1,               // action : start
                             0,               // rerun opt : none
                             1);              // sub system number

    if (ret == 0) {
        printf("jobnet(net001) is controlled normally.¥n");
    }
    else{
        printf("ERROR: return code = %d¥n", ret);
        exit(1);
    }
    exit(0);
}
```

【UNIX版】

```
#include <stdio.h>
#include <stdlib.h>
#include "jobschapiex.h"

int main ()
{
    int ret;

    ret = Mp_JobschControlEx ("user1",         /* project name      */
                              "net1",          /* jobnet name       */
                              1,               /* action : start    */
                              0,               /* rerun opt : none  */
                              1);              /* sub system number */

    if (ret == 0) {
        printf ("jobnet(net1) was normally controlled.¥n");
    }
    else{
        printf ("ERROR: return code = %d¥n", ret);
        exit(1);
    }
    exit(0);
}
```

17.2.3 ジョブネット操作(起動パラメタ指定)API

ジョブネット操作(起動パラメタ指定)API(Mp_JobschControlParam)について説明します。

記述形式

【Windows版】

```
#include "f3csbJobschControl.h"
int Mp_JobschControlParam (char *project,
                    char *netname,
                    char *parameter,
                    int action,
                    int rerunopt) ;
```

【UNIX版】

```
#include "jobschapi.h"
int Mp_JobschControlParam (char *project,
                    char *netname,
                    char *parameter,
                    int action,
                    int rerunopt) ;
```

機能説明

本APIは、jobschcontrolコマンドをC言語インタフェースから利用するためのインタフェースです。本APIを利用すると、起動対象のジョブネットの各ジョブにパラメタを渡すことができます。

ジョブ実行制御属性のジョブネットのみ利用できます。

パラメタの説明

project

操作するジョブネットの存在するプロジェクト名を指定します。なお、API実行ユーザの利用できるプロジェクトのプロジェクト名のみ指定できます。

API実行ユーザの利用できるプロジェクトのうちジョブネットが登録されているプロジェクトが1つしかない場合に限り省略できます。省略する場合はNULLを指定します。

netname

操作するジョブネット名を指定します。省略できません。

parameter

起動/再起動の場合、起動対象のジョブネットの各ジョブに渡すパラメタを200バイト以内の文字列で指定します。

起動/再起動以外の操作の場合、NULLを指定します。

文字列が指定された場合

指定されたパラメタをジョブの引数の最後に付加します。

NULLが指定された場合

再起動の場合、前回起動時に指定されたパラメタをジョブの引数の最後に付加します。

起動の場合は、パラメタを指定しません。

action

ジョブネットの操作方法を指定します。

1:

start(起動)

2:

cancel(強制終了)

3:

rerun(再起動)

4:

pause(停止)

5:

continue(停止解除)

6:

disable(無効)

7:

enable(無効解除)

8:

cope(確認)

9:

reinstate(終了取消)

10:

revoke(持越解除)

rerunopt

actionで3(rerun)を選択した場合のオプションを指定します。actionで3(rerun)を選択しなかった場合は、0を指定します。

0:

通常の再起動を行います。異常終了したジョブから起動します。異常終了したジョブより前のジョブに再起動ポイントが設定されている場合には、そのジョブから起動します。

1:

先頭ジョブからジョブネットを再起動します。

2:

異常終了したジョブに対して「次の再起動のみ無効」を設定してジョブネットを再起動します。

(異常終了したジョブを飛ばして、次のジョブから起動します。)

強制終了したジョブについても、異常終了したジョブと同様に扱います。

復帰値

復帰値	意 味	対 処
0	正常終了しました。	-
4	ジョブネットが操作できない状態です。 (ジョブネットが停止や無効状態の時に起動操作を実施した場合など)	ジョブネットの状態が変わってから再度実行してください。
8	ジョブスケジューラサービス/ジョブスケジューラデーモンが動作していません。	ジョブスケジューラサービス/ジョブスケジューラデーモンを起動して、再度実行してください。

復帰値	意 味	対 処
12	以下のいずれかです。(注) ・ パラメタの指定に誤りがあります。 ・ ユーザの権限に誤りがあります。 ・ 指定したプロジェクト/ジョブネットが存在しない、複写起動で複写されたジョブネット、または、ジョブ実行制御属性ではありません。	以下のいずれかの処置を行ってください。 ・ 正しいパラメタ(プロジェクト名/ジョブネット名/ジョブに渡すパラメタ/操作方法/再起動オプション)を設定する ・ 実行ユーザに、プロジェクトの操作権以上のアクセス権を設定する ・ 操作可能なジョブネットを設定する
16	致命的なエラーが発生しました。	システムエラーが発生した可能性があります。イベントログ/SYSLOGを確認し、システムに異常がないか確認してください。異常があった場合は、保守情報収集ツールで“ジョブスケジューラ”の情報を採取し、富士通技術員に連絡してください。異常がない場合は一時的な問題と思われるため、再度実行してください。
20	可変パラメタ起動されたジョブネットを本APIで再起動することはできません。	起動時に指定された可変パラメタで再起動するには、ジョブネット操作APIで操作してください。

注)
パラメタ指定の誤りの例：

- パラメタnetnameに指定したジョブネットが存在しない。
- パラメタactionに指定可能な値以外が指定されている。
- パラメタprojectがNULLで、かつ、コマンド実行ユーザがアクセスできるプロジェクトが複数ある。

ユーザ権限の誤りの例：

- コマンド実行ユーザにアクセス権(操作権)が不足している。
- API実行ユーザが、swadminグループに属していない。

必要ファイル

ジョブネット操作APIを使用するには、以下のファイルが必要となります。

Windows 32bit用

LIB:

f3csbJobschControl.lib

INCLUDE:

f3csbJobschControl.h

Windows 64bit用

LIB:

f3csbJobschControl_x64.lib

INCLUDE:

f3csbJobschControl_x64.h

Solaris 32bit用、Linux 32bit用共通

LIB:

libjobsch.so

INCLUDE:

jobschapi.h

Solaris 64bit用

LIB:

libjobsch_64.so

INCLUDE:

jobschapi_64.h

Linux 64bit用

LIB:

libjobsch_x64.so

INCLUDE:

jobschapi_x64.h

ジョブネット操作APIはLIBに格納されており、ジョブネット操作APIで使用する定数および構造体はINCLUDEファイルに宣言されています。

これらのファイルは以下のディレクトリに格納されています。

Windows版

LIB：

Systemwalker Operation Managerインストールディレクトリ
MpWalker.JM¥lib

INCLUDE：

Systemwalker Operation Managerインストールディレクトリ
MpWalker.JM¥include

Solaris版、Linux版共通

LIB:

/opt/FJSVJOBSC/usr/lib

INCLUDE:

/opt/FJSVJOBSC/usr/include

注意事項

- 本APIで、可変パラメタ起動されたジョブネットを再起動することはできません。

プログラム例

ジョブネット操作APIのプログラム例を以下に示します。

【Windows版】

```
#include <windows.h>
#include <stdio.h>
#include <stdlib.h>
#include "f3csbJobschControl.h"

int main()
{
    int ret;
```

```
    ret = Mp_JobschControlParam((char*)"user1",       // project name
                          (char*)"net001",            // jobnet name
                          (char*)"param1",            // parameter
                           1,                  // action : start
                           0);                 // rerun opt : none

    if (ret == 0) {
        printf("jobnet(net001) was normally controlled.¥n");
    }
    else{
        printf("ERROR: return code = %d¥n", ret);
        exit(1);
    }
    exit(0);
}
```

【UNIX版】

```
#include <stdio.h>
#include <stdlib.h>
#include "jobschapi.h"

int main()
{
    int ret;

    ret = Mp_JobschControlParam("user1",         /* project name */
                          "net001",        /* jobnet name      */
                          "param1",        /* parameter        */
                           1,              /* action : start   */
                           0);             /* rerun opt : none */

    if (ret == 0) {
        printf("jobnet(net001) was normally controlled.¥n");
    }
    else{
        printf("ERROR: return code = %d¥n", ret);
        exit(1);
    }
    exit(0);
}
```

EE GEE 17.2.4 ジョブネット操作(起動パラメタ指定)API/EE

ジョブネット操作(起動パラメタ指定)API/EE(Mp_JobschControlParamEx)について説明します。

記述形式

【Windows版】

```
#include "f3csbJobschControlEx.h"
int Mp_JobschControlParamEx (char *project,
                        char *netname,
                        char *parameter,
                        int action,
                        int rerunopt,
                        int system_num) ;
```

【UNIX版】

```
#include "jobschapiex.h"
int Mp_JobschControlParamEx (char *project,
                        char *netname,
                        char *parameter,
                        int action,
                        int rerunopt,
                        int system_num) ;
```

機能説明

本APIは、Systemwalker Operation Manager EEおよびGEEで、jobschcontrolコマンドをC言語インタフェースから利用するためのインタフェースです。本APIを利用すると、起動対象のジョブネットの各ジョブにパラメタを渡すことができます。

ジョブ実行制御属性のジョブネットのみ利用できます。

パラメタの説明

project

操作するジョブネットの存在するプロジェクト名を指定します。なお、API実行ユーザの利用できるプロジェクトのプロジェクト名のみ指定できます。API実行ユーザの利用できるプロジェクトのうちジョブネットが登録されているプロジェクトが1つしかない場合に限り省略できます。省略する場合はNULLを指定します。

netname

操作するジョブネット名を指定します。省略することはできません。

parameter

起動/再起動の場合、起動対象のジョブネットの各ジョブに渡すパラメタを200バイト以内の文字列で指定します。

起動/再起動以外の操作の場合、NULLを指定します。

文字列が指定された場合

指定されたパラメタをジョブの引数の最後に付加します。

NULLが指定された場合

再起動の場合、前回起動時に指定されたパラメタをジョブの引数の最後に付加します。

起動の場合は、パラメタを指定しません。

action

ジョブネットの操作方法を指定します。

1:

start(起動)

2:

cancel(強制終了)

3:

rerun(再起動)

4:

pause(停止)

5:

continue(停止解除)

6:

disable(無効)

7:

enable(無効解除)

8:

cope(確認)

9:

reinstate(終了取消)

10:

revoke(持越解除)

rerunopt

actionで3(rerun)を選択した場合のオプションを指定します。actionで3(rerun)を選択しなかった場合は、0を指定します。

0:

通常の再起動を行います。異常終了したジョブから起動します。異常終了したジョブより前のジョブに再起動ポイントが設定されている場合には、そのジョブから起動します。

1:

先頭ジョブからジョブネットを再起動します。

2:

異常終了したジョブに対して「次の再起動のみ無効」を設定してジョブネットを再起動します。

(異常終了したジョブを飛ばして、次のジョブから起動します。)

強制終了したジョブについても、異常終了したジョブと同様に扱います。

system_num

操作対象のサブシステムを、0から9の範囲の番号で指定します。

復帰値

復帰値	意 味	対 処
0	正常終了しました。	-
4	ジョブネットが操作できない状態です。 (ジョブネットが停止や無効状態の時に起動操作を実施した場合など)	ジョブネットの状態が変わってから再度実行してください。
8	ジョブスケジューラサービス/ジョブスケジューラデーモンが動作していません。	ジョブスケジューラサービス/ジョブスケジューラデーモンを起動して、再度実行してください。
12	以下のいずれかです。(注) ・ パラメタの指定に誤りがあります。 ・ ユーザの権限に誤りがあります。 ・ 指定したプロジェクト/ジョブネットが存在しない、複写起動で複写されたジョブネット、または、ジョブ実行制御属性ではありません。	以下のいずれかの処置を行ってください。 ・ 正しいパラメタ(プロジェクト名/ジョブネット名/ジョブに渡すパラメタ/操作方法/再起動オプション/サブシステム番号)を設定する ・ 実行ユーザに、プロジェクトの操作権以上のアクセス権を設定する

復帰値	意 味	対 処
		・ 操作可能なジョブネットを設定する
16	致命的なエラーが発生しました。	システムエラーが発生した可能性があります。イベントログ/SYSLOGを確認し、システムに異常がないか確認してください。異常があった場合は、保守情報収集ツールで“ジョブスケジューラ”の情報を採取し、富士通技術員に連絡してください。異常がない場合は一時的な問題と思われるため、再度実行してください。
20	可変パラメタ起動されたジョブネットを本APIで再起動することはできません。	起動時に指定された可変パラメタで再起動するには、ジョブネット操作APIで操作してください。

注)
パラメタ指定の誤りの例：

- パラメタnetnameに指定したジョブネットが存在しない。
- パラメタactionに指定可能な値以外が指定されている。
- パラメタprojectがNULLで、かつ、コマンド実行ユーザがアクセスできるプロジェクトが複数ある。

ユーザ権限の誤りの例：

- コマンド実行ユーザにアクセス権(操作権)が不足している。
- API実行ユーザが、swadminグループに属していない。

必要ファイル

ジョブネット操作APIを使用するには、以下のファイルが必要となります。

Windows 32bit用

LIB:

f3csbJobschControlEx.lib

INCLUDE:

f3csbJobschControlEx.h

Windows 64bit用

LIB:

f3csbJobschControlEx_x64.lib

INCLUDE:

f3csbJobschControlEx_x64.h

Solaris 32bit用、Linux 32bit用共通

LIB:

libjobschex.so

INCLUDE:

jobschapiex.h

Solaris 64bit用

LIB:

libjobschex_64.so

INCLUDE:

jobschapiex_64.h

Linux 64bit用

LIB:

libjobschex_x64.so

INCLUDE:

jobschapiex_x64.h

ジョブネット操作APIはLIBに格納されており、ジョブネット操作APIで使用する定数および構造体はINCLUDEファイルに宣言されています。

これらのファイルは以下のディレクトリに格納されています。

Windows版

LIB：

Systemwalker Operation Managerインストールディレクトリ
MpWalker.JM¥lib

INCLUDE：

Systemwalker Operation Managerインストールディレクトリ
MpWalker.JM¥include

Solaris版、Linux版共通

LIB:

/opt/FJSVJOBSC/usr/lib

INCLUDE:

/opt/FJSVJOBSC/usr/include

注意事項

- 本APIで、可変パラメタ起動されたジョブネットを再起動することはできません。

プログラム例

ジョブネット操作API/EEのプログラム例を以下に示します。

【Windows版】

```
#include <windows.h>
#include <stdio.h>
#include <stdlib.h>
#include "f3csbJobschControlEx.h"

int main()
{
    int ret;

    ret = Mp_JobschControlParamEx((char*)"user1",         // project name
                                  (char*)"net001",        // jobnet name
                                  (char*)"param1",        // parameter
                                  1,               // action : start
                                  0,               // rerun opt : none
                                  1);              // sub system number
    if (ret == 0) {
        printf("jobnet(net001) was normally controlled.¥n");
    }
    else{
```

```
        printf(”ERROR: return code = %d¥n”, ret);
        exit(1);
    }
    exit(0);
}
```

【UNIX版】

```
#include <stdio.h>
#include <stdlib.h>
#include ”jobschapiex.h”

int main ()
{
    int ret;

    ret = Mp_JobschControlParamEx (”user1”,          /* project name      */
                                   ”net1”,          /* jobnet name       */
                                   ”param1”,        /* parameter         */
                                   1,               /* action : start    */
                                   0,               /* rerun opt : none  */
                                   1);              /* sub system number */

    if (ret == 0) {
        printf (”jobnet(net1) was normally controlled.¥n”);
    }
    else{
        printf (”ERROR: return code = %d¥n”, ret);
        exit(1);
    }
    exit(0);
}
```

17.2.5　グループ操作API

グループ操作API(Mp_JobschCtlGrp)について説明します。

記述形式

【Windows版】

```
#include "f3csbJobschControl.h"
int Mp_JobschCtlGrp (char *project,
                     char *grpname,
                     int action) ;
```

【UNIX版】

```
#include "jobschapi.h"
int Mp_JobschCtlGrp (char *project,
                     char *grpname,
                     int action) ;
```

機能説明

本APIは、jobschctlgrpコマンドをC言語インタフェースから利用するためのインタフェースです。

パラメタの説明

project

操作するグループの存在するプロジェクト名を指定します。なお、API実行ユーザの利用できるプロジェクトのプロジェクト名のみ指定できます。

API実行ユーザの利用できるプロジェクトのうちグループが登録されているプロジェクトが1つしかない場合に限り省略できます。省略する場合はNULLを指定します。

grpname

操作するグループ名を指定します。省略することはできません。

action

グループの操作方法を指定します。

1:

start(起動)

2:

cancel(強制終了)

3:

rerun(再起動)

4:

pause(停止)

5:

continue(停止解除)

6:

disable(無効)

7:

enable(無効解除)

8:

cope(確認)

復帰値

復帰値	意 味	対 処
0	正常終了しました。	-
4	グループが操作できない状態です。	グループの状態が変わってから再度実行してください。
8	ジョブスケジューラサービス/ジョブスケジューラデーモンが動作していません。	ジョブスケジューラサービス/ジョブスケジューラデーモンを起動して、再度実行してください。
12	パラメタの指定に誤りがあります。または、ユーザの権限に誤りがあります。	以下のいずれかの処置を行ってください。 ・正しいパラメタ(プロジェクト名/グループ名/操作方法)を設定する ・実行ユーザに、プロジェクトの操作権以上のアクセス権を設定する
16	致命的なエラーが発生しました。	システムエラーが発生した可能性があります。イベントログ/SYSLOGを確認し、システムに異常がないか確認してください。異常があった場合は、保守情報収集ツールで

復帰値	意 味	対 処
		“ジョブスケジューラ”の情報を採取し、富士通技術員に連絡してください。異常がない場合は一時的な問題と思われるため、再度実行してください。

必要ファイル

グループ操作APIを使用するには、以下のファイルが必要となります。

Windows 32bit用

LIB:

f3csbJobschCtlGrp.lib

INCLUDE:

f3csbJobschControl.h

Windows 64bit用

LIB:

f3csbJobschCtlGrp_x64.lib

INCLUDE:

f3csbJobschControl_x64.h

Solaris 32bit用、Linux 32bit用共通

LIB:

libjobsch.so

INCLUDE:

jobschapi.h

Solaris 64bit用

LIB:

libjobsch_64.so

INCLUDE:

jobschapi_64.h

Linux 64bit用

LIB:

libjobsch_x64.so

INCLUDE:

jobschapi_x64.h

グループ操作APIはLIBに格納されており、グループ操作APIで使用する定数および構造体はINCLUDEファイルに宣言されています。

これらのファイルは以下のディレクトリに格納されています。

Windows版

LIB：

Systemwalker Operation Managerインストールディレクトリ
MpWalker.JM¥lib

INCLUDE：

Systemwalker Operation Managerインストールディレクトリ
MpWalker.JM¥include

Solaris版、Linux版共通

LIB:

/opt/FJSVJOBSC/usr/lib

INCLUDE:

/opt/FJSVJOBSC/usr/include

プログラム例

グループ操作APIのプログラム例を以下に示します。

【Windows版】

```
#include <windows.h>
#include <stdio.h>
#include <stdlib.h>
#include "f3csbJobschControl.h"

int main()
{
    int ret;

    ret = Mp_JobschCtlGrp((char*)"user1",          // project name
                          (char*)"grp001",         // group name
                          1);              // action : start

    if (ret == 0) {
        printf("group(grp001) is controlled normally.¥n");
    }
    else{
        printf("ERROR: return code = %d¥n", ret);
        exit(1);
    }
    exit(0);
}
```

【UNIX版】

```
#include <stdio.h>
#include <stdlib.h>
#include "jobschapi.h"

int main()
{
    int ret;

    ret = Mp_JobschCtlGrp("user1",          /* project name */
                          "grp001",         /* group name */
                          1);               /* action : start */

    if (ret == 0) {
        printf("group(grp001) was normally controlled.¥n");
    }
    else{
        printf("ERROR: return code = %d¥n", ret);
        exit(1);
```

```
    }
    exit(0);
}
```

17.2.6　グループ操作API/EE

グループ操作API/EE(Mp_JobschCtlGrpEx)について説明します。

記述形式

【Windows版】

```
#include "f3csbJobschControlEx.h"
int Mp_JobschCtlGrpEx (char *project,
                       char *grpname,
                       int action,
                       int system_num) ;
```

【UNIX版】

```
#include "jobschapiex.h"
int Mp_JobschCtlGrpEx (char *project,
                       char *grpname,
                       int action,
                       int system_num) ;
```

機能説明

本APIは、Systemwalker Operation Manager EEおよびGEEで、jobschctlgrpコマンドをC言語インタフェースから利用するためのインタフェースです。

パラメタの説明

project

操作するジョブネットの存在するプロジェクト名を指定します。なお、API実行ユーザの利用できるプロジェクトのプロジェクト名のみ指定できます。API実行ユーザの利用できるプロジェクトのうちジョブネットが登録されているプロジェクトが1つしかない場合に限り省略できます。省略する場合はNULLを指定します。

grpname

操作するグループ名を指定します。省略することはできません。

action

グループの操作方法を指定します。

1:

start(起動)

2:

cancel(強制終了)

3:

rerun(再起動)

4:

pause(停止)

5:

continue(停止解除)

6:

disable(無効)

7:

enable(無効解除)

8:

cope(確認)

system_num

操作対象のサブシステムを、0から9の範囲の番号で指定します。

復帰値

復帰値	意 味	対 処
0	正常終了しました。	-
4	グループが操作できない状態です。	グループの状態が変わってから再度実行してください。
8	ジョブスケジューラサービス/ジョブスケジューラデーモンが動作していません。	ジョブスケジューラサービス/ジョブスケジューラデーモンを起動して、再度実行してください。
12	パラメタの指定に誤りがあります。または、ユーザの権限に誤りがあります。	以下のいずれかの処置を行ってください。 ・正しいパラメタ(プロジェクト名/グループ名/操作方法/サブシステム番号)を設定する ・実行ユーザに、プロジェクトの操作権以上のアクセス権を設定する
16	致命的なエラーが発生しました。	システムエラーが発生した可能性があります。イベントログ/SYSLOGを確認し、システムに異常がないか確認してください。異常があった場合は、保守情報収集ツールで“ジョブスケジューラ”の情報を採取し、富士通技術員に連絡してください。異常がない場合は一時的な問題と思われるため、再度実行してください。

必要ファイル

グループ操作APIを使用するには、以下のファイルが必要となります。

Windows 32bit用

LIB:

f3csbJobschControlEx.lib

INCLUDE:

f3csbJobschControlEx.h

Windows 64bit用

LIB:

f3csbJobschControlEx_x64.lib

INCLUDE:

f3csbJobschControlEx_x64.h

Solaris 32bit用、Linux 32bit用共通

LIB:

libjobschex.so

INCLUDE:

jobschapiex.h

Solaris 64bit用

LIB:

libjobschex_64.so

INCLUDE:

jobschapiex_64.h

Linux 64bit用

LIB:

libjobschex_x64.so

INCLUDE:

jobschapiex_x64.h

グループ操作APIはLIBに格納されており、グループ操作APIで使用する定数および構造体はINCLUDEファイルに宣言されています。

これらのファイルは以下のディレクトリに格納されています。

Windows版

LIB：

Systemwalker Operation Managerインストールディレクトリ
MpWalker.JM¥lib

INCLUDE：

Systemwalker Operation Managerインストールディレクトリ
MpWalker.JM¥include

Solaris版、Linux版共通

LIB:

/opt/FJSVJOBSC/usr/lib

INCLUDE:

/opt/FJSVJOBSC/usr/include

プログラム例

グループ操作API/EEのプログラム例を以下に示します。

【Windows版】

```
#include <windows.h>
#include <stdio.h>
#include <stdlib.h>
#include "f3csbJobschControlEx.h"

int main()
{
```

```
    int ret;

    ret = Mp_JobschCtlGrpEx((char*)”user1”,          // project name
                            (char*)”grp001”,         // group name
                            1,               // action : start
                            1);              // sub system number
    if (ret == 0) {
        printf(”group(grp001) is controlled normally.¥n”);
    }
    else{
        printf(”ERROR: return code = %d¥n”, ret);
        exit(1);
    }
    exit(0);
}
```

【UNIX版】

```
#include <stdio.h>
#include <stdlib.h>
#include ”jobschapiex.h”

int main ()
{
    int ret;

    ret = Mp_JobschCtlGrpEx (”user1”,         /* project name      */
                             ”grp1”,          /* group name        */
                             1,               /* action : start    */
                             1);              /* sub system number */

    if (ret == 0) {
        printf (”group(grp1) was normally controlled.¥n”);
    }
    else{
        printf (”ERROR: return code = %d¥n”, ret);
        exit(1);
    }
    exit(0);
}
```

17.2.7 ジョブネット起動API【Windows版】

ジョブネット起動API(JSNetStart)について説明します。

記述形式

```
#include "f3csbNetStart.h"
int JSNetStart (char *project,
            char *netname,
            char *text,
            int kind,
            char *parm,
            int action);
```

機能説明

ジョブネット起動APIの機能を以下に示します。

- ユーザアプリケーションからジョブネットを起動することができます。

- ジョブネット起動時に、任意のメッセージ(イベント)を出力することができます。
- ジョブに任意のパラメタを渡すことができます。
- ジョブネットを起動したときに、同一ジョブネットが実行中だった場合は、ジョブネット起動キューにつながり、ジョブネットの終了を待って起動することができます。

パラメタの説明

project

起動するジョブネットの存在するプロジェクト名を指定します。なお、API実行ユーザの利用できるプロジェクトのプロジェクト名のみ指定できます。

API実行ユーザの利用できるプロジェクトのうちジョブネットが登録されているプロジェクトが1つしかない場合に限り省略できます。省略する場合はNULLを指定します。

netname

起動するジョブネット名を指定します。省略することはできません。

text

イベントログに出力する任意の文字列を、256バイト以内で指定します。NULLを指定した場合、イベントログにメッセージは出力されません。

kind

イベントログに出力するメッセージの種類を指定します。指定する種別を、以下の3つのうちから1つ選んでください。なお、textパラメタがNULLの場合には、このパラメタは無視されます。

LOG_INFO:

情報レベルのメッセージ(メッセージ識別子:10000)

LOG_WARN:

警告レベルのメッセージ(メッセージ識別子:10001)

LOG_ERR:

エラーレベルのメッセージ(メッセージ識別子:10002)

parm

ジョブに渡すパラメタ文字列を512バイト以内で指定します。

起動するジョブネット内のすべてのジョブに指定したパラメタが渡されます。

action

netnameパラメタで指定したジョブネットが、すでに実行中だった場合の対処を指定します。指定内容と意味を以下に示します。

ACT_QUE

ジョブネット起動キューにつなげます。

ACT_NO

起動要求を取り止めます。

復帰値

復帰値	意 味	対 処
0	起動要求は正常に受け付けられ、ジョブネット起動されました。	-

復帰値	意 味	対 処
1	ジョブネットはすでに実行中のため、ジョブネット起動キューにつながれました(ACT_QUE 指定時)。	-
2	ジョブネットはすでに実行中のため、起動要求を取り止めました(ACT_NO指定時)。	-
-1	project パラメタに指定したプロジェクト名が不当です。または、プロジェクト名が存在しません。	正しいプロジェクト名を設定してください。
-2	netname パラメタに指定したジョブネット名が不当です。または、ジョブネット名が存在しません。	正しいジョブネット名を設定してください。
-3	kindパラメタに指定したメッセージの種別が不当です。	正しいメッセージの種別を設定してください。
-4	actionパラメタに指定したジョブネットが実行中だった場合の動作が不当です。	正しい対処を設定してください。
-5	textパラメタに指定した文字列が長すぎます。	textパラメタとして指定する文字列の長さは、256バイト以内にしてください。
-6	parmパラメタに指定した文字列が長すぎます。	paramパラメタとして指定する文字列の長さは、512バイト以内にしてください。
-7	ジョブネットは停止中／無効のため起動できませんでした。ジョブネットの停止中／無効を解除して、再度実行してください。または、ジョブネットがグループに所属しているため起動できません。	以下のいずれかの処置を行ったあと、再度実行してください。 ・ジョブネットの状態を確認する ・ジョブネットをグループから外す
-8	ジョブネット起動処理が混雑しているため、起動できませんでした。しばらく待ってから再度実行してください。	一定時間空けてから再度実行するようにしてください。
-9	ジョブスケジューラサービスが起動されていないか、ジョブネット起動API 機能が無効です。ジョブネット起動API 機能を有効にして、ジョブスケジューラサービスを起動してから、再度実行してください。	以下のいずれかの処置を行ったあと、再度実行してください。 ・ジョブスケジューラサービスを起動する ・ジョブスケジューラのサービスの起動パラメタを再設定後、ジョブスケジューラのサービスを再起動する
-10	ジョブスケジューラサービスは終了処理中です。停止後、ジョブスケジューラサービスを起動して、再度実行してください。	ジョブスケジューラサービス停止後、ジョブスケジューラのサービスを再起動し、再度実行してください。
-20	ジョブネット起動依頼の受付に失敗しました。詳細はイベントログに出力されるので、イベントログを参照して、エラーとなった原因を取り除いた後に、再度実行してください。	イベントログに出力されているエラー内容からエラーの原因を取り除き、再度実行してください。

必要ファイル

ジョブネット起動APIを使用するには、以下のファイルが必要となります。

Windows 32bit用

LIB:

f3csbNetStart.lib

INCLUDE:

f3csbNetStart.h

Windows 64bit用

LIB:

f3csbNetStart_x64.lib

INCLUDE:

f3csbNetStart_x64.h

これらのファイルは、Systemwalker Operation Managerをインストールしたディレクトリの“MpWalker.JM¥lib”および“MpWalker.JM¥include”配下に格納されています。

プログラム例

ジョブネット起動APIのプログラム例を以下に示します。

```
#include <windows.h>
#include <stdio.h>
#include <stdlib.h>
#include "f3csbNetStart.h"

int main()
{
    int ret;

    ret = JSNetStart((char*)"user1",          // project name
                     (char*)"net001",         // jobnet name
                     (char*)"net001 start.",  // messages for event log
                     LOG_INFO,         // kind of event log : information
                     (char*)"-d c:¥¥data -x",  // parameters for jobs
                     ACT_QUE);         // if jobnet is running, that is queued.

    if (ret == 0) {
        printf("jobnet(net001) is started.¥n");
 }
   else if (ret == 1){
        printf("jobnet(net001) is queued, because now running.¥n");
   }
   else if (ret < 0) {
        printf("ERROR: return code = %d¥n", ret);
        exit(1);
   }
   exit(0);
}
```

EE 17.2.8 ジョブネット起動API/EE【Windows版】

ジョブネット起動API/EE(JSNetStartEx)について説明します。

記述形式

```
#include "f3csbNetStartEx.h"
int JSNetStartEx (char *project,
            char *netname,
            char *text,
            int kind,
            char *parm,
            int action,
            int system_num);
```

機能説明

ジョブネット起動API/EEの機能を以下に示します。

- ユーザアプリケーションからジョブネットを起動することができます。
- ジョブネット起動時に、任意のメッセージ(イベント)を出力することができます。
- ジョブに任意のパラメタを渡すことができます。
- ジョブネットを起動したときに、同一ジョブネットが実行中だった場合は、ジョブネット起動キューにつながり、ジョブネットの終了を待って起動することができます。

パラメタの説明

project

操作するジョブネットの存在するプロジェクト名を指定します。なお、API実行ユーザの利用できるプロジェクトのプロジェクト名のみ指定できます。API実行ユーザの利用できるプロジェクトのうちジョブネットが登録されているプロジェクトが1つしかない場合に限り省略できます。省略する場合はNULLを指定します。

netname

起動するジョブネット名を指定します。省略することはできません。

text

イベントログに出力する任意の文字列を、256バイト以内で指定します。NULLを指定した場合、イベントログにメッセージは出力されません。

kind

イベントログに出力するメッセージの種類を指定します。指定する種別を、以下の3つのうちから1つ選んでください。なお、textパラメタがNULLの場合には、このパラメタは無視されます。

LOG_INFO:

情報レベルのメッセージ(メッセージ識別子:10000)

LOG_WARN:

警告レベルのメッセージ(メッセージ識別子:10001)

LOG_ERR:

エラーレベルのメッセージ(メッセージ識別子:10002)

parm

ジョブに渡すパラメタ文字列を512バイト以内で指定します。起動するジョブネット内のすべてのジョブに指定したパラメタが渡されます。

action

netnameパラメタで指定したジョブネットが、すでに実行中だった場合の対処を指定します。指定内容と意味を以下に示します。

ACT_QUE

ジョブネット起動キューにつなげます。

ACT_NO

起動要求を取り止めます。

system_num

操作対象のサブシステムを、0から9の範囲の番号で指定します。

復帰値

復帰値	意味	対処
0	起動要求は正常に受け付けられ、ジョブネット起動されました。	-
1	ジョブネットはすでに実行中のため、ジョブネット起動キューにつながれました(ACT_QUE 指定時)。	-
2	ジョブネットはすでに実行中のため、起動要求を取り止めました(ACT_NO指定時)。	-
-1	project パラメタに指定したプロジェクト名が不当です。または、プロジェクト名が存在しません。	正しいプロジェクト名を設定してください。
-2	netname パラメタに指定したジョブネット名が不当です。または、ジョブネット名が存在しません。	正しいジョブネット名を設定してください。
-3	kindパラメタに指定したメッセージの種別が不当です。	正しいメッセージの種別を設定してください。
-4	actionパラメタに指定したジョブネットが実行中だった場合の動作が不当です。	正しい対処を設定してください。
-5	textパラメタに指定した文字列が長すぎます。	textパラメタとして指定する文字列の長さは、256バイト以内にしてください。
-6	parmパラメタに指定した文字列が長すぎます。	paramパラメタとして指定する文字列の長さは、512バイト以内にしてください。
-7	ジョブネットは停止中／無効のため起動できませんでした。ジョブネットの停止中／無効を解除して、再度実行してください。または、ジョブネットがグループに所属しているため起動できません。	以下のいずれかの処置を行ったあと、再度実行してください。 ・ジョブネットの状態を確認する ・ジョブネットをグループから外す
-8	ジョブネット起動処理が混雑しているため、起動できませんでした。しばらく待ってから再度実行してください。	一定時間空けてから再度実行するようにしてください。
-9	ジョブスケジューラサービスが起動されていないか、ジョブネット起動API機能が無効です。ジョブネット起動API機能を有効にして、ジョブスケジューラサービスを起動してから、再度実行してください。	以下のいずれかの処置を行ったあと、再度実行してください。 ・ジョブスケジューラサービスを起動する ・ジョブスケジューラのサービスの起動パラメタを再設定後、ジョブスケジューラのサービスを再起動する
-10	ジョブスケジューラサービスは終了処理中です。停止後、ジョブスケジューラサービスを起動して、再度実行してください。	ジョブスケジューラサービス停止後、ジョブスケジューラのサービスを再起動し、再度実行してください。
-20	ジョブネット起動依頼の受付に失敗しました。詳細はイベントログに出力されるので、イベントログを参照して、エラーとなった原因を取り除いた後に、再度実行してください。	イベントログに出力されているエラー内容からエラーの原因を取り除き、再度実行してください。

必要ファイル

ジョブネット起動APIを使用するには、以下のファイルが必要となります。

Windows 32bit用

LIB:

f3csbNetStartEx.lib

INCLUDE:

f3csbNetStartEx.h

Windows 64bit用

LIB:

f3csbNetStartEx_x64.lib

INCLUDE:

f3csbNetStartEx_x64.h

これらのファイルは、Systemwalker Operation Managerをインストールしたディレクトリの“MpWalker.JM¥lib”および“MpWalker.JM¥include”配下に格納されています。

プログラム例

ジョブネット起動API/EEのプログラム例を以下に示します。

```
#include <windows.h>
#include <stdio.h>
#include <stdlib.h>
#include "f3csbNetStartEx.h"

int main()
{
    int ret;

    ret = JSNetStartEx((char*)"user1",          // project name
                       (char*)"net001",         // jobnet name
                       (char*)"net001 start.",  // messages for event log
                       LOG_INFO,      // kind of event log : information
                       (char*)"-d c:¥¥data -x",  // parameters for jobs
                       ACT_QUE,  //if jobnet is running, that is queued.
                       1);               // sub system number
    if (ret == 0) {
        printf("jobnet(net001) is started.¥n");
  }
    else if (ret == 1){
        printf("jobnet(net001) is queued, because now running.¥n");
    }
    else if (ret < 0) {
        printf("ERROR: return code = %d¥n", ret);
        exit(1);
    }
    exit(0);
}
```

17.2.9　ジョブネット起動時刻変更API

ジョブネット起動時刻変更API(Mp_JobschCtlStartTime)について説明します。

記述形式

【Windows版】

```
#include "f3csbJobschControl.h"
int Mp_JobschCtlStartTime (
```

```
        int mode,
        struct jobsch_start_time *get_time,
        struct jobsch_start_time *before_time,
        struct jobsch_start_time *set_time,
        char *project_name,
        char *jobnet_name) ;
```

【UNIX版】

```
#include "jobschapi.h"
int Mp_JobschCtlStartTime (
        int mode,
        struct jobsch_start_time *get_time,
        struct jobsch_start_time *before_time,
        struct jobsch_start_time *set_time,
        char *project_name,
        char *jobnet_name) ;
```

機能説明

本APIは、ジョブネットの起動時刻を取得または変更する機能を、C言語インタフェースから利用するためのインタフェースです。

起動時刻の取得では、指定されたジョブネットの起動時刻がget_timeパラメタで指定した領域に格納されます。

起動時刻の変更では、指定されたジョブネットは、起動時刻がset_timeパラメタで指定した値に変更され、変更後の起動時刻と起動日情報により起動予定日時が再スケジュールされます。

パラメタの説明

mode

本APIの動作モードを指定します。省略できません。

0:

起動時刻を取得します。before_timeパラメタおよびset_timeパラメタは無視されます。本モードを使用するためには、ジョブネットが属するプロジェクトに対して、参照権以上を持つユーザでAPIを実行する必要があります。

1:

起動時刻を変更します。get_timeパラメタは無視されます。本モードを使用するためには、ジョブネットが属するプロジェクトに対して、更新権または登録権を持つユーザでAPIを実行する必要があります。

get_time:

起動時刻を取得するjobsch_start_time構造体の領域を指定します。省略する場合はNULLを指定します。modeパラメタに0を指定した場合は、省略できません。modeパラメタに0を指定した場合で、本APIが正常終了した場合には、取得結果が構造体に格納されます。格納される値は、jobsch_start_time構造体のtimeパラメタの説明を参照してください。

before_time:

変更前の起動時刻を指定します。指定する値は、jobsch_start_time構造体のtimeパラメタの説明を参照してください。省略する場合はNULLを指定します。modeパラメタに1を指定した場合は、省略できません。指定された起動時刻の種類および起動時刻と、ジョブネットに指定されている起動時刻の種類および起動時刻を比較し、一致した場合のみ起動時刻を変更します。一致しなかった場合には、APIがエラー復帰します。

set_time:

設定する起動時刻を指定します。指定する値は、jobsch_start_time構造体のtimeパラメタの説明を参照してください。省略する場合はNULLを指定します。modeパラメタに1を指定した場合は、省略できません。

project_name:

操作するジョブネットの属するプロジェクト名を指定します。API実行ユーザの利用できるプロジェクトのプロジェクト名のみ指定できます。API実行ユーザの利用できるプロジェクトのうちジョブネットが登録されているプロジェクトが1つしかない場合に限り省略できます。省略する場合はNULLを指定します。

jobnet_name:

操作するジョブネット名を指定します。省略できません。

復帰値

復帰値	意味	対処
0	正常終了しました。	-
4	指定のジョブネットの変更前の起動時刻が一致しません。	変更しようとするジョブネットと同じ起動時刻の種類および起動時刻を、before_timeに指定してください。
8	取得しようとしたジョブネットの起動条件が時刻起動ではありません。または、指定のジョブネットの起動条件が、変更しようとする起動時刻の種類と異なります。	取得対象のジョブネットには、起動時刻が設定されたジョブネットを指定してください。起動時刻を変更する場合は、set_timeに対象ジョブネットの起動時刻と同じ種類を指定してください。
12	指定のジョブネットはグループに属しています。	グループに属さないジョブネットを指定してください。
16	指定のジョブネットは実行中、警告、または終了遅延です。	ジョブネットの状態が変わってから再度実行してください。
24	Systemwalkerのコード系と異なるコード系で動作しています。	APIを呼び出すプログラムの実行環境を、Systemwalkerのコード系と同じコード系にしてください。
28	ジョブスケジューラサービス/デーモンが動作していません。	ジョブスケジューラサービス/デーモンを起動して、再度実行してください。
32	パラメタの指定、または、ユーザの権限に誤りがあります。または、指定したプロジェクト、ジョブネットが存在しません。	必要に応じて以下の処置を行ってください。 ・正しいパラメタ(動作モード/起動時刻/プロジェクト名/ジョブネット名)を設定する ・実行ユーザに、プロジェクトのアクセス権を設定する
36	致命的なエラーが発生しました。	システムエラーが発生した可能性があります。イベントログ/SYSLOGを確認し、システムに異常がないか確認してください。異常があった場合は、保守情報収集ツールで“ジョブスケジューラ”の情報を採取し、富士通技術員に連絡してください。異常がない場合は一時的な問題と思われるため、再度実行してください。
40	指定のジョブネットは操作中または変更中です。	ジョブネットの状態が変わってから再度実行してください。

jobsch_start_time構造体の形式

jobsch_start_time構造体の形式を以下に示します。

```
struct jobsch_start_time {
     int type;
     char rsv1[4];
     char time[128];
     char rsv2[24];
} ;
```

type:

起動時刻の種類を表す数値です。

値	種 別
1	起動時刻が1つのみ指定されているジョブネット
2	起動時刻が複数指定されているジョブネット
3	間隔起動のジョブネット

rsv1:

予約域です。使用しません。

time:

起動時刻を表す文字列です。起動時刻の種類によって、形式が異なります。

1：起動時刻が1つのみ指定されているジョブネットの場合

HHMM(時分)形式の4バイト(ヌル文字含め5バイト)の文字列です。

例）「0900」

2：起動時刻が複数指定されているジョブネットの場合

HHMM(時分)形式の4バイトの文字列がコンマ(,)で区切られた最大29バイト(4バイト×6個 コンマ×5個、ヌル文字含め30バイト)の文字列です。

例）「0900,1100,1300,1500,1700」

3：間隔起動のジョブネットの場合

HHMM,HHMM,n(開始時刻,終了時刻,起動間隔)形式の最大13バイト(ヌル文字含め14バイト)の文字列です。各項目はコンマ(,)で区切られます。開始時刻と終了時刻は、HHMM(時分)形式の4バイトの文字列です。起動間隔は、1から240(分)の1バイトから3バイトの文字列です。

例）「0900,1700,60」

rsv2:

予約域です。使用しません。

必要ファイル

ジョブネット起動時刻変更APIを使用するには、以下のファイルが必要となります。

Windows 32bit用

LIB:

Systemwalker Operation Managerインストールディレクトリ¥MpWalker.JM¥lib¥f3csbJobschControl.lib

INCLUDE:

Systemwalker Operation Managerインストールディレクトリ¥MpWalker.JM¥include¥f3csbJobschControl.h

Windows 64bit用

LIB:

Systemwalker Operation Manager インストールディレクトリ¥MpWalker.JM¥lib¥f3csbJobschControl_x64.lib

INCLUDE:

Systemwalker Operation Manager インストールディレクトリ¥MpWalker.JM¥include¥f3csbJobschControl_x64.h

Solaris 32bit用、Linux 32bit用共通

LIB:

/opt/FJSVJOBSC/usr/lib/libjobsch.so

INCLUDE:

/opt/FJSVJOBSC/usr/include/jobschapi.h

Solaris 64bit用

LIB:

/opt/FJSVJOBSC/usr/lib/libjobsch_64.so

INCLUDE:

/opt/FJSVJOBSC/usr/include/jobschapi_64.h

Linux 64bit用

LIB:

/opt/FJSVJOBSC/usr/lib/libjobsch_x64.so

INCLUDE:

/opt/FJSVJOBSC/usr/include/jobschapi_x64.h

実行に必要な権限

システム管理者、または対象のプロジェクトにアクセス権を持つユーザの権限が必要です。

注意事項

- jobsch_start_time構造体内にある起動時刻に示した起動条件のジョブネットに対してのみ有効です。
- 起動時刻の種類を変更することはできません。
- グループに属しているジョブネットの起動時刻を変更することはできません。
- 本APIは、タイムゾーンを正しく設定したプログラムで使用してください。
- Systemwalker Operation Managerが動作しているロケールと異なるロケールを指定したプログラムから呼び出しを行った場合、本APIはエラー復帰します。
- 本APIのパラメタに文字コードを含む文字列を指定する場合、Systemwalker Operation Managerが動作しているLANGに合わせた文字コードで指定してください。

プログラム例

ジョブネット起動時刻変更APIのプログラム例を以下に示します。

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <locale.h>
#include <string.h>
```

```
#include ”jobschapi.h”

int main ()
{
    int ret;
    struct jobsch_start_time b_time, a_time;

    tzset ();
    setlocale (LC_ALL, ””);

    memset(&b_time, 0x00, sizeof(b_time));

    ret = Mp_JobschCtlStartTime (
                        0,          /* mode : GET       */
                        &b_time,    /* get start time   */
                        NULL,       /* NULL             */
                        NULL,       /* NULL             */
                        ”test”,     /* project name     */
                        ”net1”);    /* jobnet name      */
    if (ret == 0) {
        printf (”Start type is %d.¥n”, b_time.type);
        printf (”Start time is %s.¥n”, b_time.time);
    }
    else{
        printf (”ERROR: get return code = %d¥n”, ret);
        exit(1);
    }

    memset(&a_time, 0x00, sizeof(a_time));
    a_time.type = b_time.type;
    strcpy(a_time.time, b_time.time);
    strcat(a_time.time, ”,1900”);

    ret = Mp_JobschCtlStartTime (
                        1,          /* mode : CHANGE    */
                        NULL,       /* NULL             */
                        &b_time,    /* before time      */
                        &a_time,    /* set time         */
                        ”test”,     /* project name     */
                        ”net1”);    /* jobnet name      */
    if (ret == 0) {
        printf (”Change type is %d.¥n”, a_time.type);
        printf (”Change time is %s.¥n”, a_time.time);
    }
    else{
        printf (”ERROR: set return code = %d¥n”, ret);
        exit(1);
    }

    exit(0);
}
```

17.2.10　ジョブネット起動時刻変更API/EE

ジョブネット起動時刻変更API/EE(Mp_JobschCtlStartTimeEx)について説明します。

記述形式

【Windows版】

```
#include "f3csbJobschControlEx.h"
int Mp_JobschCtlStartTimeEx (
```

```
        int mode,
        struct jobsch_start_time *get_time,
        struct jobsch_start_time *before_time,
        struct jobsch_start_time *set_time,
        char *project_name,
        char *jobnet_name,
        int system_num) ;
```

【UNIX版】

```
#include "jobschapiex.h"
int Mp_JobschCtlStartTimeEx (
        int mode,
        struct jobsch_start_time *get_time,
        struct jobsch_start_time *before_time,
        struct jobsch_start_time *set_time,
        char *project_name,
        char *jobnet_name,
        int system_num) ;
```

機能説明

本APIは、Systemwalker Operation Manager EEおよびGEEで、ジョブネットの起動時刻を取得または変更する機能を、C言語インタフェースから利用するためのインタフェースです。

起動時刻の取得では、指定されたジョブネットの起動時刻がget_timeパラメタで指定した領域に格納されます。

起動時刻の変更では、指定されたジョブネットは、起動時刻がset_timeパラメタで指定した値に変更され、変更後の起動時刻と起動日情報により起動予定日時が再スケジュールされます。

パラメタの説明

mode

本APIの動作モードを指定します。省略できません。

0:

起動時刻を取得します。before_timeパラメタおよびset_timeパラメタは無視されます。本モードを使用するためには、ジョブネットが属するプロジェクトに対して、参照権以上を持つユーザでAPIを実行する必要があります。

1:

起動時刻を変更します。get_timeパラメタは無視されます。本モードを使用するためには、ジョブネットが属するプロジェクトに対して、更新権または登録権を持つユーザでAPIを実行する必要があります。

get_time:

起動時刻を取得するjobsch_start_time構造体の領域を指定します。省略する場合はNULLを指定します。modeパラメタに0を指定した場合は、省略できません。modeパラメタに0を指定した場合で、本APIが正常終了した場合には、取得結果が構造体に格納されます。格納される値は、jobsch_start_time構造体のtimeパラメタの説明を参照してください。

before_time:

変更前の起動時刻を指定します。指定する値は、jobsch_start_time構造体のtimeパラメタの説明を参照してください。省略する場合はNULLを指定します。modeパラメタに1を指定した場合は、省略できません。指定された起動時刻の種類および起動時刻と、ジョブネットに指定されている起動時刻の種類および起動時刻を比較し、一致した場合のみ起動時刻を変更します。一致しなかった場合には、APIがエラー復帰します。

set_time:

設定する起動時刻を指定します。指定する値は、jobsch_start_time構造体のtimeパラメタの説明を参照してください。省略する場合はNULLを指定します。modeパラメタに1を指定した場合は、省略できません。

project_name:

操作するジョブネットの属するプロジェクト名を指定します。API実行ユーザの利用できるプロジェクトのプロジェクト名のみ指定できます。API実行ユーザの利用できるプロジェクトのうちジョブネットが登録されているプロジェクトが1つしかない場合に限り省略できます。省略する場合はNULLを指定します。

jobnet_name:

操作するジョブネット名を指定します。省略できません。

system_num:

操作対象のサブシステムを、0から9の範囲の番号で指定します。省略できません。

復帰値

復帰値	意味	対処
0	正常終了しました。	-
4	指定のジョブネットの変更前の起動時刻が一致しません。	変更しようとするジョブネットと同じ起動時刻の種類および起動時刻を、before_timeに指定してください。
8	取得しようとしたジョブネットの起動条件が時刻起動ではありません。または、指定のジョブネットの起動条件が、変更しようとする起動時刻の種類と異なります。	取得対象のジョブネットには、起動時刻が設定されたジョブネットを指定してください。起動時刻を変更する場合は、set_timeに対象ジョブネットの起動時刻と同じ種類を指定してください。
12	指定のジョブネットはグループに属しています。	グループに属さないジョブネットを指定してください。
16	指定のジョブネットは実行中、警告、または終了遅延です。	ジョブネットの状態が変わってから再度実行してください。
24	Systemwalkerのコード系と異なるコード系で動作しています。	APIを呼び出すプログラムの実行環境を、Systemwalkerのコード系と同じコード系にしてください。
28	ジョブスケジューラサービス/デーモンが動作していません。	ジョブスケジューラサービス/デーモンを起動して、再度実行してください。
32	パラメタの指定、または、ユーザの権限に誤りがあります。または、指定したプロジェクト、ジョブネットが存在しません。	必要に応じて以下の処置を行ってください。 ・正しいパラメタ(動作モード/起動時刻/プロジェクト名/ジョブネット名)を設定する ・実行ユーザに、プロジェクトのアクセス権を設定する
36	致命的なエラーが発生しました。	システムエラーが発生した可能性があります。イベントログ/SYSLOG を確認し、システムに異常がないか確認してください。異常があった場合は、保守情報収集ツールで“ジョブスケジューラ”の情報を採取し、富士通技術員に連絡してください。異常がない

復帰値	意味	対処
		場合は一時的な問題と思われるため、再度実行してください。
40	指定のジョブネットは操作中または変更中です。	ジョブネットの状態が変わってから再度実行してください。

jobsch_start_time構造体の形式

jobsch_start_time構造体の形式を以下に示します。

```
struct jobsch_start_time {
     int type;
     char rsv1[4];
     char time[128];
     char rsv2[24];
} ;
```

type:

起動時刻の種類を表す数値です。

値	種別
1	起動時刻が1つのみ指定されているジョブネット
2	起動時刻が複数指定されているジョブネット
3	間隔起動のジョブネット

rsv1:

予約域です。使用しません。

time:

起動時刻を表す文字列です。起動時刻の種類によって、形式が異なります。

1：起動時刻が1つのみ指定されているジョブネットの場合

HHMM(時分)形式の4バイト(ヌル文字含め5バイト)の文字列です。

例）「0900」

2：起動時刻が複数指定されているジョブネットの場合

HHMM(時分)形式の4バイトの文字列がコンマ (,)で区切られた最大29バイト(4バイト×6個 コンマ×5個、ヌル文字含め30バイト)の文字列です。

例）「0900,1100,1300,1500,1700」

3：間隔起動のジョブネットの場合

HHMM,HHMM,n(開始時刻,終了時刻,起動間隔)形式の最大13バイト(ヌル文字含め14バイト)の文字列です。各項目はコンマ (,)で区切られます。開始時刻と終了時刻は、HHMM(時分)形式の4バイトの文字列です。起動間隔は、1から240(分)の1バイトから3バイトの文字列です。

例）「0900,1700,60」

rsv2:

予約域です。使用しません。

必要ファイル

ジョブネット起動時刻変更APIを使用するには、以下のファイルが必要となります。

Windows 32bit用

LIB:

Systemwalker Operation Manager インストールディレクトリ¥MpWalker.JM¥lib¥f3csbJobschControlEx.lib

INCLUDE:

Systemwalker Operation Manager インストールディレクトリ¥MpWalker.JM¥include¥f3csbJobschControlEx.h

Windows 64bit用

LIB:

Systemwalker Operation Manager インストールディレクトリ¥MpWalker.JM¥lib¥f3csbJobschControlEx_x64.lib

INCLUDE:

Systemwalker Operation Manager インストールディレクトリ¥MpWalker.JM¥include¥f3csbJobschControlEx_x64.h

Solaris 32bit用、Linux 32bit用共通

LIB:

/opt/FJSVJOBSC/usr/lib/libjobschex.so

INCLUDE:

/opt/FJSVJOBSC/usr/include/jobschapiex.h

Solaris 64bit用

LIB:

/opt/FJSVJOBSC/usr/lib/libjobschex_64.so

INCLUDE:

/opt/FJSVJOBSC/usr/include/jobschapiex_64.h

Linux 64bit用

LIB:

/opt/FJSVJOBSC/usr/lib/libjobschex_x64.so

INCLUDE:

/opt/FJSVJOBSC/usr/include/jobschapiex_x64.h

実行に必要な権限

システム管理者、または対象のプロジェクトにアクセス権を持つユーザの権限が必要です。

注意事項

- jobsch_start_time構造体内にある起動時刻に示した起動条件のジョブネットに対してのみ有効です。
- 起動時刻の種類を変更することはできません。
- グループに属しているジョブネットの起動時刻を変更することはできません。
- 本APIは、タイムゾーンを正しく設定したプログラムで使用してください。
- Systemwalker Operation Managerが動作しているロケールと異なるロケールを指定したプログラムから呼び出しを行った場合、本APIはエラー復帰します。
- 本APIのパラメタに文字コードを含む文字列を指定する場合、Systemwalker Operation Managerが動作しているLANGに合わせた文字コードで指定してください。

プログラム例

ジョブネット起動時刻変更APIのプログラム例を以下に示します。

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <locale.h>
#include <string.h>
#include "jobschapiex.h"

int main ()
{
    int ret;
    struct jobsch_start_time b_time, a_time;

    tzset ();
    setlocale (LC_ALL, "");

    memset(&b_time, 0x00, sizeof(b_time));

    ret = Mp_JobschCtlStartTimeEx (
                        0,          /* mode : GET       */
                        &b_time,    /* get start time   */
                        NULL,       /* NULL             */
                        NULL,       /* NULL             */
                        "test",     /* project name     */
                        "net1",    /* jobnet name      */
                        1);         /* sub system number*/
    if (ret == 0) {
        printf ("Start type is %d.¥n", b_time.type);
        printf ("Start time is %s.¥n", b_time.time);
    }
    else{
        printf ("ERROR: get return code = %d¥n", ret);
        exit(1);
    }

    memset(&a_time, 0x00, sizeof(a_time));
    a_time.type = b_time.type;
    strcpy(a_time.time, b_time.time);
    strcat(a_time.time, ",1900");

    ret = Mp_JobschCtlStartTimeEx (
                        1,          /* mode : CHANGE    */
                        NULL,       /* NULL             */
                        &b_time,    /* before time      */
                        &a_time,    /* set time         */
                        "test",     /* project name     */
                        "net1",    /* jobnet name      */
                        1);         /* sub system number*/
    if (ret == 0) {
        printf ("Change type is %d.¥n", a_time.type);
        printf ("Change time is %s.¥n", a_time.time);
    }
    else{
        printf ("ERROR: set return code = %d¥n", ret);
        exit(1);
    }

    exit(0);
}
```

17.3　ジョブ実行制御API

本節では、ジョブ実行制御が提供するAPIについて説明します。

ジョブ実行制御API共通の動作環境、注意事項、および必要ファイルについての説明を以下に示します。

動作環境

当APIは、Systemwalker Operation Managerサーバ導入マシン上でのみ動作します。Systemwalker Operation Managerクライアントのみが導入されているマシン上では動作しません。

注意事項

- コンパイル環境として、コンパイラは、Visual C++(バージョン2022以降)を、ランタイムライブラリは“マルチスレッド”を使用してください。【Windows版】
- コンパイルしたVisual C++バージョン以降のランタイムをAPI実行環境にインストールしてください。【Windows版】
- LIBをリンクしているアプリケーションは、当バージョン/レベルで提供されたLIBファイルおよびINCLUDEファイルを使用して、再度リンクしてください。
- ジョブ実行制御APIを、同一プロセス内の複数のスレッドから、同時に呼び出さないでください。

必要ファイル

上記ジョブ実行制御APIを使用するには、以下のファイルが必要となります。

Windows 32bit用

LIB:

f3cuapi.lib

INCLUDE:

f3cuapi.h

Windows 64bit用

LIB:

f3cuapi_x64.lib

INCLUDE:

f3cuapi_x64.h

Solaris 32bit用、Linux 32bit用共通

LIB:

libmjes.so

INCLUDE:

mjesapi.h

Solaris 64bit用

LIB:

libmjes_64.so

INCLUDE:

mjesapi_64.h

Linux 64bit用

LIB:

libmjes_x64.so

INCLUDE:

mjesapi_x64.h

ジョブ実行制御の各APIはLIBファイルに格納し、各APIで使用する定数および構造体はINCLUDEファイルに宣言します。

これらのファイルは、以下のディレクトリ配下に格納されています。

Windows版

LIB：

Systemwalker Operation Managerインストールディレクトリ¥MpWalker.JM¥lib

INCLUDE：

Systemwalker Operation Managerインストールディレクトリ¥MpWalker.JM¥include

Solaris版、Linux版共通

LIB：

/opt/FJSVMJS/usr/lib

INCLUDE：

/opt/FJSVMJS/usr/include

17.3.1 導入/稼働情報獲得API【Windows版】

導入/稼働情報獲得API(Mp_StatusMJES)について説明します。

機能説明

ジョブ実行制御のバージョンレベル、および稼働状態を獲得することができます。

記述形式

```
#include "f3cuapi.h"
long Mp_StatusMJES()
```

復帰値

復帰値	意 味	対 処
0	ジョブ実行制御は稼働していません。	-
1以上	ジョブ実行制御は稼働中です。 具体的な返り値は、製品バージョンレベルによって異なります。 －V17.2.0の場合 170*65536+30=(11141150)10=(AA001E)16 －V17.1.0/V17.1.1の場合 170*65536+20=(11141140)10=(AA0014)16 －V17.0.0/V17.0.1の場合 170*65536+10=(11141130)10=(AA000A)16 －V16.0.0の場合 160*65536+10=(10485770)10=(A0000A)16 －V13.8.0の場合 140*65536+40=(9175080)10=(8C0028)16 －V13.7.0の場合 140*65536+30=(9175070)10=(8C001E)16 －V13.6.0/V13.6.1の場合 140*65536+20=(9175060)10=(8C0014)16 －V13.4.0の場合	-

復帰値	意 味	対 処
	140*65536+10=(9175050)10=(8C000A)16 －V13.3.0/V13.3.1の場合 130*65536+40=(8519720)10=(820028)16 なお、V13.2.0以前のSystemwalker Operation Managerの場合は、上位2バイトにバージョン、下位2バイトにレベルが返ります。 Systemwalker Operation Manager V13.2.0の場合は、バージョンは130(1個目のピリオドの前の整数×10)、レベルは30(2個目のピリオドの前の整数×10+10+2個目のピリオドの後ろの整数)となります。 －V13.2.0の場合 130*65536+30=(8519710)10=(82001E)16	
-1	システムエラーが発生しました。	保守情報収集ツールで“ジョブ実行制御”の情報を採取し、富士通技術員に連絡してください。

なお、システムエラーが発生した場合は、GetLastError()で詳細エラーコードを調べることができます。

プログラム例

導入/稼働情報獲得APIのプログラム例を以下に示します。

```
#include <windows.h>
#include <stdio.h>
#include "f3cuapi.h"

int main (int argc, char **argv)
{
        long rtn;

        rtn = Mp_StatusMJES ();
        switch (rtn) {
        case 0:
                printf ("Systemwalker MpMjes service is not running.¥n");
                return (1);
        case MJSESYSER:
                printf ("GetLastError() is %d¥n", GetLastError());
                printf ("System error occurred.¥n");
                return (1);
        }
        printf ("Version : %d¥n", rtn / 65536);
        printf ("Level   : %d¥n", rtn % 65536);
        return (0);
}
```

EE

17.3.2　導入/稼働情報獲得API/EE【Windows版】

導入/稼働情報獲得API/EE(Mp_StatusMJES_Ex)について説明します。

機能説明

Systemwalker Operation Manager EEで、ジョブ実行制御のバージョンレベル、および稼働状態を獲得することができます。

記述形式

```
#include "f3cuapi.h"
long Mp_StatusMJES_Ex(int system_num)
```

パラメタの説明

system_num

操作対象のサブシステムを、0から9の範囲の番号で指定します。

復帰値

復帰値	意 味	対 処
0	ジョブ実行制御は稼働していません。	-
1以上	ジョブ実行制御は稼働中です。 具体的な返り値は、製品バージョンレベルによって異なります。 －V17.2.0の場合 170*65536+30=(11141150)10=(AA001E)16 －V17.1.0/V17.1.1の場合 170*65536+20=(11141140)10=(AA0014)16 －V17.0.0/V17.0.1の場合 170*65536+10=(11141130)10=(AA000A)16 －V16.0.0の場合 160*65536+10=(10485770)10=(A0000A)16 －V13.8.0の場合 140*65536+40=(9175080)10=(8C0028)16 －V13.7.0の場合 140*65536+30=(9175070)10=(8C001E)16 －V13.6.0/V13.6.1の場合 140*65536+20=(9175060)10=(8C0014)16 －V13.4.0の場合 140*65536+10=(9175050)10=(8C000A)16 －V13.3.0/V13.3.1の場合 130*65536+40=(8519720)10=(820028)16 なお、V13.2.0以前のSystemwalker Operation Managerの場合は、上位2バイトにバージョン、下位2バイトにレベルが返ります。 Systemwalker Operation Manager V13.2.0の場合は、バージョンは130(1個目のピリオドの前の整数×10)、レベルは30(2個目のピリオドの前の整数×10+10+2個目のピリオドの後ろの整数)となります。 －V13.2.0の場合 130*65536+30=(8519710)10=(82001E)16	-
-1	システムエラーが発生しました。	保守情報収集ツールで“ジョブ実行制御”の情報を採取し、富士通技術員に連絡してください。

なお、システムエラーが発生した場合は、GetLastError()で詳細エラーコードを調べることができます。

プログラム例

導入/稼働情報獲得API/EEのプログラム例を以下に示します。

```
#include <windows.h>
#include <stdio.h>
#include "f3cuapi.h"

int main (int argc, char **argv)
{
        long rtn;
        int system_num;

        system_num = 0;
        rtn = Mp_StatusMJES_Ex (system_num);
        switch (rtn) {
        case 0:
               printf ("Systemwalker MpMjes service is not running.¥n");
               return (1);
        case MJSESYSER:
               printf ("GetLastError() is %d¥n", GetLastError());
               printf ("System error occurred.¥n");
               return (1);
        }
        printf ("Version : %d¥n", rtn / 65536);
        printf ("Level   : %d¥n", rtn % 65536);
        return (0);
}
```

17.3.3 ジョブ投入API

ジョブ投入API(Mp_SubmitJob)について説明します。

機能説明

ジョブをSystemwalker Operation Managerを導入しているサーバに投入します。投入に成功するとジョブを識別する情報として“ジョブ投入ホスト名”、“ジョブ名”および“ジョブ番号”が返却されます。

記述形式

【Windows版】

```
#include "f3cuapi.h"
long Mp_SubmitJob (char *rhost,
                char *submitinf,
                char *env[ ],
                char *jobhost,
                char *jobname,
                long *jobno)
```

【UNIX版】

```
#include "mjesapi.h "
int Mp_SubmitJob (char *rhost,
                char *submitinf,
                char *env[ ],
                char *jobhost,
                char *jobname,
                int *jobno)
```

パラメタの説明

rhost

NULLを指定してください。

submitinf

ジョブ投入情報としてqsubコマンドの引数を指定します。引数の記述形式は、qsubコマンドにおけるオプションおよびオペランドの指定方法と同じです。

qsubコマンドの引数の記述形式および注意事項については、“7.18 qsub ジョブの投入コマンド”を参照してください。

env[ ]

ジョブに引き継ぐ環境変数を指定します。引き継ぐ必要がない場合は、NULLを指定してください。

また、submitinfにおいて、“-x”を指定した場合のみ、当環境変数がジョブに引き継がれます。

jobhost

ジョブが正常に投入された場合、ジョブを識別する情報として、ジョブ投入ホスト名が返却されます。

jobname

ジョブが正常に投入された場合は、ジョブを識別する情報として、ジョブ実行制御が決定したジョブ名が返却されます。ジョブ投入時にジョブ名を指定した場合は、そのジョブ名でジョブが投入されます。ただし、省略時およびジョブファイル内でジョブ名を宣言している場合は、本パラメタで指定した領域に返却されるジョブ名により、投入したジョブのジョブ名を知ることができます。

jobno

ジョブが正常に投入された場合、ジョブを識別する情報として、ジョブ実行制御が割り当てたジョブ番号が返却されます。当ジョブ番号によって、ジョブに対する操作が可能です。

復帰値

復帰値	意 味	対 処
0	正常にジョブは投入されました。	-
-1	システムエラーが発生しました。	保守情報収集ツールで“ジョブ実行制御”の情報を採取し、富士通技術員に連絡してください。
-1000以下	ジョブ実行制御において、エラーが検出されたため、投入処理を中断しました。 なお、エラーメッセージ獲得APIを使用して、エラーメッセージを獲得することができます。	エラーメッセージ獲得APIを使用して、エラーメッセージを獲得し、そのエラーメッセージに対応した処置を実施してください。

注意事項

引数が無効アドレスを指している場合、当関数は異常終了し、その動作は定義されません。

プログラム例

ジョブ投入APIのプログラム例を以下に示します。

【Windows版】

```
#include <windows.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include ”f3cuapi.h”
```

```
int main (int argc, char **argv)
{
        char submitinf[1024+1];
        char jobhost [64+1];
        char jobname [64+1];
        long jobno = 0;
        long rtn = 0;
        char errmsg [256+1];

        /*
         * initialization working area
         */
        memset(submitinf, 0x00, sizeof(submitinf));
        memset(jobhost, 0x00, sizeof(jobhost));
        memset(jobname, 0x00, sizeof(jobname));
        memset(errmsg, 0x00, sizeof(errmsg));

        if (argc != 4) {
                printf ("Usage: %s <queue-name> <job-name> <job-file>¥n", argv[0]);
                return (1);
        }

        /*
         * set submitinf
         */
        sprintf_s(submitinf,sizeof(submitinf),"-q %s -j %s %s",argv[1],argv[2],argv[3]);

        /*
         * call Mp_SubmitJob() API
         */
        rtn = Mp_SubmitJob (NULL, submitinf, NULL, jobhost, jobname, &jobno);

        /*
         * check return code of Mp_SubmitJob() API
         */
        if (rtn == 0) {
                printf ("Job %s(%d.%s) was submitted.¥n", jobname, jobno, jobhost);
                return (0);
        } else if (rtn == -1) {
                printf ("GetLastError() is %d.¥n", GetLastError());
        } else if (rtn <= -1000) {
                printf ("Error code is %d.¥n", rtn);
                Mp_GetMJESerror (errmsg);
                printf ("errmsg = %s¥n", errmsg);
        }
        return (1);
}
```

【UNIX版】

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <errno.h>
#include "mjesapi.h"

int main (int argc, char **argv)
{
        char submitinf[1024+1];
        char jobhost [64+1];
        char jobname [64+1];
```

```
        int jobno = 0;
        int rtn = 0;
        char errmsg [256+1];

        /*
         * initialization working area
         */
        memset(submitinf, 0x00, sizeof(submitinf));
        memset(jobhost, 0x00, sizeof(jobhost));
        memset(jobname, 0x00, sizeof(jobname));
        memset(errmsg, 0x00, sizeof(errmsg));

        if (argc != 4) {
                printf ("Usage: %s <queue-name> <job-name> <job-file>¥n", argv[0]);
                return (1);
        }

        /*
         * set submitinf
         */
        sprintf(submitinf,"-q %s -j %s %s",argv[1],argv[2],argv[3]);

        /*
         * call Mp_SubmitJob() API
         */
        rtn = Mp_SubmitJob (NULL, submitinf, NULL, jobhost, jobname, &jobno);

        /*
         * check return code of Mp_SubmitJob() API
         */
        if (rtn == 0) {
                printf ("Job %s(%d.%s) was submitted.¥n", jobname, jobno, jobhost);
                return (0);
        } else if (rtn == -1) {
                printf ("errno is %d.¥n", errno);
        } else if (rtn <= -1000) {
                printf ("Error code is %d.¥n", rtn);
                Mp_GetMJESerror (errmsg);
                printf ("errmsg = %s¥n", errmsg);
        }
        return (1);
}
```

17.3.4　ジョブ保留API【Windows版】

ジョブ保留API(Mp_HoldJob)について説明します。

機能説明

ジョブを保留状態とし、実行されるのを抑止します。

記述形式

```
#include "f3cuapi.h"
long Mp_HoldJob (char *rhost,
                char *jobname,
                long jobno,
                char *jobhost,
                char *userforadmin)
```

パラメタの説明

rhost

NULLを指定してください。

jobname

保留するジョブのジョブ名を指定します。NULLを指定することもできます。NULLを指定した場合は、jobnoで指定されたジョブ番号によって保留するジョブを指定してください。

jobno

保留するジョブのジョブ番号を指定します。jobnameにNULLを指定した場合、および同名ジョブが複数存在する場合は、必ず指定してください。jobnameを指定していて、同名ジョブが存在しない場合は、ジョブ番号を省略可能です。省略する場合は、0を指定してください。

jobhost

NULLを指定してください。

userforadmin

システム管理者(Administratorsグループに属するユーザ)が他人のジョブを保留する場合に指定します。保留対象ジョブの所有者名を指定してください。所有者名を指定しない場合はNULLを指定してください。

復帰値

復帰値	意 味	対 処
0	正常にジョブは保留されました。	-
-1	システムエラーが発生しました。	保守情報収集ツールで“ジョブ実行制御”の情報を採取し、富士通技術員に連絡してください。
-1000以下	ジョブ実行制御において、エラーが検出されたため、保留処理を中断しました。 なお、エラーメッセージ獲得APIを使用して、エラーメッセージを獲得することができます。	エラーメッセージ獲得APIを使用して、エラーメッセージを獲得し、そのエラーメッセージに対応した処置を実施してください。

注意事項

引数が無効アドレスを指している場合、当関数は異常終了し、その動作は定義されません。

プログラム例

ジョブ保留APIのプログラム例を以下に示します。

```
#include <windows.h>
#include <stdio.h>
#include ”f3cuapi.h”

int main (int argc, char **argv)
{
        long jobno = 0;
        long rtn = 0;
        char errmsg[256+1];
```

```
        /*
         * initialization working area
         */
        memset(errmsg, 0x00, sizeof(errmsg));

        if (argc != 2) {
                printf ("Usage: %s <jobno>¥n", argv[0]);
                return (1);
        }

        /*
         * set jobno
         */
        jobno = atoi(argv[1]);

        /*
         *  call Mp_HoldJob() API
         */
        rtn = Mp_HoldJob (NULL, NULL, jobno, NULL, NULL);

        /*
         * check return code of Mp_HoldJob() API
         */
        if (rtn == 0) {
                printf ("Job %d has been held.¥n", jobno);
                return (0);
        } else if (rtn == -1) {
                printf ("GetLastError() is %d.¥n", GetLastError());
        } else if (rtn <= -1000) {
                printf ("Error code is %d.¥n", rtn);
                Mp_GetMJESerror (errmsg);
                printf ("errmsg = %s¥n", errmsg);
        }
        return (1);
}
```

EE

17.3.5　ジョブ保留API/EE【Windows版】

ジョブ保留API/EE(Mp_HoldJob_Ex)について説明します。

機能説明

Systemwalker Operation Manager EEで、ジョブを保留状態とし、実行されるのを抑止します。

記述形式

```
#include "f3cuapi.h"
long Mp_HoldJob_Ex (char *rhost,
                    char *jobname,
                    long jobno,
                    char *jobhost,
                    char *userforadmin,
                    int system_num)
```

パラメタの説明

rhost

NULLを指定してください。

jobname

保留するジョブのジョブ名を指定します。NULLを指定することもできます。NULLを指定した場合は、jobnoで指定されたジョブ番号によって保留するジョブを指定してください。

jobno

保留するジョブのジョブ番号を指定します。jobnameにNULLを指定した場合、および同名ジョブが複数存在する場合は、必ず指定してください。jobnameを指定していて、同名ジョブが存在しない場合は、ジョブ番号を省略可能です。省略する場合は、0を指定してください。

jobhost

NULLを指定してください。

userforadmin

システム管理者(Administratorsグループに属するユーザ)が他人のジョブを保留する場合に指定します。保留対象ジョブの所有者名を指定してください。所有者名を指定しない場合はNULLを指定してください。

system_num

操作対象のサブシステムを、0から9の範囲の番号で指定します。

復帰値

ジョブ保留API(Mp_HoldJob_Ex)は、以下に示す値を返します。

復帰値	意 味	対 処
0	正常にジョブは保留されました。	-
-1	システムエラーが発生しました。	保守情報収集ツールで“ジョブ実行制御”の情報を採取し、富士通技術員に連絡してください。
-1000以下	ジョブ実行制御において、エラーが検出されたため、保留処理を中断しました。 なお、エラーメッセージ獲得APIを使用して、エラーメッセージを獲得することができます。	エラーメッセージ獲得APIを使用して、エラーメッセージを獲得し、そのエラーメッセージに対応した処置を実施してください。

注意事項

引数が無効アドレスを指している場合、当関数は異常終了し、その動作は定義されません。

プログラム例

ジョブ保留API/EEのプログラム例を以下に示します。

```
#include <windows.h>
#include <stdio.h>
#include "f3cuapi.h"

int main (int argc, char **argv)
{
        long jobno = 0;
        long rtn = 0;
        char errmsg[256+1];
        int system_num;
```

```
        /*
         * initialization working area
         */
        memset(errmsg, 0x00, sizeof(errmsg));

        if (argc != 2) {
                printf ("Usage: %s <jobno>¥n", argv[0]);
                return (1);
        }

        /*
         * set jobno
         */
        jobno = atoi(argv[1]);

        /*
         *  call Mp_HoldJob_Ex() API
         */
        system_num = 0;
        rtn = Mp_HoldJob_Ex (NULL, NULL, jobno, NULL, NULL, system_num);

        /*
         * check return code of Mp_HoldJob_Ex() API
         */
        if (rtn == 0) {
                printf ("Job %d has been held.¥n", jobno);
                return (0);
        } else if (rtn == -1) {
                printf ("GetLastError() is %d.¥n", GetLastError());
        } else if (rtn <= -1000) {
                printf ("Error code is %d.¥n", rtn);
                Mp_GetMJESerror (errmsg);
                printf ("errmsg = %s¥n", errmsg);
        }
        return (1);
}
```

17.3.6　ジョブ保留解除API【Windows版】

ジョブ保留解除API(Mp_RlsJob)について説明します。

機能説明

ジョブの保留状態を解除し、実行可能状態とします。

記述形式

```
#include "f3cuapi.h"
long Mp_RlsJob (char *rhost,
                char *jobname,
                long jobno,
                char *jobhost,
                char *userforadmin)
```

パラメタの説明

rhost

NULLを指定してください。

jobname

保留解除するジョブのジョブ名を指定します。NULLを指定することもできます。NULLを指定した場合は、jobnoで指定されたジョブ番号によって保留解除するジョブを指定してください。

jobno

保留解除するジョブのジョブ番号を指定します。jobnameにNULLを指定した場合、および同名ジョブが複数存在する場合は、必ず指定してください。jobnameを指定していて、同名ジョブが存在しない場合は、ジョブ番号を省略可能です。省略する場合は、0を指定してください。

jobhost

NULLを指定してください。

userforadmin

システム管理者(Administratorsグループに属するユーザ)が他人のジョブを保留解除する場合に指定します。保留解除対象ジョブの所有者名を指定してください。所有者名を指定しない場合はNULLを指定してください。

復帰値

復帰値	意 味	対 処
0	正常にジョブは保留解除されました。	-
-1	システムエラーが発生しました。	保守情報収集ツールで“ジョブ実行制御”の情報を採取し、富士通技術員に連絡してください。
-1000以下	ジョブ実行制御において、エラーが検出されたため、保留解除処理を中断しました。なお、エラーメッセージ獲得APIを使用して、エラーメッセージを獲得することができます。	エラーメッセージ獲得APIを使用して、エラーメッセージを獲得し、そのエラーメッセージに対応した処置を実施してください。

注意事項

引数が無効アドレスを指している場合、当関数は異常終了し、その動作は定義されません。

プログラム例

ジョブ保留解除APIのプログラム例を以下に示します。

```
#include <windows.h>
#include <stdio.h>
#include "f3cuapi.h"

int main (int argc, char **argv)
{
        long jobno = 0;
        long rtn = 0;
        char errmsg[256+1];

        /*
         * initialization working area
         */
        memset(errmsg, 0x00, sizeof(errmsg));

        if (argc != 2) {
                printf ("Usage: %s <jobno>¥n", argv[0]);
```

```
                return (1);
        }

        /*
         * set jobno
         */
        jobno = atoi(argv[1]);

        /*
         *  call Mp_RlsJob() API
         */
        rtn = Mp_RlsJob (NULL, NULL, jobno, NULL, NULL);

        /*
         * check return code of Mp_RlsJob() API
         */
        if (rtn == 0) {
                printf ("Job %d has been released.¥n", jobno);
                return (0);
        } else if (rtn == -1) {
                printf ("GetLastError() is %d.¥n", GetLastError());
        } else if (rtn <= -1000) {
                printf ("Error code is %d.¥n", rtn);
                Mp_GetMJEServer (errmsg);
                printf ("errmsg = %s¥n", errmsg);
        }
        return (1);
}
```

EE

17.3.7　ジョブ保留解除API/EE【Windows版】

ジョブ保留解除API/EE(Mp_RlsJob_Ex)について説明します。

機能説明

Systemwalker Operation Manager EEで、ジョブの保留状態を解除し、実行可能状態とします。

記述形式

```
#include "f3cuapi.h"
long Mp_RlsJob_Ex (char *rhost,
                char *jobname,
                long jobno,
                char *jobhost,
                char *userforadmin,
                int system_num)
```

パラメタの説明

rhost

NULLを指定してください。

jobname

保留を解除するジョブのジョブ名を指定します。NULLを指定することもできます。NULLを指定した場合は、jobnoで指定されたジョブ番号によって保留解除するジョブを指定してください。

jobno

保留を解除するジョブのジョブ番号を指定します。jobnameにNULLを指定した場合、および同名ジョブが複数存在する場合は、必ず指定してください。jobnameを指定していて、同名ジョブが存在しない場合は、ジョブ番号を省略可能です。省略する場合は、0を指定してください。

jobhost

NULLを指定してください。

userforadmin

システム管理者(Administratorsグループに属するユーザ)が他人のジョブを保留解除する場合に指定します。保留解除対象ジョブの所有者名を指定してください。所有者名を指定しない場合はNULLを指定してください。

system_num

操作対象のサブシステムを、0から9の範囲の番号で指定します。

復帰値

復帰値	意 味	対 処
0	正常にジョブは保留解除されました。	-
-1	システムエラーが発生しました。	保守情報収集ツールで“ジョブ実行制御”の情報を採取し、富士通技術員に連絡してください。
-1000以下	ジョブ実行制御において、エラーが検出されたため、保留解除処理を中断しました。なお、エラーメッセージ獲得APIを使用して、エラーメッセージを獲得することができます。	エラーメッセージ獲得APIを使用して、エラーメッセージを獲得し、そのエラーメッセージに対応した処置を実施してください。

注意事項

引数が無効アドレスを指している場合、当関数は異常終了し、その動作は定義されません。

プログラム例

ジョブ保留解除API/EEのプログラム例を以下に示します。

```
#include <windows.h>
#include <stdio.h>
#include "f3cuapi.h"

int main (int argc, char **argv)
{
        long jobno = 0;
        long rtn = 0;
        char errmsg[256+1];
        int system_num;

        /*
         * initialization working area
         */
        memset(errmsg, 0x00, sizeof(errmsg));

        if (argc != 2) {
```

```
                printf ("Usage: %s <jobno>¥n", argv[0]);
                return (1);
        }

        /*
         * set jobno
         */
        jobno = atoi(argv[1]);

        /*
         *  call Mp_RlsJob_Ex() API
         */
        system_num = 0;
        rtn = Mp_RlsJob_Ex (NULL, NULL, jobno, NULL, NULL, system_num);

        /*
         * check return code of Mp_RlsJob_Ex() API
         */
        if (rtn == 0) {
                printf ("Job %d has been released.¥n", jobno);
                return (0);
        } else if (rtn == -1) {
                printf ("GetLastError() is %d.¥n", GetLastError());
        } else if (rtn <= -1000) {
                printf ("Error code is %d.¥n", rtn);
                Mp_GetMJESerror (errmsg);
                printf ("errmsg = %s¥n", errmsg);
        }
        return (1);
}
```

17.3.8 ジョブ削除API

ジョブ削除API(Mp_DeleteJob)について説明します。

機能説明

ジョブをSystemwalker Operation Managerを導入しているサーバから削除します。実行中ジョブは実行が中止されます。

記述形式

【Windows版】

```
#include "f3cuapi.h"
long Mp_DeleteJob (char *rhost,
                char *jobname,
                long jobno,
                char *jobhost,
                char *userforadmin)
```

【UNIX版】

```
#include "mjesapi.h "
int Mp_DeleteJob (char *rhost,
                char *jobname,
                int jobno,
                char *jobhost,
                char *userforadmin)
```

パラメタの説明

rhost

NULLを指定してください。

jobname

削除するジョブのジョブ名を指定します。NULLを指定することもできます。NULLを指定した場合は、jobnoで指定されたジョブ番号によって削除するジョブを指定してください。

jobno

削除するジョブのジョブ番号を指定します。jobnameにNULLを指定した場合、および同名ジョブが複数存在する場合は、必ず指定してください。jobnameを指定していて、同名ジョブが存在しない場合は、ジョブ番号を省略可能です。省略する場合は、0を指定してください。

jobhost

NULLを指定してください。

userforadmin

システム管理者(Administratorsグループに属するユーザ/スーパーユーザ)が他人のジョブを削除する場合に指定します。削除対象ジョブの所有者名を指定してください。所有者名を指定しない場合はNULLを指定してください。

復帰値

復帰値	意 味	対 処
0	正常にジョブは削除されました。	-
-1	システムエラーが発生しました。	保守情報収集ツールで“ジョブ実行制御”の情報を採取し、富士通技術員に連絡してください。
-1000以下	ジョブ実行制御において、エラーが検出されたため、削除処理を中断しました。なお、エラーメッセージ獲得APIを使用して、エラーメッセージを獲得することができます。	エラーメッセージ獲得APIを使用して、エラーメッセージを獲得し、そのエラーメッセージに対応した処置を実施してください。

注意事項

引数が無効アドレスを指している場合、当関数は異常終了し、その動作は定義されません。

プログラム例

ジョブ削除APIのプログラム例を以下に示します。

【Windows版】

```
#include <windows.h>
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include "f3cuapi.h"

int main (int argc, char **argv)
{
        long jobno = 0;
        long rtn = 0;
```

```
        char errmsg[256+1];

        /*
         * initialization working area
         */
        memset(errmsg, 0x00, sizeof(errmsg));

        if (argc != 2) {
                printf ("Usage: %s <jobno>¥n", argv[0]);
                return (1);
        }

        /*
         * set jobno
         */
        jobno = atoi(argv[1]);

        /*
         *  call Mp_DeleteJob() API
         */
        rtn = Mp_DeleteJob (NULL, NULL, jobno, NULL, NULL);

        /*
         * check return code of Mp_DeleteJob() API
         */
        if (rtn == 0) {
                printf ("Job %d has been deleted.¥n", jobno);
                return (0);
        } else if (rtn == -1) {
                printf ("GetLastError() is %d.¥n", GetLastError());
        } else if (rtn <= -1000) {
                printf ("Error code is %d.¥n", rtn);
                Mp_GetMJESError (errmsg);
                printf ("errmsg = %s¥n", errmsg);
        }
        return (1);
}
```

【UNIX版】

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <errno.h>
#include "mjesapi.h"

int main (int argc, char **argv)
{
        int jobno = 0;
        int rtn = 0;
        char errmsg[256+1];

        /*
         * initialization working area
         */
        memset(errmsg, 0x00, sizeof(errmsg));

        if (argc != 2) {
                printf ("Usage: %s <jobno>¥n", argv[0]);
                return (1);
        }
```

```
        /*
         * set jobno
         */
        jobno = atoi(argv[1]);

        /*
         *  call Mp_DeleteJob() API
         */
        rtn = Mp_DeleteJob (NULL, NULL, jobno, NULL, NULL);

        /*
         * check return code of Mp_DeleteJob() API
         */
        if (rtn == 0) {
                printf ("Job %d has been deleted.¥n", jobno);
                return (0);
        } else if (rtn == -1) {
                printf ("errno is %d.¥n", errno);
        } else if (rtn <= -1000) {
                printf ("Error code is %d.¥n", rtn);
                Mp_GetMJESerror (errmsg);
                printf ("errmsg = %s¥n", errmsg);
        }
        return (1);
}
```

17.3.9 ジョブ削除API/EE

ジョブ削除API/EE(Mp_DeleteJob_Ex)について説明します。

機能説明

Systemwalker Operation Manager EEで、ジョブをSystemwalker Operation Managerを導入しているサーバから削除します。実行中ジョブは実行が中止されます。

記述形式

【Windows版】

```
#include "f3cuapi.h"
long Mp_DeleteJob_Ex (char *rhost,
                char *jobname,
                long jobno,
                char *jobhost,
                char *userforadmin,
                int system_num)
```

【UNIX版】

```
#include "mjesapi.h "
int Mp_DeleteJob_Ex (char *rhost,
                char *jobname,
                int jobno,
                char *jobhost,
                char *userforadmin,
                int system_num)
```

パラメタの説明

rhost

NULLを指定してください。

jobname

削除するジョブのジョブ名を指定します。NULLを指定することもできます。NULLを指定した場合は、jobnoで指定されたジョブ番号によって削除するジョブを指定してください。

jobno

削除するジョブのジョブ番号を指定します。jobnameにNULLを指定した場合、および同名ジョブが複数存在する場合は、必ず指定してください。jobnameを指定していて、同名ジョブが存在しない場合は、ジョブ番号を省略可能です。省略する場合は、0を指定してください。

jobhost

NULLを指定してください。

userforadmin

システム管理者(Administratorsグループに属するユーザ/スーパーユーザ)が他人のジョブを削除する場合に指定します。削除対象ジョブの所有者名を指定してください。所有者名を指定しない場合はNULLを指定してください。

system_num

操作対象のサブシステムを、0から9の範囲の番号で指定します。

復帰値

復帰値	意 味	対 処
0	正常にジョブは削除されました。	-
-1	システムエラーが発生しました。	保守情報収集ツールで“ジョブ実行制御”の情報を採取し、富士通技術員に連絡してください。
-1000以下	ジョブ実行制御において、エラーが検出されたため、削除処理を中断しました。なお、エラーメッセージ獲得APIを使用して、エラーメッセージを獲得することができます。	エラーメッセージ獲得APIを使用して、エラーメッセージを獲得し、そのエラーメッセージに対応した処置を実施してください。

注意事項

引数が無効アドレスを指している場合、当関数は異常終了し、その動作は定義されません。

プログラム例

ジョブ削除API/EEのプログラム例を以下に示します。

【Windows版】

```
#include <windows.h>
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include "f3cuapi.h"

int main (int argc, char **argv)
{
        long jobno = 0;
```

```
        long rtn = 0;
        char errmsg[256+1];
        int system_num;

        /*
         * initialization working area
         */
        memset(errmsg, 0x00, sizeof(errmsg));

        if (argc != 2) {
                printf ("Usage: %s <jobno>¥n", argv[0]);
                return (1);
        }

        /*
         * set jobno
         */
        jobno = atoi(argv[1]);

        /*
         *  call Mp_DeleteJob_Ex() API
         */
        system_num = 0;
        rtn = Mp_DeleteJob_Ex (NULL, NULL, jobno, NULL, NULL, system_num);

        /*
         * check return code of Mp_DeleteJob_Ex() API
         */
        if (rtn == 0) {
                printf ("Job %d has been deleted.¥n", jobno);
                return (0);
        } else if (rtn == -1) {
                printf ("GetLastError() is %d.¥n", GetLastError());
        } else if (rtn <= -1000) {
                printf ("Error code is %d.¥n", rtn);
                Mp_GetMJEServer (errmsg);
                printf ("errmsg = %s¥n", errmsg);
        }
        return (1);
}
```

【UNIX版】

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <errno.h>
#include "mjesapi.h"

int main (int argc, char **argv)
{
        int jobno = 0;
        int rtn = 0;
        char errmsg[256+1];
        int system_num;

        /*
         * initialization working area
         */
        memset(errmsg, 0x00, sizeof(errmsg));

        if (argc != 2) {
```

```
                printf ("Usage: %s <jobno>¥n", argv[0]);
                return (1);
        }

        /*
         * set jobno
         */
        jobno = atoi(argv[1]);

        /*
         *  call Mp_DeleteJob_Ex() API
         */
        system_num = 0;
        rtn = Mp_DeleteJob_Ex (NULL, NULL, jobno, NULL, NULL, system_num);

        /*
         * check return code of Mp_DeleteJob_Ex() API
         */
        if (rtn == 0) {
                printf ("Job %d has been deleted.¥n", jobno);
                return (0);
        } else if (rtn == -1) {
                printf ("errno is %d.¥n", errno);
        } else if (rtn <= -1000) {
                printf ("Error code is %d.¥n", rtn);
                Mp_GetMJESerror (errmsg);
                printf ("errmsg = %s¥n", errmsg);
        }
        return (1);
}
```

17.3.10　ジョブ移動API【Windows版】

ジョブ移動API(Mp_MoveJob)について説明します。

機能説明

ジョブを他のキューへ移動します。

記述形式

```
#include "f3cuapi.h"
long Mp_MoveJob (char *rhost,
                char *jobname,
                long jobno,
                char *jobhost,
                char *queue,
                char *userforadmin)
```

パラメタの説明

rhost

NULLを指定してください。

jobname

移動するジョブのジョブ名を指定します。NULLを指定することもできます。NULLを指定した場合は、jobnoで指定されたジョブ番号によって移動するジョブを指定してください。

jobno

移動するジョブのジョブ番号を指定します。jobnameにNULLを指定した場合、および同名ジョブが複数存在する場合は、必ず指定してください。jobnameを指定していて、同名ジョブが存在しない場合は、ジョブ番号を省略可能です。省略する場合は、0を指定してください。

jobhost

NULLを指定してください。

queue

移動先キュー名を指定します。

userforadmin

システム管理者(Administratorsグループに属するユーザ)が他人のジョブを移動する場合に指定します。移動対象ジョブの所有者名を指定してください。所有者名を指定しない場合はNULLを指定してください。

復帰値

復帰値	意 味	対 処
0	正常にジョブは移動されました。	-
-1	システムエラーが発生しました。	保守情報収集ツールで“ジョブ実行制御”の情報を採取し、富士通技術員に連絡してください。
-1000以下	ジョブ実行制御において、エラーが検出されたため、移動処理を中断しました。なお、エラーメッセージ獲得APIを使用して、エラーメッセージを獲得することができます。	エラーメッセージ獲得APIを使用して、エラーメッセージを獲得し、そのエラーメッセージに対応した処置を実施してください。

注意事項

引数が無効アドレスを指している場合、当関数は異常終了し、その動作は定義されません。

分散実行キューから他のキューへの移動、または他のキューから分散実行キューへの移動はできません。

プログラム例

ジョブ移動APIのプログラム例を以下に示します。

```
#include <windows.h>
#include <stdio.h>
#include "f3cuapi.h"

int main (int argc, char **argv)
{
        long jobno = 0;
        long rtn = 0;
        char errmsg[256+1];

        /*
         * initialization working area
         */
        memset(errmsg, 0x00, sizeof(errmsg));

        if (argc != 3) {
```

```
                printf ("Usage: %s <jobno> <queuename>¥n", argv[0]);
                return (1);
        }

        /*
         * set jobno
         */
        jobno = atoi(argv[1]);

        /*
         *  call Mp_MoveJob() API
         */
        rtn = Mp_MoveJob (NULL, NULL, jobno, NULL, argv[2],NULL);

        /*
         * check return code of Mp_MoveJob() API
         */
        if (rtn == 0) {
                printf ("Job %d has been moved.¥n", jobno);
                return (0);
        } else if (rtn == -1) {
                printf ("GetLastError() is %d.¥n", GetLastError());
        } else if (rtn <= -1000) {
                printf ("Error code is %d.¥n", rtn);
                Mp_GetMJESerror (errmsg);
                printf ("errmsg = %s¥n", errmsg);
        }
        return (1);
}
```

EE

17.3.11 ジョブ移動API/EE【Windows版】

ジョブ移動API/EE(Mp_MoveJob_Ex)について説明します。

機能説明

Systemwalker Operation Manager EEで、ジョブを他のキューへ移動します。

記述形式

```
#include "f3cuapi.h"
long Mp_MoveJob_Ex (char *rhost,
                char *jobname,
                long jobno,
                char *jobhost,
                char *queue,
                char *userforadmin,
                int system_num)
```

パラメタの説明

rhost

NULLを指定してください。

jobname

移動するジョブのジョブ名を指定します。NULLを指定することもできます。NULLを指定した場合は、jobnoで指定されたジョブ番号によって移動するジョブを指定してください。

jobno

移動するジョブのジョブ番号を指定します。jobnameにNULLを指定した場合、および同名ジョブが複数存在する場合は、必ず指定してください。jobnameを指定していて、同名ジョブが存在しない場合は、ジョブ番号を省略可能です。省略する場合は、0を指定してください。

jobhost

NULLを指定してください。

queue

移動先キュー名を指定します。

userforadmin

システム管理者(Administratorsグループに属するユーザ)が他人のジョブを移動する場合に指定します。移動対象ジョブの所有者名を指定してください。所有者名を指定しない場合はNULLを指定してください。

system_num

操作対象のサブシステムを、0から9の範囲の番号で指定します。

復帰値

復帰値	意 味	対 処
0	正常にジョブは移動されました。	-
-1	システムエラーが発生しました。	保守情報収集ツールで“ジョブ実行制御”の情報を採取し、富士通技術員に連絡してください。
-1000以下	ジョブ実行制御において、エラーが検出されたため、移動処理を中断しました。なお、エラーメッセージ獲得APIを使用して、エラーメッセージを獲得することができます。	エラーメッセージ獲得APIを使用して、エラーメッセージを獲得し、そのエラーメッセージに対応した処置を実施してください。

注意事項

引数が無効アドレスを指している場合、当関数は異常終了し、その動作は定義されません。

分散実行キューから他のキューへの移動、または他のキューから分散実行キューへの移動はできません。

プログラム例

ジョブ移動API/EEのプログラム例を以下に示します。

```
#include <windows.h>
#include <stdio.h>
#include "f3cuapi.h"

int main (int argc, char **argv)
{
        long jobno = 0;
        long rtn = 0;
        char errmsg[256+1];
        int system_num;

        /*
         * initialization working area
         */
```

```
        memset(errmsg, 0x00, sizeof(errmsg));

        if (argc != 3) {
                printf ("Usage: %s <jobno> <queuename>¥n", argv[0]);
                return (1);
        }

        /*
         * set jobno
         */
        jobno = atoi(argv[1]);

        /*
         *  call Mp_MoveJob_Ex() API
         */
        system_num = 0;
        rtn = Mp_MoveJob_Ex (NULL, NULL, jobno, NULL, argv[2], NULL, system_num);

        /*
         * check return code of Mp_MoveJob_Ex() API
         */
        if (rtn == 0) {
                printf ("Job %d has been moved.¥n", jobno);
                return (0);
        } else if (rtn == -1) {
                printf ("GetLastError() is %d.¥n", GetLastError());
        } else if (rtn <= -1000) {
                printf ("Error code is %d.¥n", rtn);
                Mp_GetMJESerror (errmsg);
                printf ("errmsg = %s¥n", errmsg);
        }
        return (1);
}
```

17.3.12　ジョブ状態獲得API

ジョブ状態獲得API(Mp_StatusJob)について説明します。

機能説明

ジョブの状態を獲得します。ジョブが終了状態である場合、ジョブの終了コードを獲得できます。

記述形式

【Windows版】

```
#include "f3cuapi.h"
long Mp_StatusJob (char *rhost,
                char *jobname,
                long jobno,
                char *jobhost,
                ulong *compcode,
                char *userforadmin)
```

【UNIX版】

```
#include "mjesapi.h"
int Mp_StatusJob (char *rhost,
                char *jobname,
                int jobno,
                char *jobhost,
```

```
        uint *compcode,
        char *userforadmin)
```

パラメタの説明

rhost

NULLを指定してください。

jobname

状態を獲得するジョブのジョブ名を指定します。NULLを指定することもできます。NULLを指定した場合は、jobnoで指定されたジョブ番号によって状態を獲得するジョブを指定してください。

jobno

状態を獲得するジョブのジョブ番号を指定します。jobnameにNULLを指定した場合、および同名ジョブが複数存在する場合は、必ず指定してください。jobnameを指定していて、同名ジョブが存在しない場合は、ジョブ番号を省略可能です。省略する場合は、0を指定してください。

jobhost

NULLを指定してください。

compcode

ジョブの終了コード返却用領域のアドレスを指定します。ジョブが終了状態の場合、終了コードが格納されます。

userforadmin

システム管理者(Administratorsグループに属するユーザ/スーパーユーザ)が他人のジョブの状態を獲得する場合に指定します。状態の獲得対象ジョブの所有者名を指定してください。所有者名を指定しない場合はNULLを指定してください。

復帰値

復帰値	意 味	対 処
0	指定したジョブは存在しません。	-
1	実行待ち状態です。	-
2	実行保留状態です。	-
3	実行中です。	-
4	出力保留状態です。	-
5	ジョブは終了状態です。	-
-1	システムエラーが発生しました。	保守情報収集ツールで“ジョブ実行制御”の情報を採取し、富士通技術員に連絡してください。
-1000以下	ジョブ実行制御において、エラーが検出されたため、状態の獲得処理を中断しました。なお、エラーメッセージ獲得APIを使用して、エラーメッセージを獲得することができます。	エラーメッセージ獲得APIを使用して、エラーメッセージを獲得し、そのエラーメッセージに対応した処置を実施してください。

注意事項

引数が無効アドレスを指している場合、当関数は異常終了し、その動作は定義されません。

プログラム例

ジョブ状態獲得APIのプログラム例を以下に示します。

【Windows版】

```
#include <windows.h>
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include "f3cuapi.h"

int main (int argc, char **argv)
{
        long jobno = 0;
        long rtn = 0;
        unsigned long compcode = 0;
        char errmsg[256+1];

        /*
         * initialization working area
         */
        memset(errmsg, 0x00, sizeof(errmsg));

        if (argc != 2) {
                printf ("Usage: %s <jobno>¥n", argv[0]);
                return (1);
        }

        /*
         * set jobno
         */
        jobno = atoi(argv[1]);

        /*
         *  call Mp_StatusJob() API
         */
        rtn = Mp_StatusJob (NULL, NULL, jobno, NULL, &compcode, NULL);

        /*
         * check return code of Mp_StatusJob() API
         */
        if (rtn == 0) {
                printf ("Job %d is not found.¥n", jobno);
                return (0);
        } else if (rtn == 1) {
                printf ("Job %d is waiting for execution.¥n", jobno);
                return (0);
        } else if (rtn == 2) {
                printf ("Job %d is holding for execution.¥n", jobno);
                return (0);
        } else if (rtn == 3) {
                printf ("Job %d is executing.¥n", jobno);
                return (0);
        } else if (rtn == 4) {
                printf ("Job %d is holding for output.¥n", jobno);
                return (0);
        } else if (rtn == 5) {
                printf ("Job %d was completed. completion code = %d¥n", jobno, compcode);
                return (0);
        } else if (rtn == -1) {
                printf ("GetLastError() is %d.¥n", GetLastError());
        } else if (rtn <= -1000) {
```

```
                printf ("Error code is %d.¥n", rtn);
                Mp_GetMJESerror (errmsg);
                printf ("errmsg = %s¥n", errmsg);
        }
        return (1);
}
```

【UNIX版】

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <errno.h>
#include "mjesapi.h"

int main (int argc, char **argv)
{
        int jobno = 0;
        int rtn = 0;
        unsigned int compcode = 0;
        char errmsg[256+1];

        /*
         * initialization working area
         */
        memset(errmsg, 0x00, sizeof(errmsg));

        if (argc != 2) {
                printf ("Usage: %s <jobno>¥n", argv[0]);
                return (1);
        }

        /*
         * set jobno
         */
        jobno = atoi(argv[1]);

        /*
         *  call Mp_StatusJob() API
         */
        rtn = Mp_StatusJob (NULL, NULL, jobno, NULL, &compcode, NULL);

        /*
         * check return code of Mp_StatusJob() API
         */
        if (rtn == 0) {
                printf ("Job %d is not found.¥n", jobno);
                return (0);
        } else if (rtn == 1) {
                printf ("Job %d is waiting for execution.¥n", jobno);
                return (0);
        } else if (rtn == 2) {
                printf ("Job %d is holding for execution.¥n", jobno);
                return (0);
        } else if (rtn == 3) {
                printf ("Job %d is executing.¥n", jobno);
                return (0);
        } else if (rtn == 4) {
                printf ("Job %d is holding for output.¥n", jobno);
                return (0);
        } else if (rtn == 5) {
                printf ("Job %d was completed. completion code = %d¥n", jobno, compcode);
```

```
                return (0);
        } else if (rtn == -1) {
                printf ("errno is %d.¥n", errno);
        } else if (rtn <= -1000) {
                printf ("Error code is %d.¥n", rtn);
                Mp_GetMJESerror (errmsg);
                printf ("errmsg = %s¥n", errmsg);
        }
        return (1);
}
```

17.3.13 ジョブ状態獲得API/EE

ジョブ状態獲得API/EE(Mp_StatusJob_Ex)について説明します。

機能説明

Systemwalker Operation Manager EEで、ジョブの状態を獲得します。ジョブが終了状態である場合、ジョブの終了コードを獲得できます。

記述形式

【Windows版】

```
#include "f3cuapi.h"
long Mp_StatusJob_Ex (char *rhost,
                char *jobname,
                long jobno,
                char *jobhost,
                ulong *compcode,
                char *userforadmin,
                int system_num)
```

【UNIX版】

```
#include "mjesapi.h"
int Mp_StatusJob_Ex (char *rhost,
                char *jobname,
                int jobno,
                char *jobhost,
                uint *compcode,
                char *userforadmin,
                int system_num)
```

パラメタの説明

rhost

NULLを指定してください。

jobname

状態を獲得するジョブのジョブ名を指定します。NULLを指定することもできます。NULLを指定した場合は、jobnoで指定されたジョブ番号によって状態を獲得するジョブを指定してください。

jobno

状態を獲得するジョブのジョブ番号を指定します。jobnameにNULLを指定した場合、および同名ジョブが複数存在する場合は、必ず指定してください。jobnameを指定していて、同名ジョブが存在しない場合は、ジョブ番号を省略可能です。省略する場合は、0を指定してください。

jobhost

NULLを指定してください。

compcode

ジョブの終了コード返却用領域のアドレスを指定します。ジョブが終了状態の場合、終了コードが格納されます。

userforadmin

システム管理者(Administratorsグループに属するユーザ/スーパーユーザ)が他人のジョブの状態を獲得する場合に指定します。状態の獲得対象ジョブの所有者名を指定してください。所有者名を指定しない場合はNULLを指定してください。

system_num

操作対象のサブシステムを、0から9の範囲の番号で指定します。

復帰値

復帰値	意 味	対 処
0	指定したジョブは存在しません。	-
1	実行待ち状態です。	-
2	実行保留状態です。	-
3	実行中です。	-
4	出力保留状態です。	-
5	ジョブは終了状態です。	-
-1	システムエラーが発生しました。	保守情報収集ツールで“ジョブ実行制御”の情報を採取し、富士通技術員に連絡してください。
-1000以下	ジョブ実行制御において、エラーが検出されたため、状態の獲得処理を中断しました。なお、エラーメッセージ獲得APIを使用して、エラーメッセージを獲得することができます。	エラーメッセージ獲得APIを使用して、エラーメッセージを獲得し、そのエラーメッセージに対応した処置を実施してください。

注意事項

引数が無効アドレスを指している場合、当関数は異常終了し、その動作は定義されません。

プログラム例

ジョブ状態獲得API/EEのプログラム例を以下に示します。

【Windows版】

```
#include <windows.h>
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include "f3cuapi.h"

int main (int argc, char **argv)
{
        long jobno = 0;
        long rtn = 0;
```

```
        unsigned long compcode = 0;
        char errmsg[256+1];
        int system_num;

        /*
         * initialization working area
         */
        memset(errmsg, 0x00, sizeof(errmsg));

        if (argc != 2) {
                printf ("Usage: %s <jobno>¥n", argv[0]);
                return (1);
        }

        /*
         * set jobno
         */
        jobno = atoi(argv[1]);

        /*
         *  call Mp_StatusJob_Ex() API
         */
        system_num = 0;
        rtn = Mp_StatusJob_Ex (NULL, NULL, jobno, NULL, &compcode, NULL, system_num);

        /*
         * check return code of Mp_StatusJob_Ex() API
         */
        if (rtn == 0) {
                printf ("Job %d is not found.¥n", jobno);
                return (0);
        } else if (rtn == 1) {
                printf ("Job %d is waiting for execution.¥n", jobno);
                return (0);
        } else if (rtn == 2) {
                printf ("Job %d is holding for execution.¥n", jobno);
                return (0);
        } else if (rtn == 3) {
                printf ("Job %d is executing.¥n", jobno);
                return (0);
        } else if (rtn == 4) {
                printf ("Job %d is holding for output.¥n", jobno);
                return (0);
        } else if (rtn == 5) {
                printf ("Job %d was completed. completion code = %d¥n", jobno, compcode);
                return (0);
        } else if (rtn == -1) {
                printf ("GetLastError() is %d.¥n", GetLastError());
        } else if (rtn <= -1000) {
                printf ("Error code is %d.¥n", rtn);
                Mp_GetMJEServer (errmsg);
                printf ("errmsg = %s¥n", errmsg);
        }
        return (1);
}
```

【UNIX版】

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <errno.h>
```

```
#include "mjesapi.h"

int main (int argc, char **argv)
{
        int jobno = 0;
        int rtn = 0;
        unsigned int compcode = 0;
        char errmsg[256+1];
        int system_num;

        /*
         * initialization working area
         */
        memset(errmsg, 0x00, sizeof(errmsg));

        if (argc != 2) {
                printf ("Usage: %s <jobno>¥n", argv[0]);
                return (1);
        }

        /*
         * set jobno
         */
        jobno = atoi(argv[1]);

        /*
         *  call Mp_StatusJob_Ex() API
         */
        system_num = 0;
        rtn = Mp_StatusJob_Ex (NULL, NULL, jobno, NULL, &compcode, NULL, system_num);

        /*
         * check return code of Mp_StatusJob_Ex() API
         */
        if (rtn == 0) {
                printf ("Job %d is not found.¥n", jobno);
                return (0);
        } else if (rtn == 1) {
                printf ("Job %d is waiting for execution.¥n", jobno);
                return (0);
        } else if (rtn == 2) {
                printf ("Job %d is holding for execution.¥n", jobno);
                return (0);
        } else if (rtn == 3) {
                printf ("Job %d is executing.¥n", jobno);
                return (0);
        } else if (rtn == 4) {
                printf ("Job %d is holding for output.¥n", jobno);
                return (0);
        } else if (rtn == 5) {
                printf ("Job %d was completed. completion code = %d¥n", jobno, compcode);
                return (0);
        } else if (rtn == -1) {
                printf ("errno is %d.¥n", errno);
        } else if (rtn <= -1000) {
                printf ("Error code is %d.¥n", rtn);
                Mp_GetMJEServer (errmsg);
                printf ("errmsg = %s¥n", errmsg);
        }
        return (1);
}
```

17.3.14　ジョブ一覧情報獲得API【Windows版】

ジョブ一覧情報獲得API(Mp_ListJob)について説明します。

機能説明

ジョブ情報の一覧を獲得します。当関数発行元プロセスのユーザが所有するすべてのジョブの情報が獲得されます。システム管理者(Administratorsグループに属するユーザ)の場合、Systemwalker Operation Managerを導入しているサーバ内のすべてのジョブの情報が獲得されます。ただし、userforadminを指定した場合は、当該ユーザが所有するジョブの情報のみが獲得されます。

記述形式

```
#include "f3cuapi.h"
long Mp_ListJob (char *rhost,
                 long mode,
                 void **jobinf,
                 char *userforadmin)
```

パラメタの説明

rhost

NULLを指定してください。

mode

以下に示す処理モードを指定します。GETJOBを行った後、必ずFREEJOBを行ってください。

GETJOB:

ジョブ一覧情報を獲得します。

FREEJOB:

GETJOBで獲得したジョブ一覧情報領域を解放します。

jobinf

GETJOBの場合:

ジョブ情報構造体(Mp_JOBINF_1)の配列のアドレスを通知する領域のアドレスを指定します。

FREEJOBの場合:

GETJOBで返却されたジョブ情報構造体の配列のアドレスを通知した領域のアドレスをそのまま指定してください。また、GETJOBを行った後FREEJOBを行うまでの間、該当領域の内容を変更しないようにしてください。

userforadmin

システム管理者(Administratorsグループに属するユーザ)が他人のジョブの状態のみを獲得する場合に指定します。状態の獲得対象ジョブの所有者名を指定してください。所有者名を指定しない場合はNULLを指定してください。

復帰値

復帰値	意 味	対 処
0	該当するジョブが存在しません。	-
1以上	正常にジョブ情報の一覧が獲得されました。ジョブ数が返却されます。	-

復帰値	意 味	対 処
-1	システムエラーが発生しました。	保守情報収集ツールで“ジョブ実行制御”の情報を採取し、富士通技術員に連絡してください。
-1000以下	ジョブ実行制御において、エラーが検出されたため、処理を中断しました。なお、エラーメッセージ獲得APIを使用して、エラーメッセージを獲得することができます。	エラーメッセージ獲得APIを使用して、エラーメッセージを獲得し、そのエラーメッセージに対応した処置を実施してください。

注意事項

引数が無効アドレスを指している場合、当関数は異常終了し、その動作は定義されません。

ジョブ情報構造体(Mp_JOBINF_1)の形式

ジョブ情報構造体の形式を以下に示します。

当構造体は、INCLUDEファイル“f3cuapi.h”内に定義されます。

```
typedef struct _Mp_JOBINF_1{
  char      jobname[];        /* ジョブ名                 */
  char      jobhost[];        /* 投入元ホスト名           */
  long      jobno;            /* ジョブ番号               */
  char      user[];           /* ジョブ所有利用者         */
  char      queue[];          /* キュー名                 */
  long      jobtype;          /* ジョブ種類               */
  long      status;           /* ジョブ状態               */
  char      comment[];        /* 投入時に指定したコメント*/

}Mp_JOBINF_1
```

jobname

ジョブ名が設定されます。

jobhost

投入元ホスト名が設定されます。

jobno

ジョブの番号が設定されます。

user

ジョブの所有利用者が設定されます。

queue

キュー名が設定されます。

jobtype

ジョブの種類が設定されます。設定されるジョブの種類を以下に示します。

値	種 類
1	デマンドジョブ
2	スケジュールジョブ

status

ジョブの状態が設定されます。設定されるジョブの状態を以下に示します。

値	状 態
1	実行待ち
2	実行保留
3	実行中
4	出力保留

comment

投入時に指定したコメントが設定されます。

プログラム例

ジョブ一覧情報獲得APIのプログラム例を以下に示します。

```
#include <windows.h>
#include <stdio.h>
#include "f3cuapi.h"

int main (int argc, char **argv)
{
        Mp_JOBINF_1 *jobinf = 0;
        Mp_JOBINF_1 *jinf = 0;

        long mode;
        long idx;
        long rtn = 0;
        char errmsg[256+1];

        /*
         * initialization working area
         */
        memset(errmsg, 0x00, sizeof(errmsg));

        mode = GETJOB;
        /*
         *  call Mp_ListJob() API
         */
        rtn = Mp_ListJob (NULL, mode, (void**)&jobinf, NULL);

        /*
         * check return code of Mp_ListJob() API
         */
        if (rtn == -1) {
                printf ("GetLastError() is %d.¥n", GetLastError());
                return (1);
        } else if (rtn <= -1000) {
                printf ("Error code is %d.¥n", rtn);
                Mp_GetMJESerror (errmsg);
                printf ("errmsg = %s¥n", errmsg);
                return (1);
        } else if (rtn == 0) {
                printf ("Job is nothing.¥n");
                return (0);
        }

        jinf = jobinf;

        printf("job name             jobno user name           status¥n");
```

```
        for (idx = 0; idx < rtn; idx++) {

                printf("%20.20s %5.5d %20.20s ",
                        jinf->jobname, jinf->jobno, jinf->user);
                switch(jinf->status) {
                case 1  :
                        printf("waiting   ");
                        break;
                case 2  :
                        printf("holding   ");
                        break;
                case 3  :
                        printf("running   ");
                        break;
                case 4  :
                        printf("outhold   ");
                        break;
                }
                printf("¥n");
                jinf++;

        }

        mode = FREEJOB;

        /*
         *  call Mp_ListJob() API
         */
        Mp_ListJob (NULL, mode, (void**)&jobinf, NULL);

        return(0);
}
```

17.3.15 ジョブ一覧情報獲得API/EE【Windows版】

ジョブ一覧情報獲得API/EE(Mp_ListJob_Ex)について説明します。

機能説明

Systemwalker Operation Manager EEで、ジョブ情報の一覧を獲得します。当関数発行元プロセスのユーザが所有するすべてのジョブの情報が獲得されます。システム管理者(Administratorsグループに属するユーザ)の場合、Systemwalker Operation Managerを導入しているサーバ内のサブシステムごとのすべてのジョブ情報が獲得されます。ただし、userforadminを指定した場合は、当該ユーザが所有するジョブの情報のみが獲得されます。

記述形式

```
#include "f3cuapi.h"
long Mp_ListJob_Ex (char *rhost,
                long mode,
                void **jobinf,
                char *userforadmin,
                int system_num)
```

パラメタの説明

rhost

NULLを指定してください。

mode

以下に示す処理モードを指定します。GETJOBを行った後、必ずFREEJOBを行ってください。

GETJOB:

ジョブ一覧情報を獲得します。

FREEJOB:

GETJOBで獲得したジョブ一覧情報領域を解放します。

jobinf

GETJOBの場合:

ジョブ情報構造体(Mp_JOBINF_1)の配列のアドレスを通知する領域のアドレスを指定します。

FREEJOBの場合:

GETJOBで返却されたジョブ情報構造体の配列のアドレスを通知した領域のアドレスをそのまま指定してください。また、GETJOBを行った後FREEJOBを行うまでの間、該当領域の内容を変更しないようにしてください。

userforadmin

システム管理者(Administratorsグループに属するユーザ)が他人のジョブの状態のみを獲得する場合に指定します。状態の獲得対象ジョブの所有者名を指定してください。所有者名を指定しない場合はNULLを指定してください。

system_num

操作対象のサブシステムを、0から9の範囲の番号で指定します。

復帰値

復帰値	意 味	対 処
0	該当するジョブが存在しません。	-
1以上	正常にジョブ情報の一覧が獲得されたことを示し、そのジョブ数が返却されます。	-
-1	システムエラーが発生しました。	保守情報収集ツールで“ジョブ実行制御”の情報を採取し、富士通技術員に連絡してください。
-1000以下	ジョブ実行制御において、エラーが検出されたため、処理を中断しました。なお、エラーメッセージ獲得APIを使用して、エラーメッセージを獲得することができます。	エラーメッセージ獲得APIを使用して、エラーメッセージを獲得し、そのエラーメッセージに対応した処置を実施してください。

注意事項

引数が無効アドレスを指している場合、当関数は異常終了し、その動作は定義されません。

ジョブ情報構造体(Mp_JOBINF_1)の形式

ジョブ情報構造体の形式を以下に示します。

当構造体は、INCLUDEファイル"f3cuapi.h"内に定義されます。

```
typedef struct _Mp_JOBINF_1{
  char      jobname[];        /* ジョブ名                */
  char      jobhost[];        /* 投入元ホスト名          */
  long      jobno;            /* ジョブ番号              */
```

```
  char      user[];          /* ジョブ所有利用者        */
  char      queue[];         /* キュー名                */
  long      jobtype;         /* ジョブ種類              */
  long      status;          /* ジョブ状態              */
  char      comment[];       /* 投入時に指定したコメント*/
}Mp_JOBINF_1
```

以下に項目ごとの説明を記述します。

jobname

ジョブ名が設定されます。

jobhost

投入元ホスト名が設定されます。

jobno

ジョブの番号が設定されます。

user

ジョブの所有利用者が設定されます。

queue

キュー名が設定されます。

jobtype

ジョブの種類が設定されます。設定されるジョブの種類を以下に示します。

値	種 類
1	デマンドジョブ
2	スケジュールジョブ

status

ジョブの状態が設定されます。設定されるジョブの状態を以下に示します。

値	状 態
1	実行待ち
2	実行保留
3	実行中
4	出力保留

comment

投入時に指定したコメントが設定されます。

プログラム例

ジョブ一覧情報獲得API/EEのプログラム例を以下に示します。

```
#include <windows.h>
#include <stdio.h>
#include "f3cuapi.h"

int main (int argc, char **argv)
{
        Mp_JOBINF_1 *jobinf = 0;
        Mp_JOBINF_1 *jinf = 0;
```

```
long mode;
long idx;
long rtn = 0;
char errmsg[256+1];
int system_num;

/*
 * initialization working area
 */
memset(errmsg, 0x00, sizeof(errmsg));

mode = GETJOB;

/*
 *  call Mp_ListJob_Ex() API
 */
system_num = 0;
rtn = Mp_ListJob_Ex (NULL, mode, (void**)&jobinf, NULL, system_num);

/*
 * check return code of Mp_ListJob_Ex() API
 */
if (rtn == -1) {
        printf ("GetLastError() is %d.¥n", GetLastError());
        return (1);
} else if (rtn <= -1000) {
        printf ("Error code is %d.¥n", rtn);
        Mp_GetMJESerror (errmsg);
        printf ("errmsg = %s¥n", errmsg);
        return (1);
} else if (rtn == 0) {
        printf ("Job is nothing.¥n");
        return (0);
}

jinf = jobinf;

printf("job name             jobno user name            status¥n");
for (idx = 0; idx < rtn; idx++) {

        printf("%20.20s %5.5d %20.20s ",
                jinf->jobname, jinf->jobno, jinf->user);
        switch(jinf->status) {
        case 1  :
                printf("waiting   ");
                break;
        case 2  :
                printf("holding   ");
                break;
        case 3  :
                printf("running   ");
                break;
        case 4  :
                printf("outhold   ");
                break;
        }
        printf("¥n");
        jinf++;

}

mode = FREEJOB;
```

```
        /*
         *  call Mp_ListJob_Ex() API
         */
        Mp_ListJob_Ex (NULL, mode, (void**)&jobinf, NULL, system_num);

        return(0);
}
```

17.3.16　キュー一覧情報獲得API【Windows版】

キュー一覧情報獲得API(Mp_ListQueue)について説明します。

機能説明

キュー情報の一覧を獲得します。

記述形式

```
#include "f3cuapi.h"
long Mp_ListQueue (char *rhost,
                   long mode,
                   void **queueinf)
```

パラメタの説明

rhost

NULLを指定してください。

mode

以下に示す処理モードを指定します。

GETQSTを行った後、必ずFREEQSTを行ってください。

GETQST:

キュー一覧情報を獲得します。

FREEQST:

GETQSTで獲得したキュー一覧情報領域を解放します。

queueinf

GETQSTの場合:

キュー情報構造体(Mp_QUEUEINF_1)配列のアドレスを通知する領域のアドレスを指定します。

FREEQSTの場合:

GETQSTで返却されたジョブ情報構造体の配列のアドレスを通知した領域のアドレスをそのまま指定してください。また、GETQSTを行った後FREEQSTを行うまでの間、該当領域の内容を変更しないようにしてください。

復帰値

復帰値	意 味	対 処
1以上	正常にキュー情報の一覧が獲得されたことを示し、そのキューの数が返却されます。	-

復帰値	意 味	対 処
-1	システムエラーが発生しました。	保守情報収集ツールで“ジョブ実行制御”の情報を採取し、富士通技術員に連絡してください。
-1000以下	ジョブ実行制御において、エラーが検出されたため、処理を中断しました。なお、エラーメッセージ獲得APIを使用して、エラーメッセージを獲得することができます。	エラーメッセージ獲得APIを使用して、エラーメッセージを獲得し、そのエラーメッセージに対応した処置を実施してください。

注意事項

引数が無効アドレスを指している場合、当関数は異常終了し、その動作は定義されません。

キュー情報構造体(Mp_QUEUEINF_1)の形式

キュー情報構造体の形式を以下に示します。

当構造体は、INCLUDEファイル“f3cuapi.h”内に定義されます。

```
typedef struct _Mp_QUEUEINF_1 {
  char      queuename[];        /*キュー名              */
  long      queuetype;          /*キュー種別            */
  char      status;             /*キュー状態            */
  long      maxjob;             /*最大投入可能ジョブ数 */
  long      maxexec;            /*最大実行可能ジョブ数 */
  long      elapsed;            /*経過時間制限値(秒) */
  long      dfltprty;           /*デフォルト優先順位   */
  long      dfltdprty;          /*デフォルト優先度     */
  long      job_wait;           /*実行待ちジョブ数     */
  long      job_hold;           /*保留中ジョブ数       */
  long      job_run;            /*実行中ジョブ数       */
  char      hostgrp[];          /*分散ホストグループ名 */

} Mp_QUEUEINF_1
```

queuename

キュー名が設定されます。

queuetype

キューの種別が設定されます。設定されるキューの種別を以下に示します。

値	種 別
1	ローカルキュー
2	分散実行キュー

status

キューの状態が設定されます。設定されるキューの状態を以下に示します。

値	状 態
1	稼働中
2	停止中

maxjob

最大投入可能ジョブ数が設定されます。

maxexec

最大実行可能ジョブ数が設定されます。

elapsed

経過時間制限値が秒単位で設定されます。

dfltprty

デフォルト優先順位が設定されます。

dfltdprty

デフォルト優先度が設定されます。

job_wait

APIを呼び出した時点での実行待ちジョブ数が設定されます。

job_hold

APIを呼び出した時点での保留中ジョブ数が設定されます。

job_run

APIを呼び出した時点での実行中ジョブ数が設定されます。

hostgrp

分散ホストグループ名が設定されます。ローカルキューの場合は、NULLが設定されます。

プログラム例

キュー一覧情報獲得APIのプログラム例を以下に示します。

```
#include <windows.h>
#include <stdio.h>
#include "f3cuapi.h"

int main (int argc, char **argv)
{

        Mp_QUEUEINF_1 *queinf = 0;
        Mp_QUEUEINF_1 *qinf = 0;

        long mode;
        long idx;
        long rtn = 0;
        char errmsg[256+1];

        /*
         * initialization working area
         */
        memset(errmsg, 0x00, sizeof(errmsg));

        mode = GETQST;
        /*
         *  call Mp_ListQueue() API
         */
        rtn = Mp_ListQueue (NULL, mode, (void**)&queinf);

        /*
         * check return code of Mp_ListQueue() API
         */
```

```
        if (rtn == -1){
                printf ("GetLastError() is %d.¥n", GetLastError());
                return (1);
        } else if (rtn <= -1000) {
                printf ("Error code is %d.¥n", rtn);
                Mp_GetMJEServer (errmsg);
                printf ("errmsg = %s¥n", errmsg);
                return (1);
        }

        qinf = queinf;

        printf("queue name      wait  hold  run   status¥n");
        for (idx = 0; idx < rtn; idx++) {

                printf("%15.15s %5.5d %5.5d %5.5d ",
                        qinf->queuename, qinf->job_wait, qinf->job_hold, qinf->job_run);
                switch(qinf->status){
                case 1  :
                        printf("active  ");
                        break;
                case 0  :
                default :
                        printf("inactive");
                        break;
                }
                printf("¥n");
                qinf++;

        }

        mode = FREEQST;

        /*
         *  call Mp_ListQueue() API
         */
        Mp_ListQueue (NULL, mode, (void**)&queinf);

        return(0);

}
```

17.3.17　キュー一覧情報獲得API/EE【Windows版】

キュー一覧情報獲得API/EE(Mp_ListQueue_Ex)について説明します。

機能説明

Systemwalker Operation Manager EEで、キュー情報の一覧を獲得します。

記述形式

```
#include "f3cuapi.h"
long Mp_ListQueue_Ex (char *rhost,
                long mode,
                void **queueinf,
                int system_num)
```

パラメタの説明

rhost

NULLを指定してください。

mode

以下に示す処理モードを指定します。GETQSTを行った後、必ずFREEQSTを行ってください。

GETQST:

キュー一覧情報を獲得します。

FREEQST:

GETQSTで獲得したキュー一覧情報領域を解放します。

queueinf

GETQSTの場合:

キュー情報構造体(Mp_QUEUEINF_1)配列のアドレスを通知する領域のアドレスを指定します。

FREEQSTの場合:

GETQSTで返却されたジョブ情報構造体の配列のアドレスを通知した領域のアドレスをそのまま指定してください。また、GETQSTを行った後FREEQSTを行うまでの間、該当領域の内容を変更しないようにしてください。

system_num

操作対象のサブシステムを、0から9の範囲の番号で指定します。

復帰値

復帰値	意 味	対 処
1以上	正常にキュー情報の一覧が獲得されたことを示し、そのキューの数が返却されます。	-
-1	システムエラーが発生しました。	保守情報収集ツールで“ジョブ実行制御”の情報を採取し、富士通技術員に連絡してください。
-1000以下	ジョブ実行制御において、エラーが検出されたため、処理を中断しました。なお、エラーメッセージ獲得APIを使用して、エラーメッセージを獲得することができます。	エラーメッセージ獲得APIを使用して、エラーメッセージを獲得し、そのエラーメッセージに対応した処置を実施してください。

注意事項

引数が無効アドレスを指している場合、当関数は異常終了し、その動作は定義されません。

キュー情報構造体(Mp_QUEUEINF_1)の形式

キュー情報構造体の形式を以下に示します。

当構造体は、INCLUDEファイル“f3cuapi.h”内に定義されます。

```
typedef struct _Mp_QUEUEINF_1 {
  char      queuename[];        /*キュー名             */
  long      queuetype;          /*キュー種別           */
  char      status;             /*キュー状態           */
  long      maxjob;             /*最大投入可能ジョブ数 */
  long      maxexec;            /*最大実行可能ジョブ数 */
```

```
  long      elapsed;            /*経過時間制限値(秒) */
  long      dfltprty;           /*デフォルト優先順位    */
  long      dfltdprty;          /*デフォルト優先度      */
  long      job_wait;           /*実行待ちジョブ数      */
  long      job_hold;           /*保留中ジョブ数        */
  long      job_run;            /*実行中ジョブ数        */
  char      hostgrp[];          /*分散ホストグループ名 */
} Mp_QUEUEINF_1
```

以下に項目ごとの説明を記述します。

queuename

キュー名が設定されます。

queuetype

キューの種別が設定されます。設定されるキューの種別を以下に示します。

値	種別
1	ローカルキュー
2	分散実行キュー

status

キューの状態が設定されます。設定されるキューの状態を以下に示します。

値	状態
1	稼働中
2	停止中

maxjob

最大投入可能ジョブ数が設定されます。

maxexec

最大実行可能ジョブ数が設定されます。

elapsed

経過時間制限値が秒単位で設定されます。

dfltprty

デフォルト優先順位が設定されます。

dfltdprty

デフォルト優先度が設定されます。

job_wait

APIを呼び出した時点での実行待ちジョブ数が設定されます。

job_hold

APIを呼び出した時点での保留中ジョブ数が設定されます。

job_run

APIを呼び出した時点での実行中ジョブ数が設定されます。

hostgrp

分散ホストグループ名が設定されます。ローカルキューの場合は、NULLが設定されます。

プログラム例

キュー一覧情報獲得API/EEのプログラム例を以下に示します。

```
#include <windows.h>
#include <stdio.h>
#include "f3cuapi.h"

int main (int argc, char **argv)
{

        Mp_QUEUEINF_1 *queinf = 0;
        Mp_QUEUEINF_1 *qinf = 0;
        int system_num;

        long mode;
        long idx;
        long rtn = 0;
        char errmsg[256+1];

        /*
         * initialization working area
         */
        memset(errmsg, 0x00, sizeof(errmsg));

        mode = GETQST;
        /*
         *  call Mp_ListQueue_Ex() API
         */
        system_num = 0;
        rtn = Mp_ListQueue_Ex (NULL, mode, (void**)&queinf, system_num);

        /*
         * check return code of Mp_ListQueue_Ex() API
         */
        if (rtn == -1){
                printf ("GetLastError() is %d.¥n", GetLastError());
                return (1);
        } else if (rtn <= -1000) {
                printf ("Error code is %d.¥n", rtn);
                Mp_GetMJESerror (errmsg);
                printf ("errmsg = %s¥n", errmsg);
                return (1);
        }

        qinf = queinf;

        printf("queue name      wait  hold  run   status¥n");
        for (idx = 0; idx < rtn; idx++) {

                printf("%15.15s %5.5d %5.5d %5.5d ",
                        qinf->queuename, qinf->job_wait, qinf->job_hold,
                        qinf->job_run);
                switch(qinf->status){
                case 1  :
                        printf("active  ");
                        break;
                case 0  :
                default :
                        printf("inactive");
                        break;
                }
                printf("¥n");
                qinf++;
```

```
        }

        mode = FREEQST;

        /*
         *  call Mp_ListQueue_Ex() API
         */
        Mp_ListQueue_Ex (NULL, mode, (void**)&queinf, system_num);

        return(0);

}
```

17.3.18　エラーメッセージ獲得API

エラーメッセージ獲得API(Mp_GetMJEServer)について説明します。

機能説明

ジョブ実行制御が提供するAPI使用時において、最近発生したエラー(-1000以下)のエラーメッセージを獲得します。

同一プロセス内に限ります。別プロセスにおいて獲得することはできません。

記述形式

【Windows版】

```
#include "f3cuapi.h"
long Mp_GetMJEServer (char *errormsg)
```

【UNIX版】

```
#include "mjesapi.h"
int Mp_GetMJEServer (char *errormsg)
```

パラメタの説明

errormsg

エラーメッセージを返却する領域のアドレスを指定してください。

当領域は、256バイト用意してください。

復帰値

復帰値	意 味	対 処
0	正常終了しました。	-

注意事項

引数が無効アドレスを指している場合、当関数は異常終了し、その動作は定義されません。

17.4　アクション管理API【Windows版】

本節では、アクションを管理するAPIについて説明します。

アクションを管理するAPIは、アクション実行API、アクション停止API、およびアクション状態通知APIに分類できます。

アクション管理API共通の動作環境、注意事項、および必要ファイルについての 説明を以下に示します。

動作環境

当APIは、Systemwalker Operation Managerサーバ導入マシン上でのみ動作します。Systemwalker Operation Managerクライアントのみが導入されているマシン上では動作しません。

注意事項

- コンパイル環境として、コンパイラは、Visual C++(バージョン2022以降)を、ランタイムライブラリは“マルチスレッド(DLL)”を使用してください。
- コンパイルしたVisual C++バージョン以降のランタイムをAPI実行環境にインストールしてください。
- LIBファイルおよびINCLUDEファイルは、同じバージョン/レベルで提供されたものを使用してください。
- V13.2.0以降のSystemwalker Centric ManagerとSystemwalker Operation Managerが混在した環境および、その混在した環境からV13.2.0以降のSystemwalker Centric Managerのみをアンインストールした環境では、実行に必要な権限が変更されます。詳細については、“Systemwalker Centric Manager APIガイド”を参照してください。

必要ファイル

上記アクション管理APIを使用するには、以下のファイルが必要となります。

Windows 32bit用

LIB:

f3crhxac.lib

INCLUDE:

f3crhxac.h

Windows 64bit用

LIB:

f3crhxac_x64.lib

INCLUDE:

f3crhxac_x64.h

アクション管理の各APIはLIBファイルに格納し、各APIで使用する定数および構造体はINCLUDEファイルに宣言します。

これらのファイルは、Systemwalker Operation Managerをインストールしたディレクトリの“MpWalker¥lib”および“MpWalker¥include”配下に格納されています。

17.4.1 アクション実行API

本項では、“アクション実行API”について説明します。

“アクション実行API”の一覧を以下に示します。

- 音声通知API(Mp_PlaySound2)
- 音声停止API(Mp_StopSound)
- ポップアップメッセージ送信(指定ユーザ)API(Mp_PopupUser2)
- ポップアップメッセージ送信(ドメイン)API(Mp_PopupDomain2)
- ポップアップメッセージ送信(全ユーザ)API(Mp_PopupSession2)
- MS-Mail送信API(Mp_SendMSMail2)
- E-Mail送信API(Mp_SendEMail)
- ショートメールへのメッセージ送信API(Mp_CallPager2)

注意

一度に送信できるポップアップメッセージの最大数は、送信先1台につき25件となります。25件を超えたものは送信されません。ポップアップメッセージのOKボタンを押してポップアップメッセージのダイアログを閉じ、25件を下回った場合に新たなポップアップメッセージの送信が行えるようになります。

17.4.1.1 音声通知API

音声通知API(Mp_PlaySound2)について説明します。

記述形式

```
long Mp_PlaySound2 (unsigned char *AppName,
                    int kind,
                    unsigned char *text_or_filename,
                    int repeat,
                    MpSoundParam *SoundParam
                    char *ExeHost)
```

機能説明

以下の音声を通知します。

- メッセージテキストの読上げ
- WAVファイルの再生
- BEEP音

パラメタの説明

AppName

呼び出し元のアプリケーション名のアドレスを指定します。アプリケーション名は64バイト以内で指定します。

kind

通知する音声の種類を指定します。このパラメタには、次の値のいずれかを指定します。

SOUND_VOICE:

メッセージテキストを読み上げます。

SOUND_WAVE:

WAVファイルを再生します。

SOUND_BEEP:

BEEP音です。

text_or_filename

kindの指定値により、次の値のいずれかを指定します。

SOUND_VOICEを指定した場合:

読み上げるメッセージテキストのアドレスを指定します。

SOUND_WAVEを指定した場合:

再生するファイルのフルパス名のアドレスを260バイト以内で指定します。なお、ファイルはExeHostで指定したホストに存在するものを指定してください。

SOUND_BEEPを指定した場合:

無効です。

repeat

音声を通知する回数を指定します。0を指定した場合は、停止処理がされるまで繰り返します。

SoundParam

音声オプション情報構造体(MpSoundParam)のアドレスを指定します。SOUND_VOICEを指定した場合に、システムの設定値を変更する場合に指定します。SOUND_VOICE以外の場合は、無効になります。設定の必要がない場合は、NULLを指定します。また、各項目で0を指定した場合、システムの設定値が使われます。

ExeHost

アクションを実行するホスト名(アクション実行を選択したクライアント)のアドレスを指定します。NULLを指定した場合は、アクション環境設定画面で指定したホストでアクションが実行されます。

復帰値

正常に要求を受け取った場合は、0以上の整数を返します。0未満の場合はエラーとなります。エラーの場合の詳細を以下に示します。

復帰値	意 味	対 処
MPACT_PARAMERR	パラメタエラーです。	パラメタに誤りがないか確認してください。
MPACT_NOMEMORY	メモリ不足です。	ページファイルのサイズを拡張するか、メモリを増設してください。
MPACT_NOTSOUND	サウンド再生用の環境がインストールされていません。	Microsoft Speech API(SAPI)5.x対応音声合成エンジンを実装された製品がインストールされているか確認してください。
MPACT_NOTMOVE	アクション管理サーバが起動されていません。	サービスSystemwalker MpAosfBが起動されていることを確認してください。
MPACT_COMFAIL	アクション管理サーバとの通信でエラーが発生しました。	サービスSystemwalker MpAosfBおよびSystemwalker MpAosfXが起動されていることを確認してください。
MPACT_SYSERR	アクション管理サーバでエラーが発生しました。	イベントログ/SYSLOGに出力されたメッセージを参照しエラーを取り除いた後、Systemwalkerを再起動してください。エラーを取り除き、Systemwalker を再起動しても再現する場合は、保守情報収集ツールを使用し[イベント監視]の資料を採取して、富士通技術員に連絡してください。
MPACT_NOTDLL	アクション要求用DLLがロードできません。	以下のフォルダにアクション管理に必要なファイルf3crhxac.dll(注)があるか確認してください。 Systemwalkerのインストールディレクトリ¥mpwalker¥bin 存在しない場合は、Systemwalkerを再インストールするか、保守情報収集ツールを使用し[イベント監視]の資料を採取して、富士通技術員に連絡してください。
MPACT_EXCEPT	不当な領域のアドレスが指定されました。	内部論理異常が発生しています。保守情報収集ツールを使用し資料を採取して、富士通技術員に連絡してください。

注）64bitプラットフォームの場合、f3crhxac.dllはf3crhxac_64.dllになります。

音声オプション情報構造体(MpSoundParam)の形式

```
typedef struct MpSoundParam_tag {
    DWORD dwVoiceType ;    // 声の種類
    DWORD dwIntonation ;   // 声のイントネーションのレベル
    DWORD dwPitch ;        // 声の高さ
    DWORD dwSpeed ;        // 読上げのスピード
    DWORD dwSpeedRate  ;   // 読上げスピードの変化の割合
    DWORD dwTone ;         // 高域強調の有無
    DWORD dwVolume ;       // 声のボリューム
} MpSoundParam ;
```

dwVoiceType

声種を指定します。

VOICE_MALE:

男性の声です。

VOICE_FEMALE:

女性の声です。

dwIntonation

イントネーションのレベルを1～4で指定します。

dwPitch

発声の声の高さを1～5で指定します。

dwSpeed

読上げスピードを1～10で指定します。

dwSpeedRate

読上げスピードの変化の割合を1～10で指定します。

dwTone

高域強調の有無を指定します。

なし:

1

あり:

2

DWORD dwVolume

ボリュームを1～10で指定します。

プログラム例

音声通知APIのプログラム例を以下に示します。

```
#include <windows.h>
```

```
#include ”f3crhxac.h”

unsigned char AppName[] = ”音声通知API” ;   // 依頼元アプリケーション名
int kind               = SOUND_VOICE ;     // テキストの読上げ
unsigned char text[]    = ”異常が発生しました。オペレータに連絡してください” ;
int repeat             = 0 ;               // 停止処理がされるまで繰り返す
MpSoundParam  *SndPrm  = NULL ;            // 省略値を使用
long           lrc ;

/* 音声通知依頼 */
lrc = Mp_PlaySound2(AppName,kind,text,repeat,NULL,NULL) ;
if ( lrc < 0 ) {
    /* エラー処理 */
}

/* 正常終了 */
```

17.4.1.2 音声停止API

音声停止API(Mp_StopSound)について説明します。

記述形式

```
long Mp_StopSound (unsigned char *AppName,
                   long handle)
```

機能説明

音声通知を停止します。

パラメタの説明

AppName

音声停止API(Mp_PlaySound)で指定した呼び出し元のアプリケーション名のアドレスを指定します。

handle

Mp_PlaySound関数で返された値を指定します。0を指定した場合は、再生中の音声を停止します。

復帰値

正常に要求を受け取った場合は、0を返します。0未満の場合はエラーとなります。エラーの場合の詳細を以下に示します。

復帰値	意 味	対 処
MPACT_PARAMERR	パラメタエラーです。	パラメタに誤りがないか確認してください。
MPACT_NOMEMORY	メモリ不足です。	ページファイルのサイズを拡張するか、メモリを増設してください。
MPACT_NOTACTION	指定されたアクションは存在(すでに終了している)しません。	-
MPACT_NOTSTOP	指定されたアクションは、停止できません。	-
MPACT_NOTMOVE	アクション管理サーバが起動されていません。	サービスSystemwalker MpAosfBが起動されていることを確認してください。

復帰値	意 味	対 処
MPACT_COMFAIL	アクション管理サーバとの通信でエラーが発生しました。	サービスSystemwalker MpAosfB、およびSystemwalker MpAosfXが起動されていることを確認してください。
MPACT_SYSERR	アクション管理サーバでエラーが発生しました。	イベントログ/SYSLOGに出力されたメッセージを参照しエラーを取り除いた後、Systemwalkerを再起動してください。エラーを取り除き、Systemwalker を再起動しても再現する場合は、保守情報収集ツールを使用し[イベント監視]の資料を採取して、富士通技術員に連絡してください。
MPACT_NOTDLL	アクション要求用DLLがロードできません。	以下のフォルダにアクション管理に必要なファイルf3crhxac.dll(注)があるか確認してください。 Systemwalkerのインストールディレクトリ¥mpwalker¥bin 存在しない場合は、Systemwalkerを再インストールするか、保守情報収集ツールを使用し[イベント監視]の資料を採取して、富士通技術員に連絡してください。
MPACT_EXCEPT	不当な領域のアドレスが指定されました。	内部論理異常が発生しています。保守情報収集ツールを使用し資料を採取して、富士通技術員に連絡してください。

注）64bitプラットフォームの場合、f3crhxac.dllはf3crhxac_64.dllになります。

プログラム例

音声停止APIのプログラム例を以下に示します。

```
#include <windows.h>
#include "f3crhxac.h"

unsigned char AppName[] = "音声通知API" ;       // Mp_PlaySound()に指定したアプリ名
long handle             = 1 ;                   // Mp_PlaySound()の戻り値
long lrc ;

lrc = Mp_StopSound(AppName,handle) ;
if ( lrc != 0 ) {
    /* エラー処理 */
}

/* 正常終了 */
```

17.4.1.3 ポップアップメッセージ送信(指定ユーザ)API

ポップアップメッセージ送信(指定ユーザ)API(Mp_PopupUser2)について説明します。

記述形式

```
long Mp_PopupUser2 (unsigned char *AppName,
                    MpAddress *AddressList,
                    int nAddressList,
                    unsigned char *MsgText,
                    char *ExeHost)
```

機能説明

指定されたユーザがログインしているコンピュータまたは、指定されたコンピュータにポップアップメッセージを表示します。

パラメタの説明

AppName

呼び出し元のアプリケーション名のアドレスを指定します。アプリケーション名は64バイト以内で指定します。

AddressList

宛先情報構造体(MpAddress)のアドレスを指定します。ポップアップメッセージを送信するユーザ名またはコンピュータ名を指定します。複数指定する場合は、配列で指定します。なお、NULLを指定した場合は、ドメインにログインしているすべてのユーザに送信します。

nAddressList

AddressListに指定した配列の数を指定します。

MsgText

ポップアップに表示するメッセージのアドレスを指定します。メッセージは128バイト以内で指定します。

ExeHost

アクションを実行するホスト名(アクション実行を選択したクライアント)のアドレスを指定します。NULLを指定した場合は、アクション環境設定画面で指定したホストでアクションが実行されます。

復帰値

正常に要求を受け取った場合は、0以上の整数を返します。0未満の場合はエラーとなります。エラーの場合の詳細を以下に示します。

復帰値	意 味	対 処
MPACT_PARAMERR	パラメタエラーです。	パラメタに誤りがないか確認してください。
MPACT_NOMEMORY	メモリ不足です。	ページファイルのサイズを拡張するか、メモリを増設してください。
MPACT_NOTADDRESS	有効なユーザ名またはコンピュータ名がありません。	宛先情報構造体（MpAddress）に指定したaddressに誤りがあります。正しい宛先名に変更してください。
MPACT_NOTMOVE	アクション管理サーバが起動されていません。	サービスSystemwalker MpAosfBが起動されていることを確認してください。
MPACT_COMFAIL	アクション管理サーバとの通信に失敗しました。	サービスSystemwalker MpAosfB、およびSystemwalker MpAosfXが起動されていることを確認してください。
MPACT_SYSERR	アクション管理サーバでエラーが発生しました。	イベントログ/SYSLOGに出力されたメッセージを参照しエラーを取り除いた後、Systemwalkerを再起動してください。エラーを取り除き、Systemwalker を再起動しても再現する場合は、保守情報収集ツールを使用し[イベント監視]の資料を採取して、富士通技術員に連絡してください。
MPACT_NOTDLL	アクション要求用DLLがロードできません。	以下のフォルダにアクション管理に必要なファイルf3crhxac.dll(注)があるか確認してく

復帰値	意 味	対 処
		ださい。 Systemwalkerのインストールディレクトリ¥mpwalker¥bin 存在しない場合は、Systemwalkerを再インストールするか、保守情報収集ツールを使用し[イベント監視]の資料を採取して、富士通技術員に連絡してください。
MPACT_EXCEPT	不当な領域のアドレスが指定されました。	内部論理異常が発生しています。保守情報収集ツールを使用し資料を採取して、富士通技術員に連絡してください。

注）64bitプラットフォームの場合、f3crhxac.dllはf3crhxac_64.dllになります。

注意事項

- ポップアップメッセージは、Microsoftネットワークで使用できます。

宛先情報構造体(MpAddress)の形式

```
typedef struct MpAddress_tag {
    int type ;
    unsigned char *address ;
    char *addrid ;
} MpAddress ;
```

type

addressに指定したアドレスの種別を指定します。

TYPE_ADDRESS:

ユーザ名またはコンピュータ名です。

TYPE_NAME:

宛先名です。

address

ユーザ名/コンピュータ名または宛先名のアドレスを指定します。

addrid

Systemwalker Centric Managerがインストールされていて、typeにTYPE_NAMEを指定した場合に有効となります。詳細は、“Systemwalker Centric Manager リファレンスマニュアル”を参照してください。

プログラム例

ポップアップメッセージ送信(指定ユーザ)APIのプログラム例を以下に示します。

```
#include <windows.h>
#include "f3crhxac.h"

unsigned char AppName[] = "ポップアップAPI";   // 依頼元アプリケーション名
MpAddress addr[2] ;                            // 送信先アドレス格納域
int naddr ;
unsigned char text[] = "システムを停止します。";   // 送信するメッセージテキスト
long lrc ;
```

```
addr[0].type = TYPE_ADDRESS ;       // ユーザ operatorとuser01にメッ
addr[0].address = (unsigned char*)"operator" ;      // セージを送信する
addr[1].type = TYPE_ADDRESS ;
addr[1].address = (unsigned char*)"user01" ;
naddr = 2 ;

lrc = Mp_PopupUser2(AppName,addr,naddr,text,NULL) ;
if ( lrc < 0 ) {
    /* エラー処理 */
}

/* 正常終了 */
```

17.4.1.4 ポップアップメッセージ送信(ドメイン)API

ポップアップメッセージ送信(ドメイン)API(Mp_PopupDomain2)について説明します。

記述形式

```
long Mp_PopupDomain2 (unsigned char *AppName,
                      char *domain,
                      unsigned char *MsgText,
                      char *Exehost)
```

機能説明

指定されたドメインに接続しているすべてのコンピュータにポップアップメッセージを表示します。

パラメタの説明

AppName

呼び出し元のアプリケーション名のアドレスを指定します。アプリケーション名は64バイト以内で指定します。

domain

メッセージを表示するドメイン名のアドレスを指定します。ドメイン名は15バイト以内で指定します。

MsgText

ポップアップに表示するメッセージのアドレスを指定します。メッセージは128バイト以内で指定します。

ExeHost

アクションを実行するホスト名(アクション実行を選択したクライアント)のアドレスを指定します。NULLを指定した場合は、アクション環境設定画面で指定したホストでアクションが実行されます。

復帰値

正常に要求を受け取った場合は、0以上の整数を返します。0未満の場合はエラーとなります。エラーの場合の詳細を以下に示します。

復帰値	意 味	対 処
MPACT_PARAMERR	パラメタエラーです。	パラメタに誤りがないか確認してください。
MPACT_NOMEMORY	メモリ不足です。	ページファイルのサイズを拡張するか、メモリを増設してください。
MPACT_NOTMOVE	アクション管理サーバが起動されていません。	サービスSystemwalker MpAosfBが起動されていることを確認してください。

復帰値	意 味	対 処
MPACT_COMFAIL	アクション管理サーバとの通信に失敗しました。	サービスSystemwalker MpAosfB、およびSystemwalker MpAosfXが起動されていることを確認してください。
MPACT_SYSERR	アクション管理サーバでエラーが発生しました。	イベントログ/SYSLOGに出力されたメッセージを参照しエラーを取り除いた後、Systemwalkerを再起動してください。エラーを取り除き、Systemwalker を再起動しても再現する場合は、保守情報収集ツールを使用し[イベント監視]の資料を採取して、富士通技術員に連絡してください。
MPACT_NOTDLL	アクション要求用DLLがロードできません。	以下のフォルダにアクション管理に必要なファイルf3crhxac.dll(注)があるか確認してください。 Systemwalkerのインストールディレクトリ¥mpwalker¥bin 存在しない場合は、Systemwalkerを再インストールするか、保守情報収集ツールを使用し[イベント監視]の資料を採取して、富士通技術員に連絡してください。
MPACT_EXCEPT	不当な領域のアドレスが指定されました。	内部論理異常が発生しています。保守情報収集ツールを使用し資料を採取して、富士通技術員に連絡してください。

注）64bitプラットフォームの場合、f3crhxac.dllはf3crhxac_64.dllになります。

注意事項

- ポップアップメッセージは、Microsoftネットワークで使用できます。

プログラム例

ポップアップメッセージ送信(ドメイン)APIのプログラム例を以下に示します。

```
#include <windows.h>
#include "f3crhxac.h"

unsigned char AppName[] = "ポップアップAPI";  // 依頼元アプリケーション名
char domain[] = "Domain";                     // 送信先のドメイン名
unsigned char text[]    = "システムを停止します。";    // 送信するメッセージテキスト
long lrc ;

lrc = Mp_PopupDomain2(AppName,domain,text,NULL) ;
if ( lrc < 0 ) {
    /* エラー処理 */
}

/* 正常終了 */
```

17.4.1.5　ポップアップメッセージ送信(全ユーザ)API

ポップアップメッセージ送信(全ユーザ)API(Mp_PopupSession2)について説明します。

記述形式

```
long Mp_PopupSession2 (unsigned char *AppName,
                       unsigned char *MsgText,
                       char *ExeHost)
```

機能説明

サーバに接続しているすべてのセッションにポップアップメッセージを送信します。

パラメタの説明

AppName

呼び出し元のアプリケーション名のアドレスを指定します。アプリケーション名は64バイト以内で指定します。

MsgText

ポップアップに表示するメッセージのアドレスを指定します。メッセージは128バイト以内で指定します。

ExeHost

アクションを実行するホスト名(アクション実行を選択したクライアント)のアドレスを指定します。NULLを指定した場合は、アクション環境設定画面で指定したホストでアクションが実行されます。

復帰値

正常に要求を受け取った場合は、0以上の整数を返します。0未満の場合はエラーとなります。エラーの場合の詳細を以下に示します。

復帰値	意 味	対 処
MPACT_PARAMERR	パラメタエラーです。	パラメタに誤りがないか確認してください。
MPACT_NOMEMORY	メモリ不足です。	ページファイルのサイズを拡張するか、メモリを増設してください。
MPACT_NOTMOVE	アクション管理サーバが起動されていません。	サービスSystemwalker MpAosfBが起動されていることを確認してください。
MPACT_COMFAIL	アクション管理サーバとの通信に失敗しました。	サービスSystemwalker MpAosfB、およびSystemwalker MpAosfXが起動されていることを確認してください。
MPACT_SYSERR	アクション管理サーバでエラーが発生しました。	イベントログ/SYSLOGに出力されたメッセージを参照しエラーを取り除いた後、Systemwalkerを再起動してください。エラーを取り除き、Systemwalker を再起動しても再現する場合は、保守情報収集ツールを使用し[イベント監視]の資料を採取して、富士通技術員に連絡してください。
MPACT_NOTDLL	アクション要求用DLLがロードできません。	以下のフォルダにアクション管理に必要なファイルf3crhxac.dll(注)があるか確認してください。 Systemwalkerのインストールディレクトリ¥mpwalker¥bin 存在しない場合は、Systemwalkerを再インストールするか、保守情報収集ツールを使用し[イベント監視]の資料を採取して、富士通技術員に連絡してください。

復帰値	意 味	対 処
MPACT_EXCEPT	不当な領域のアドレスが指定されました。	内部論理異常が発生しています。保守情報収集ツールを使用し資料を採取して、富士通技術員に連絡してください。

注）64bitプラットフォームの場合、f3crhxac.dllはf3crhxac_64.dllになります。

注意事項

- ポップアップメッセージは、Microsoftネットワークで使用できます。

プログラム例

ポップアップメッセージ送信(全ユーザ)APIのプログラム例を以下に示します。

```
#include <windows.h>
#include "f3crhxac.h"

unsigned char AppName[] = "ポップアップAPI";    //  依頼元アプリケーション名
unsigned char text[]    = "システムを停止します。"; //  送信するメッセージテキスト
long lrc ;

lrc = Mp_PopupSession2(AppName,text,NULL) ;
if ( lrc < 0 ) {
    /* エラー処理 */
}

/* 正常終了 */
```

17.4.1.6 MS-Mail送信API

MS-Mail送信API(Mp_SendMSMail2)について説明します。

記述形式

```
long Mp_SendMSMail2 (unsigned char *AppName,
                     MpAddress *AddressList,
                     int nAddressList,
                     MpDataList *MailData,
                     int nMailData,
                     unsigned char *MailTitle,
                     MpMMParam *MmailParam,
                     char *ExeHost)
```

機能説明

MS-Mailの送信をアクション管理サーバに要求します。

パラメタの説明

AppName

呼び出し元のアプリケーション名のアドレスを指定します。アプリケーション名は64バイト以内で指定します。

AddressList

宛先情報構造体(MpAddress)のアドレスを指定します。メールを送信するユーザ名を指定します。複数指定する場合は、配列で指定します。

nAddressList

AddressListに指定した配列数を指定します。

MailData

メール送信データ情報構造体(MpDataList)のアドレスを指定します。複数のデータを送信する場合は、配列で指定します。

nMailData

MailDataに指定した配列数を指定します。

MailTitle

メールのタイトルを指定します。

MMailParam

メール送信管理情報構造体(MpMMParam)のアドレスを指定します。メールを送信するための情報を指定します。NULLの場合、各項目の省略値を使用します。

ExeHost

アクションを実行するホスト名(アクション実行を選択したクライアント)のアドレスを指定します。NULLを指定した場合は、アクション環境設定画面で指定したホストでアクションが実行されます。

復帰値

正常に要求を受け取った場合は、0以上の整数を返します。0未満の場合はエラーとなります。エラーの場合の詳細を以下に示します。

復帰値	意 味	対 処
MPACT_PARAMERR	パラメタエラーです。	パラメタに誤りがないか確認してください。
MPACT_NOMEMORY	メモリ不足です。	ページファイルのサイズを拡張するか、メモリを増設してください。
MPACT_NOTADDRESS	有効なメールアドレスがありません。	宛先情報構造体（MpAddress）に指定したaddressに誤りがあります。正しい宛先名に変更してください。
MPACT_NOTMOVE	アクション管理サーバが起動されていません。	サービスSystemwalker MpAosfBが起動されていることを確認してください。
MPACT_COMFAIL	アクション管理サーバとの通信に失敗しました。	サービスSystemwalker MpAosfB、およびSystemwalker MpAosfXが起動されていることを確認してください。
MPACT_SYSERR	アクション管理サーバでエラーが発生しました。	イベントログ/SYSLOGに出力されたメッセージを参照しエラーを取り除いた後、Systemwalkerを再起動してください。エラーを取り除き、Systemwalker を再起動しても再現する場合は、保守情報収集ツールを使用し[イベント監視]の資料を採取して、富士通技術員に連絡してください。
MPACT_NOTDLL	アクション要求用DLLがロードできません。	以下のフォルダにアクション管理に必要なファイルf3crhxac.dll(注)があるか確認してください。 Systemwalkerのインストールディレクトリ¥mpwalker¥bin

復帰値	意 味	対 処
		存在しない場合は、Systemwalkerを再インストールするか、保守情報収集ツールを使用し[イベント監視]の資料を採取して、富士通技術員に連絡してください。
MPACT_EXCEPT	不当な領域のアドレスが指定されました。	内部論理異常が発生しています。保守情報収集ツールを使用し資料を採取して、富士通技術員に連絡してください。

注）64bitプラットフォームの場合、f3crhxac.dllはf3crhxac_64.dllになります。

備考

- メールデータにメモリデータは1つのみ指定可能。また、メモリデータは配列の先頭に指定します。メモリデータがない場合は、すべて添付ファイルとして送信します。
- メモリデータは、'¥0'で終了させます。改行は、CR(0x0d)+LF(0x0a)のコードで行います。

注意事項

Outlook Expressがインストールされている場合は、MS-Mailの受信はできません。E-Mailで実施してください。

宛先情報構造体(MpAddress)の形式

```
typedef struct MpAddress_tag {
    int type ;
    unsigned char *address ;
    char *addrid ;
} MpAddress ;
```

type

addressに指定したアドレスの種別を指定します。

TYPE_ADDRESS:

送信先のメールアドレスです。

TYPE_NAME:

宛先名です。

address

メールアドレスまたは宛先名のアドレスを指定します。

addrid

Systemwalker Centric Managerがインストールされていて、typeにTYPE_NAMEを指定した場合に有効となります。詳細は、“Systemwalker Centric Manager リファレンスマニュアル”を参照してください。

メール送信データ情報構造体(MpDataList)の形式

```
typedef struct MpDataList_tag {
    int type ;
    unsigned char *data ;
} MpDataList ;
```

type

dataに指定するデータの種別を指定します。

TYPE_LMEMORY:

メモリデータです。

TYPE_FILENAME:

ファイル名です。

TYPE_TEMPFILE:

ファイル名(メール送信が正常終了した後、アクション管理がファイルを削除する)です。

data

TYPE_LMEMORYの場合は、1028バイト以内で指定したメモリデータのアドレスを指定します。

TYPE_FILENAMEおよびTYPE_TEMPFILEの場合は、MAX_PATH以内で指定したファイル名のアドレスを指定します。

メール送信管理情報構造体(MpMMParam)の形式

```
typedef struct MpMMParam_tag {
    char *Profile ;
    char *Password ;
} MpMMParam ;
```

Profile

ログインするユーザ名のアドレスを指定します。NULLを指定した場合は、省略値を使用します。

Password

ログインするユーザのパスワードのアドレスを指定します。NULLを指定した場合は、省略値を使用します。

プログラム例

MS-Mail送信APIのプログラム例を以下に示します。

```
#include <windows.h>
#include "f3crhxac.h"

unsigned char AppName[] = "MS-Mail送信API" ;  // 依頼元アプリケーション名
MpAddress addr[2] ;                          // 送信先アドレス格納域
int naddr ;
MpDataList data[2] ;
int ndata ;
unsigned char Title[]   = "システム異常通知" ;  // メールタイトル
MpMMParam*  MailPrm = NULL ;                    // 省略値を使用
long lrc ;

addr[0].type = TYPE_ADDRESS ;               // ユーザ operatorとuser01
addr[0].address = (unsigned char*)"operator" ;              // にメッセージを送信する
addr[1].type = TYPE_ADDRESS ;
addr[1].address = (unsigned char*)"user01" ;
naddr = 2 ;

data[0].type = TYPE_LMEMORY ;                   // 送信データの設定
data[0].data = (unsigned char*)"異常が発生しました。¥r¥n詳細は添付ファイルを参照してください。¥r¥n" ;
data[1].type = TYPE_FILENAME ;
data[1].data = (unsigned char*)"D:¥¥TEMP¥¥ERR001.LOG" ;
ndata = 2 ;
```

```
lrc = Mp_SendMSMail2(AppName,addr,naddr,data,ndata,Title,MailPrm,NULL) ;
if ( lrc < 0 ) {
    /* エラー処理 */
}

/* 正常終了 */
```

17.4.1.7 E-Mail送信API

E-Mail送信API(Mp_SendEMail)について説明します。

記述形式

```
long Mp_SendEMail (unsigned char *AppName,
                   MpAddress *AddressList,
                   int nAddressList,
                   MpDataList *MailData,
                   int nMailData,
                   char *FromAddress,
                   unsigned char *MailTitle,
                   char *SMTPServer)
```

機能説明

E-Mailの送信をアクション管理サーバに要求します。

パラメタの説明

AppName

呼び出し元のアプリケーション名のアドレスを指定します。アプリケーション名は64バイト以内で指定します。

AddressList

宛先情報構造体(MpAddress)のアドレスを指定します。複数指定する場合は、配列で指定します。

宛先情報構造体については、“17.4.1.6 MS-Mail送信API”を参照してください。

nAddressList

AddressListに指定したアドレスの数(配列数)を指定します。

MailData

メール送信データ情報構造体(MpDataList)のアドレスを指定します。複数のデータを送信する場合は、配列で指定します。

メール送信データ情報構造体については、“17.4.1.6 MS-Mail送信API”を参照してください。

nMailData

MailDataに指定した配列数を指定します。

FromAddress

メール送信元のメールアドレスを指定します。NULLを指定した場合は、省略値を使用します。

MailTitle

メールのタイトルを指定します。

SMTPServer

SMTPサーバ名を指定します。NULLを指定した場合は、省略値を使用します。

復帰値

正常に要求を受け取った場合は、0以上の整数を返します。0未満の場合はエラーとなります。エラーの場合の詳細を以下に示します。

復帰値	意 味	対 処
MPACT_PARAMERR	パラメタエラーです。	パラメタに誤りがないか確認してください。
MPACT_NOMEMORY	メモリ不足です。	ページファイルのサイズを拡張するか、メモリを増設してください。
MPACT_NOTADDRESS	有効なメールアドレスがありません。	宛先情報構造体（MpAddress）に指定したaddressに誤りがあります。正しい宛先名に変更してください。
MPACT_NOTMOVE	アクション管理サーバが起動されていません。	サービスSystemwalker MpAosfBが起動されていることを確認してください。
MPACT_COMFAIL	アクション管理サーバとの通信に失敗しました。	サービスSystemwalker MpAosfB、およびSystemwalker MpAosfXが起動されていることを確認してください。
MPACT_SYSERR	アクション管理サーバでエラーが発生しました。	イベントログ/SYSLOGに出力されたメッセージを参照しエラーを取り除いた後、Systemwalkerを再起動してください。エラーを取り除き、Systemwalker を再起動しても再現する場合は、保守情報収集ツールを使用し[イベント監視]の資料を採取して、富士通技術員に連絡してください。
MPACT_NOTDLL	アクション要求用DLLがロードできません。	以下のフォルダにアクション管理に必要なファイルf3crhxac.dll(注)があるか確認してください。 Systemwalkerのインストールディレクトリ¥mpwalker¥bin 存在しない場合は、Systemwalkerを再インストールするか、保守情報収集ツールを使用し[イベント監視]の資料を採取して、富士通技術員に連絡してください。
MPACT_EXCEPT	不当な領域のアドレスが指定されました。	内部論理異常が発生しています。保守情報収集ツールを使用し資料を採取して、富士通技術員に連絡してください。

注）64bitプラットフォームの場合、f3crhxac.dllはf3crhxac_64.dllになります。

備考

- メールデータの先頭に指定されたデータは、メールの本体(テキスト)として送信します。メールの本体部がない場合は、先頭の配列のメンバdataにNULLを指定します。
- メールデータにTYPE_LMEMORYは1つのみ指定できます。また、TYPE_LMEMORYは配列の先頭に指定する必要があります。
- メモリデータは、'¥0'で終了させます。改行は、CR(0x0d)+LF(0x0a)のコードで行います。
- メール本体にSJISコードが指定されている場合は、テキストはISO-2022-JP形式に変換後、送信します。添付ファイルは、Base64の形式に変換して送信します。

プログラム例

E-Mail送信APIのプログラム例を以下に示します。

```
#include <windows.h>
#include "f3crhxac.h"

unsigned char AppName[] = "E-Mail送信API" ; // 依頼元アプリケーション名
MpAddress addr[2] ;                         // 送信先アドレス格納域
int naddr ;
MpDataList data[2] ;
int ndata ;
char FromAddr[]         = "MpWalker@fujitsu.co.jp" ; // 送信元アドレス
unsigned char Title[]           = "System Report" ;         // メールタイトル
char *SMTPSrv          = NULL ;              // 省略値を使用
long lrc ;

addr[0].type = TYPE_ADDRESS ;         // ユーザ operatorとuser01にメッ
addr[0].address = (unsigned char*)"operator@fujitsu.co.jp" ;  // セージを送信する
addr[1].type = TYPE_ADDRESS ;
addr[1].address = (unsigned char*)"user01@fujitsu.co.jp" ;
naddr = 2 ;

data[0].type = TYPE_LMEMORY ;           // 送信データの設定
data[0].data = (unsigned char*)"異常が発生しました。¥r¥n詳細は添付ファイルを参照してください。¥r¥n" ;
data[1].type = TYPE_FILENAME ;
data[1].data = (unsigned char*)"D:¥¥TEMP¥¥ERR001.LOG" ;
ndata = 2 ;

lrc = Mp_SendEMail(AppName,addr,naddr,data,ndata,FromAddr,Title,SMTPSrv) ;
if ( lrc < 0 ) {
  /* エラー処理 */
}

/* 正常終了 */
```

17.4.1.8　ショートメールへのメッセージ送信API

ショートメールへのメッセージ送信API(Mp_CallPager2)について説明します。

記述形式

```
long Mp_CallPager2 (unsigned char *AppName,
                    MpPagerList *AddressList,
                    int nAddressList,
                    char *MsgText,
                    int msgtype,
                    char *exehost)
```

機能説明

ショートメールへのメッセージ送信を行います。

パラメタの説明

AppName

呼び出し元のアプリケーション名のアドレスを指定します。アプリケーション名は64バイト以内で指定します。

AddressList

ショートメール送信情報構造体(MpPagerList)のアドレスを指定します。ショートメールの電話番号を指定します。複数ある場合は配列で指定します。

nAddressList

AddressListに指定したアドレスの数(配列数)を指定します。

MsgText

送信するメッセージのアドレスを指定します。メッセージは、送信するコード(定型メッセージのコードも含む)を指定します。

または、ショートメールの会社でサポートしている文字を半角文字で指定します。

MsgType

MsgTextに指定したメッセージの形式を指定します。

PAGER_CODE:

番号形式です。

PAGER_TEXT:

メッセージテキストです。

exehost

アクションを実行するホスト名(アクション実行を選択したクライアント)のアドレスを指定します。NULLを指定した場合は、アクション環境設定画面で指定したホストでアクションが実行されます。

復帰値

正常に要求を受け取った場合は、0以上の整数を返します。0未満の場合はエラーとなります。エラーの場合の詳細を以下に示します。

復帰値	意 味	対 処
MPACT_PARAMERR	パラメタエラーです。	パラメタに誤りがないか確認してください。
MPACT_NOMEMORY	メモリ不足です。	ページファイルのサイズを拡張するか、メモリを増設してください。
MPACT_NOTADDRESS	有効なメールアドレスがありません。	宛先情報構造体（MpAddress）に指定したaddressに誤りがあります。正しい宛先名に変更してください。
MPACT_NOTMODEM	モデムがインストールされていません。	モデムのインストール状態、および正しく接続されているかを確認してください。モデムに関する確認は、OSのヘルプに記載されている内容を参考に実施してください。
MPACT_NOTMOVE	アクション管理サーバが起動されていません。	サービスSystemwalker MpAosfBが起動されていることを確認してください。
MPACT_COMFAIL	アクション管理サーバとの通信に失敗しました。	サービスSystemwalker MpAosfB、およびSystemwalker MpAosfXが起動されていることを確認してください。
MPACT_SYSERR	アクション管理サーバでエラーが発生しました。	イベントログ/SYSLOGに出力されたメッセージを参照しエラーを取り除いた後、Systemwalkerを再起動してください。エラーを取り除き、Systemwalker を再起動しても再現する場合は、保守情報収集ツールを使用

復帰値	意 味	対 処
		し[イベント監視]の資料を採取して、富士通技術員に連絡してください。
MPACT_NOTDLL	アクション要求用DLLがロードできません。	以下のフォルダにアクション管理に必要なファイルf3crhxac.dll(注)があるか確認してください。 Systemwalkerのインストールディレクトリ¥mpwalker¥bin 存在しない場合は、Systemwalkerを再インストールするか、保守情報収集ツールを使用し[イベント監視]の資料を採取して、富士通技術員に連絡してください。
MPACT_EXCEPT	不当な領域のアドレスが指定されました。	内部論理異常が発生しています。保守情報収集ツールを使用し資料を採取して、富士通技術員に連絡してください。

注）64bitプラットフォームの場合、f3crhxac.dllはf3crhxac_64.dllになります。

備考

送信できるメッセージの長さは、ショートメールおよびモデムの種類により異なります。NTTドコモ 携帯電話およびモデムの説明書を参照してください。なお、送信できるメッセージの長さを超えた場合は、正しく送信できないことがあります。

ショートメール送信情報構造体(MpPagerList)の形式

```
typedef struct MpPagerList_tag {
    int type ;
    unsigned char *address ;
    char *addrid ;
    char *vender ;
} MpPagerList ;
```

type

addressに指定したアドレスの種別を指定します。

TYPE_ADDRESS:

ショートメールの電話番号です。

TYPE_NAME:

宛先名です。

address

電話番号または宛先名のアドレスを指定します。

電話番号の場合、ショートメールの電話番号を以下の形式で指定します。

“ショートメールセンターの電話番号”＋“T”＋“通知先の電話番号”

ショートメールセンターの電話番号

NTTドコモ ショートメールセンターの電話番号を設定します。

参考) NTTドコモ ショートメールセンターの電話番号は、2007年9月現在、0903101655です。詳細は、契約会社に確認してください。

T

区切り文字です。“T”または“t”を設定してください。

通知先の電話番号

通知先の電話番号を設定してください。

例) 通知先 09012345678 にショートメールを送信する場合の設定例

```
0903101655T09012345678
```

addrid

Systemwalker Centric Managerがインストールされていて、typeにTYPE_NAMEを指定した場合に有効となります。詳細は、“Systemwalker Centric Manager リファレンスマニュアル”を参照してください。

vender

ショートメールの種類を指定します。Systemwalkerがサポートしているショートメールの種類を以下に示します。他の種類を指定する場合は、[アクション環境設定]ダイアログボックスで登録したショートメールの会社名を指定し、メッセージを変換する出口(サポート外のショートメールのメッセージ変換用出口)を作成する必要があります。

なお、サポート外のショートメールのメッセージ変換用出口については、“19.9 サポート外のショートメールのメッセージ変換用出口【Windows版】”を参照してください。

MPPAGER_DOCOMO:

NTT DoCoMo ショートメール

プログラム例

ショートメールへのメッセージ送信APIのプログラム例を以下に示します。

```
#include <windows.h>
#include "f3crhxac.h"

unsigned char AppName[] = "ショートメールAPI" ;  // 依頼元アプリケーション名
MpPagerList addr[2] ;                          // 送信先アドレス格納域
int naddr ;
char text[]             = "イジョウハッセイ" ;  // 送信するメッセージ
int type               = PAGER_TEXT ;
long lrc ;


addr[0].type = TYPE_ADDRESS ;           // 2箇所にショートメール
addr[0].address = (unsigned char*)"0903101655T1112223333" ;     // を送信する
addr[0].vender = (char*)MPPAGER_DOCOMO ;
addr[1].type = TYPE_ADDRESS ;
addr[1].address = (unsigned char*)"0903101655T1114445555" ;
addr[1].vender = (char*)MPPAGER_DOCOMO ;
naddr = 2;

lrc = Mp_CallPager2(AppName,addr,naddr,text,type,NULL) ;
if ( lrc < 0 ) {
    /* エラー処理 */
}

/* 正常終了 */
```

17.4.2　アクション実行API(旧バージョン互換)

本項では、旧バージョンのSystemwalker Operation Managerで提供された“アクション実行API”について説明します。

“アクション実行API”の一覧を以下に示します。

- 音声通知API(Mp_PlaySound)
- ポップアップメッセージ送信(指定ユーザ)API(Mp_PopupUser)
- ポップアップメッセージ送信(ドメイン)API(Mp_PopupDomain)
- ポップアップメッセージ送信(全ユーザ)API(Mp_PopupSession)

旧バージョンのSystemwalker Operation Managerとは、ここでは以下のバージョンを指しています。

- SystemWalker/OperationMGR V4.0

17.4.2.1　音声通知API

音声通知API(Mp_PlaySound)について説明します。

記述形式

```
long Mp_PlaySound (unsigned char *AppName,
                   int kind,
                   unsigned char *text_or_filename,
                   int repeat,
                   MpSoundParam *SoundParam)
```

機能説明

以下の音声を通知します。

- メッセージテキストの読上げ
- WAVファイルの再生
- BEEP音

パラメタの説明

AppName

呼び出し元のアプリケーション名のアドレスを指定します。アプリケーション名は64バイト以内で指定します。

kind

通知する音声の種類を指定します。このパラメタには、次の値のいずれかを指定します。

SOUND_VOICE:

メッセージテキストを読み上げます。

SOUND_WAVE:

WAVファイルを再生します。

SOUND_BEEP:

BEEP音です。

text_or_filename

kindの指定値により、次の値のいずれかを指定します。

SOUND_VOICEを指定した場合:

読み上げるメッセージテキストのアドレスを指定します。

SOUND_WAVEを指定した場合:

再生するファイルのフルパス名のアドレスをMAX_PATH以内で指定します。

SOUND_BEEPを指定した場合:

無効です。

repeat

音声を通知する回数を指定します。0を指定した場合は、停止処理がされるまで繰り返します。

SoundParam

音声オプション情報構造体(MpSoundParam)のアドレスを指定します。SOUND_VOICEを指定した場合に、システムの設定値を変更する場合に指定します。SOUND_VOICE以外の場合は、無効になります。設定の必要がない場合は、NULLを指定します。また、各項目で0を指定した場合、システムの設定値が使われます。

復帰値

正常に要求を受け取った場合は、0以上の整数を返します。0未満の場合はエラーとなります。エラーの場合の詳細を以下に示します。

復帰値	意 味	対 処
MPACT_PARAMERR	パラメタエラーです。	パラメタに誤りがないか確認してください。
MPACT_NOMEMORY	メモリ不足です。	ページファイルのサイズを拡張するか、メモリを増設してください。
MPACT_NOTSOUND	サウンド再生用の環境がインストールされていません。	Microsoft Speech API(SAPI)5.x対応音声合成エンジンを実装された製品がインストールされているか確認してください。
MPACT_NOTMOVE	アクション管理サーバが起動されていません。	サービスSystemwalker MpAosfBが起動されていることを確認してください。
MPACT_COMFAIL	アクション管理サーバとの通信に失敗しました。	サービスSystemwalker MpAosfB、およびSystemwalker MpAosfXが起動されていることを確認してください。
MPACT_SYSERR	アクション管理サーバでエラーが発生しました。	イベントログ/SYSLOGに出力されたメッセージを参照しエラーを取り除いた後、Systemwalkerを再起動してください。エラーを取り除き、Systemwalker を再起動しても再現する場合は、保守情報収集ツールを使用し[イベント監視]の資料を採取して、富士通技術員に連絡してください。
MPACT_NOTDLL	アクション要求用DLLがロードできません。	以下のフォルダにアクション管理に必要なファイルf3crhxac.dllがあるか確認してください。 Systemwalkerのインストールディレクトリ¥mpwalker¥bin 存在しない場合は、Systemwalkerを再インストールするか、保守情報収集ツールを使用し[イベント監視]の資料を採取して、富士通技術員に連絡してください。

復帰値	意味	対処
MPACT_EXCEPT	不当な領域のアドレスが指定されました。	内部論理異常が発生しています。保守情報収集ツールを使用し資料を採取して、富士通技術員に連絡してください。

音声オプション情報構造体(MpSoundParam)の形式

```
typedef struct MpSoundParam_tag {
    DWORD dwVoiceType ;
    DWORD dwIntonation ;
    DWORD dwPitch ;
    DWORD dwSpeed ;
    DWORD dwSpeedRate ;
    DWORD dwTone ;
    DWORD dwVolume ;
} MpSoundParam ;
```

dwVoiceType

声種を指定します。

VOICE_MALE:

男性の声です。

VOICE_FEMALE:

女性の声です。

dwIntonation

イントネーションのレベルを1～4で指定します。

dwPitch

発声の声の高さを1～5で指定します。

dwSpeed

読上げスピードを1～10で指定します。

dwSpeedRate

読上げスピードの変化の割合を1～10で指定します。

dwTone

高域強調の有無を指定します。

なし:

1

あり:

2

DWORD dwVolume

ボリュームを1～10で指定します。

プログラム例

音声通知APIのプログラム例を以下に示します。

```
#include <windows.h>
#include "f3crhxac.h"

unsigned char AppName[] = "音声通知API" ;  // 依頼元アプリケーション名
int kind               = SOUND_VOICE ;    // テキストの読上げ
unsigned char text[]    = "異常が発生しました。オペレータに連絡してください" ;
int repeat             = 0 ;           // 停止処理がされるまで繰り返す
MpSoundParam  *SndPrm  = NULL ;        // 省略値を使用
long           lrc ;

/* 音声通知依頼 */
lrc = Mp_PlaySound(AppName,kind,text,repeat,NULL) ;
if ( lrc < 0 ) {
    /* エラー処理 */
}

/* 正常終了 */
```

17.4.2.2　ポップアップメッセージ送信(指定ユーザ)API

ポップアップメッセージ送信(指定ユーザ)API(Mp_PopupUser)について説明します。

記述形式

```
long Mp_PopupUser (unsigned char *AppName,
                   MpAddress *AddressList,
                   int nAddressList,
                   unsigned char *MsgText)
```

機能説明

指定されたユーザがログインしているコンピュータまたは、指定されたコンピュータにポップアップメッセージを表示します。

パラメタの説明

AppName

呼び出し元のアプリケーション名のアドレスを指定します。アプリケーション名は64バイト以内で指定します。

AddressList

宛先情報構造体(MpAddress)のアドレスを指定します。ポップアップメッセージを送信するユーザ名またはコンピュータ名を指定します。複数指定する場合は、配列で指定します。なお、NULLを指定した場合は、ドメインにログインしているすべてのユーザに送信します。

nAddressList

AddressListに指定した配列の数を指定します。

MsgText

ポップアップに表示するメッセージのアドレスを指定します。メッセージは128バイト以内で指定します。

復帰値

正常に要求を受け取った場合は、0以上の整数を返します。0未満の場合はエラーとなります。エラーの場合の詳細を以下に示します。

復帰値	意味	対処
MPACT_PARAMERR	パラメタエラーです。	パラメタに誤りがないか確認してください。
MPACT_NOMEMORY	メモリ不足です。	ページファイルのサイズを拡張するか、メモリを増設してください。
MPACT_NOTADDRESS	有効なユーザ名またはコンピュータ名がありません。	宛先情報構造体（MpAddress）に指定したaddressに誤りがあります。正しい宛先名に変更してください。
MPACT_NOTMOVE	アクション管理サーバが起動されていません。	サービスSystemwalker MpAosfBが起動されていることを確認してください。
MPACT_COMFAIL	アクション管理サーバとの通信に失敗しました。	サービスSystemwalker MpAosfB、およびSystemwalker MpAosfXが起動されていることを確認してください。
MPACT_SYSERR	アクション管理サーバでエラーが発生しました。	イベントログ/SYSLOGに出力されたメッセージを参照しエラーを取り除いた後、Systemwalkerを再起動してください。エラーを取り除き、Systemwalker を再起動しても再現する場合は、保守情報収集ツールを使用し[イベント監視]の資料を採取して、富士通技術員に連絡してください。
MPACT_NOTDLL	アクション要求用DLLがロードできません。	以下のフォルダにアクション管理に必要なファイルf3crhxac.dllがあるか確認してください。 Systemwalkerのインストールディレクトリ¥mpwalker¥bin 存在しない場合は、Systemwalkerを再インストールするか、保守情報収集ツールを使用し[イベント監視]の資料を採取して、富士通技術員に連絡してください。
MPACT_EXCEPT	不当な領域のアドレスが指定されました。	内部論理異常が発生しています。保守情報収集ツールを使用し資料を採取して、富士通技術員に連絡してください。

注意事項

ポップアップメッセージは、Microsoftネットワークでのみ使用できます。

宛先情報構造体(MpAddress)の形式

```
typedef struct MpAddress_tag {
    int type ;
    unsigned char *address ;
    char *addrid ;
} MpAddress ;
```

type

addressに指定したアドレスの種別を指定します。

TYPE_ADDRESS:

ユーザ名またはコンピュータ名です。

TYPE_NAME:

宛先名です。

address

ユーザ名/コンピュータ名または宛先名のアドレスを指定します。

addrid

Systemwalker Centric Managerがインストールされていて、typeにTYPE_NAMEを指定した場合に有効となります。詳細は、“Systemwalker Centric Manager リファレンスマニュアル”を参照してください。

プログラム例

ポップアップメッセージ送信(指定ユーザ)APIのプログラム例を以下に示します。

```
#include ”f3crhxac.h”

unsigned char AppName[] = ”ポップアップAPI” ;  // 依頼元アプリケーション名
MpAddress addr[2] ;                 // 送信先アドレス格納域
int naddr ;
unsigned char text[]    = ”システムを停止します。” ; // 送信するメッセ
                                                  //ージテキスト
long lrc ;

addr[0].type = TYPE_ADDRESS ;       // ユーザ operatorとuser01にメッ
addr[0].address = (unsigned char*)”operator” ;      // セージを送信する
addr[1].type = TYPE_ADDRESS ;
addr[1].address = (unsigned char*)”user01” ;
naddr = 2 ;

lrc = Mp_PopupUser(AppName,addr,naddr,text) ;
if ( lrc < 0 ) {
    /* エラー処理 */
}

/* 正常終了 */
```

17.4.2.3　ポップアップメッセージ送信(ドメイン)API

ポップアップメッセージ送信(ドメイン)API(Mp_PopupDomain)について説明します。

記述形式

```
long Mp_PopupDomain (unsigned char *AppName,
                char *domain,
                unsigned char *MsgText)
```

機能説明

指定されたドメインに接続しているすべてのコンピュータにポップアップメッセージを表示します。

パラメタの説明

AppName

呼び出し元のアプリケーション名のアドレスを指定します。アプリケーション名は64バイト以内で指定します。

domain

メッセージを表示するドメイン名のアドレスを指定します。ドメイン名は15バイト以内で指定します。

MsgText

ポップアップに表示するメッセージのアドレスを指定します。メッセージは128バイト以内で指定します。

復帰値

正常に要求を受け取った場合は、0以上の整数を返します。0未満の場合はエラーとなります。エラーの場合の詳細を以下に示します。

復帰値	意味	対処
MPACT_PARAMERR	パラメタエラーです。	パラメタに誤りがないか確認してください。
MPACT_NOMEMORY	メモリ不足です。	ページファイルのサイズを拡張するか、メモリを増設してください。
MPACT_NOTMOVE	アクション管理サーバが起動されていません。	サービスSystemwalker MpAosfBが起動されていることを確認してください。
MPACT_COMFAIL	アクション管理サーバとの通信に失敗しました。	サービスSystemwalker MpAosfB、およびSystemwalker MpAosfXが起動されていることを確認してください。
MPACT_SYSERR	アクション管理サーバでエラーが発生しました。	イベントログ/SYSLOGに出力されたメッセージを参照しエラーを取り除いた後、Systemwalkerを再起動してください。エラーを取り除き、Systemwalker を再起動しても再現する場合は、保守情報収集ツールを使用し[イベント監視]の資料を採取して、富士通技術員に連絡してください。
MPACT_NOTDLL	アクション要求用DLLがロードできません。	以下のフォルダにアクション管理に必要なファイルf3crhxac.dllがあるか確認してください。 Systemwalkerのインストールディレクトリ¥mpwalker¥bin 存在しない場合は、Systemwalkerを再インストールするか、保守情報収集ツールを使用し[イベント監視]の資料を採取して、富士通技術員に連絡してください。
MPACT_EXCEPT	不当な領域のアドレスが指定されました。	内部論理異常が発生しています。保守情報収集ツールを使用し資料を採取して、富士通技術員に連絡してください。

注意事項

ポップアップメッセージは、Microsoftネットワークでのみ使用できます。

プログラム例

ポップアップメッセージ送信(ドメイン)APIのプログラム例を以下に示します。

```
#include "f3crhxac.h"

unsigned char AppName[] = "ポップアップAPI";   // 依頼元アプリケーション名
char domain[]           = "Domain";            // 送信先のドメイン名
unsigned char text[]    = "システムを停止します。"; // 送信するメッセージテキスト
long lrc ;

lrc = Mp_PopupDomain(AppName,domain,text) ;
```

```
if ( lrc < 0 ) {
    /* エラー処理 */
}

/* 正常終了 */
```

17.4.2.4　ポップアップメッセージ送信(全ユーザ)API

ポップアップメッセージ送信(全ユーザ)API(Mp_PopupSession)について説明します。

記述形式

```
long Mp_PopupSession (unsigned char *AppName,
                      unsigned char *MsgText)
```

機能説明

サーバに接続しているすべてのセッションにポップアップメッセージを送信します。

パラメタの説明

AppName

呼び出し元のアプリケーション名のアドレスを指定します。アプリケーション名は64バイト以内で指定します。

MsgText

ポップアップに表示するメッセージのアドレスを指定します。メッセージは128バイト以内で指定します。

復帰値

正常に要求を受け取った場合は、0以上の整数を返します。0未満の場合はエラーとなります。エラーの場合の詳細を以下に示します。

復帰値	意 味	対 処
MPACT_PARAMERR	パラメタエラーです。	パラメタに誤りがないか確認してください。
MPACT_NOMEMORY	メモリ不足です。	ページファイルのサイズを拡張するか、メモリを増設してください。
MPACT_NOTMOVE	アクション管理サーバが起動されていません。	サービスSystemwalker MpAosfBが起動されていることを確認してください。
MPACT_COMFAIL	アクション管理サーバとの通信に失敗しました。	サービスSystemwalker MpAosfB、およびSystemwalker MpAosfXが起動されていることを確認してください。
MPACT_SYSERR	アクション管理サーバでエラーが発生しました。	イベントログ/SYSLOGに出力されたメッセージを参照しエラーを取り除いた後、Systemwalkerを再起動してください。エラーを取り除き、Systemwalker を再起動しても再現する場合は、保守情報収集ツールを使用し[イベント監視]の資料を採取して、富士通技術員に連絡してください。
MPACT_NOTDLL	アクション要求用DLLがロードできません。	以下のフォルダにアクション管理に必要なファイルf3crhxac.dllがあるか確認してください。 Systemwalkerのインストールディレクトリ¥mpwalker¥bin 存在しない場合は、Systemwalkerを再インス

復帰値	意 味	対 処
		トールするか、保守情報収集ツールを使用し[イベント監視]の資料を採取して、富士通技術員に連絡してください。
MPACT_EXCEPT	不当な領域のアドレスが指定されました。	内部論理異常が発生しています。保守情報収集ツールを使用し資料を採取して、富士通技術員に連絡してください。

注意事項

ポップアップメッセージは、Microsoftネットワークでのみ使用できます。

プログラム例

ポップアップメッセージ送信(全ユーザ)APIのプログラム例を以下に示します。

```
#include ”f3crhxac.h”

unsigned char AppName[] = ”ポップアップAPI”;        // 依頼元アプリケーション名
unsigned char text[]    = ”システムを停止します。”; // 送信するメッセージテキスト
long lrc ;

lrc = Mp_PopupSession(AppName,text) ;
if ( lrc < 0 ) {
    /* エラー処理 */
}

/* 正常終了 */
```

17.4.3 アクション停止API

本項では、“アクション停止API”について説明します。

“アクション停止API”の一覧を以下に示します。

・ アクション停止API(Mp_StopAction)

17.4.3.1 アクション停止API

アクションの停止API(Mp_StopAction)について説明します。

記述形式

```
long Mp_StopAction (unsigned char *AppName,
                    long handle)
```

機能説明

アクション管理サーバに依頼したアクションを停止します。

パラメタの説明

AppName

停止するアクションを要求した(アクション実行用のAPIに指定した)呼び出し元のアプリケーション名のアドレスを指定します。

handle

停止するアクションの管理番号(アクション実行用のAPIの復帰値)を指定します。

復帰値

復帰値	意味	対処
0	停止処理を行います。	-
MPACT_NOTACTION	指定されたアクションは存在しません(すでに終了している)。	-
MPACT_NOTSTOP	実行中のため、停止できません。	すでに実行中のため停止できません。
MPACT_NOTACCESS	停止する権限がありません	Administrator権限で実行してください。
MPACT_PARAMERR	パラメタエラーです。	パラメタに誤りがないか確認してください。
MPACT_NOMEMORY	メモリ不足です。	ページファイルのサイズを拡張するか、メモリを増設してください。
MPACT_NOTMOVE	アクション管理サーバが起動されていません。	サービスSystemwalker MpAosfBが起動されていることを確認してください。
MPACT_COMFAIL	アクション管理サーバとの通信に失敗しました。	サービスSystemwalker MpAosfB、およびSystemwalker MpAosfXが起動されていることを確認してください。
MPACT_SYSERR	アクション管理サーバでエラーが発生しました。	イベントログ/SYSLOGに出力されたメッセージを参照しエラーを取り除いた後、Systemwalkerを再起動してください。エラーを取り除き、Systemwalker を再起動しても再現する場合は、保守情報収集ツールを使用し[イベント監視]の資料を採取して、富士通技術員に連絡してください。
MPACT_NOTDLL	アクション要求用DLLがロードできません。	以下のフォルダにアクション管理に必要なファイルf3crhxac.dllがあるか確認してください。 Systemwalkerのインストールディレクトリ¥mpwalker¥bin 存在しない場合は、Systemwalkerを再インストールするか、保守情報収集ツールを使用し[イベント監視]の資料を採取して、富士通技術員に連絡してください。
MPACT_EXCEPT	不当な領域のアドレスが指定されました。	内部論理異常が発生しています。保守情報収集ツールを使用し資料を採取して、富士通技術員に連絡してください。

プログラム例

アクション停止APIのプログラム例を以下に示します。

```
#include "f3crhxac.h"

unsigned char AppName[] = "実行API" ;     // アクション実行APIに指定したアプリ名
long handle           = 1 ;             // アクション実行APIの戻り値
```

```
long lrc ;

lrc = Mp_StopAction(AppName,handle) ;
if ( lrc != 0 ) {
    /* エラー処理 */
}

/* 正常終了 */
```

17.4.4 アクション状態通知API

本項では、“アクション状態通知API”について説明します。

“アクション状態通知API”の一覧を以下に示します。

- アクション状態通知API(Mp_GetActionInfo2)
- アクション情報領域の解放API(Mp_FreeActionInfo2)

17.4.4.1 アクション状態通知API

アクション状態通知API(Mp_GetActionInfo2)について説明します。

記述形式

```
long Mp_GetActionInfo2 (long handle,
                        MpActionInfo2 **ActionInfo,
                        int *nActionInfo)
```

機能説明

指定されたアクションの現在の状態を通知します。

パラメタの説明

handle

アクションの管理番号(アクション実行APIの復帰値)を指定します。

ActionInfo

アクション状態情報構造体(MpActionInfo2)を指す領域のアドレスを指定します。APIでアクション状態情報構造体が確保されます。本パラメタが指す領域にアクション状態情報構造体が設定されます。アクション状態情報構造体は、アクション情報領域の解放API(Mp_FreeActionInfo2)で解放されます。

nActionInfo

アクションの数を格納する領域のアドレスを指定します。本パラメタが指す領域にアクション状態情報構造体の配列数が設定されます。

復帰値

復帰値	意 味	対 処
0	停止処理を行います。	-
MPACT_NOTACTION	指定されたアクションは存在しません(すでに終了している)。	-
MPACT_PARAMERR	パラメタエラーです。	パラメタに誤りがないか確認してください。

復帰値	意味	対処
MPACT_ALLOCFAIL	領域の獲得に失敗しました。	ページファイルのサイズを拡張するか、メモリを増設してください。
MPACT_NOTMOVE	アクション管理サーバが起動されていません。	サービスSystemwalker MpAosfBが起動されていることを確認してください。
MPACT_COMFAIL	アクション管理サーバとの通信に失敗しました。	サービスSystemwalker MpAosfB、およびSystemwalker MpAosfXが起動されていることを確認してください。
MPACT_SYSERR	アクション管理サーバでエラーが発生しました。	イベントログ/SYSLOGに出力されたメッセージを参照しエラーを取り除いた後、Systemwalkerを再起動してください。エラーを取り除き、Systemwalker を再起動しても再現する場合は、保守情報収集ツールを使用し[イベント監視]の資料を採取して、富士通技術員に連絡してください。
MPACT_NOTDLL	アクション要求用DLLがロードできません。	以下のフォルダにアクション管理に必要なファイルf3crhxac.dllがあるか確認してください。 Systemwalkerのインストールディレクトリ¥mpwalker¥bin 存在しない場合は、Systemwalkerを再インストールするか、保守情報収集ツールを使用し[イベント監視]の資料を採取して、富士通技術員に連絡してください。
MPACT_EXCEPT	不当な領域のアドレスが指定されました。	内部論理異常が発生しています。保守情報収集ツールを使用し資料を採取して、富士通技術員に連絡してください。

アクション状態情報構造体(MpActionInfo2)の形式

```
typedef struct MpActionInfo2_tag {
    long no ;
    long action ;
    char *exehost;
    MpActionAddr *addr ;
    int  naddr ;
    int  status ;
    int  code ;
} MpActionInfo2 ;
```

no

アクションの管理番号が設定されます。

action

アクションの種類が設定されます。

ACTION_EMAIL:

E-Mailの送信です。

ACTION_MSMAIL:

MS-Mailの送信です。

ACTION_POPUP:

ポップアップメッセージの送信です。

ACTION_PAGER:

ショートメールへのメッセージ送信です。

ACTION_SOUND:

サウンドの再生です。

exehost

アクションを実行するホスト名が設定されます。

addr

アクション実行先情報構造体(MpActionAddr)のアドレスを指定します。アクションの実行先のアドレス情報を格納します。

naddr

addrに設定したアドレス数が設定されます。

status

アクションの状態が設定されます。

STATUS_NORMAL:

正常終了しました。

STATUS_EXEC:

実行中です。

STATUS_EXEWAIT:

実行待ちです。

STATUS_ERROR:

異常終了しました。

STATUS_STOP:

一時停止中です。

code

状態が異常終了の場合に設定します。なお、詳細コードについては、後述の“詳細コード”を参照してください。

アクション実行先情報構造体(MpActionAddr)の形式

```
typedef struct MpActionAddr_tag {
    unsigned char *name ;
    char *address ;
} MpActionAddr ;
```

name

アクション実行APIにおいて、宛先名を指定した場合は、指定された宛先名のアドレスが設定されます。アドレスを指定した場合は、NULLが設定されます。

address

送信先(アドレス)のアドレスが設定されます。

メールの場合:

メールアドレス設定されます。

ポップアップメッセージの場合:

ユーザ名またはコンピュータ名が設定されます。

ショートメールの場合:

電話番号が設定されます。

詳細コード

アクション状態通知API(Mp_GetActionInfo)内で獲得されたアクションの状態について記述します。

詳細コード	意味	アクション
MPACT_D_NOMEMORY	アクション実行中にメモリ不足が発生した	すべて
MPACT_D_FILEACCESS	アクション実行中にファイル操作でエラーが発生した	
MPACT_D_NOFILE	指定されたファイルが存在しない	
MPACT_D_DEFFILE	アクション実行環境が定義されていない	
MPACT_D_FAIL	何らかの理由でアクション実行に失敗した	
MPACT_D_NOTMOVE	アクション実行サーバが起動されていない	
MPACT_D_NOHOST	指定されたアクション実行ホストが存在しない	
MPACT_D_SENDFAIL	メッセージの送信に失敗した	ポップアップ
MPACT_D_NOTSERVER	ＳＭＴＰサーバが見つからない	E-Mail
MPACT_D_NOTMYHOST	自ホスト名が獲得できない	
MPACT_D_SMTPCOMMAND	ＳＭＴＰサーバでの処理でエラーが発生した	
MPACT_D_SMTPCOMM	ＳＭＴＰサーバとの通信でエラーが発生した	
MPACT_D_NOTCONNECT	ＳＭＴＰサーバと接続できない	
MPACT_D_FROMADDR	送信元アドレスが指定されていない	
MPACT_D_SIGNIN	サインインに失敗した	MS-Mail
MPACT_D_NOSEND	メール送信処理でエラーが発生した	
MPACT_D_COMMOPEN	ＣＯＭポート(モデム)のオープンに失敗した	ショートメール
MPACT_D_COMMFAIL	ＣＯＭポート(モデム)の入出力でエラーが発生した	
MPACT_D_INITCOMM	ＣＯＭポート(モデム)の設定に失敗した	

詳細コード	意 味	アクション
MPACT_D_LINEBUSY	通話中のため、送信できなかった	
MPACT_D_LINEFAIL	電話回線の異常のため、送信できなかった	
MPACT_D_CONVERTFAIL	メッセージの変換処理に失敗した	
MPACT_D_NOTSOUND	音声再生用の環境が設定されていない	音声通知
MPACT_D_NOTDEVICE	他のアプリケーション音声再生用のドライバを使用中	
MPACT_D_CREATESOUND	音声再生処理に失敗した	

プログラム例

アクション状態通知APIのプログラム例を以下に示します。

```
#include <windows.h>
#include "f3crhxac.h"

long handle = 1 ;              // 状態を獲得するアクションの管理番号
MpActionInfo2 *pInfo = NULL ;  // 情報の領域のアドレス設定域
int nInfo = 0 ;
int lrc;

lrc = Mp_GetActionInfo2(handle,&pInfo,&nInfo);
if ( lrc != 0 ) {
    /* エラー処理 */
}

/* アクション情報(pInfo)の参照 */

/* アクション情報域(pInfo)の解放 */
Mp_FreeActionInfo2(pInfo,nInfo) ;
```

17.4.4.2 アクション情報領域の解放API

アクション情報領域の解放API(Mp_FreeActionInfo2)について説明します。

記述形式

```
void Mp_FreeActionInfo2 (MpActionInfo2 *ActionInfo,
                         int nAction)
```

機能説明

アクション状態通知API(Mp_GetActionInfo2)内で獲得された領域を解放します。

パラメタの説明

ActionInfo

アクション状態通知APIで獲得されたアクション状態情報構造体(MpActionInfo2)のアドレスを指定します。

アクション状態情報構造体については、“17.4.4.1 アクション状態通知API”を参照してください。

nAction

ActionInfoの領域に設定されているアクションの数を示します。

プログラム例

アクション情報領域の解放APIのプログラム例を以下に示します。

```
#include <windows.h>
#include "f3crhxac.h"

long handle = 1 ;                // 状態を獲得するアクションの管理番号
MpActionInfo2 *pInfo = NULL ;    // 情報の領域のアドレス設定域
int nInfo = 0 ;
long lrc;

lrc = Mp_GetActionInfo2(handle,&pInfo,&nInfo);
if ( lrc != 0 ) {
    /* エラー処理 */
}

/* アクション情報(pInfo)の参照 */

/* アクション情報域(pInfo)の解放 */
Mp_FreeActionInfo2(pInfo,nInfo) ;
```

17.4.5 アクション状態通知API(旧バージョン互換)

本項では、旧バージョンのSystemwalker Operation Managerで提供された“アクション状態通知API”について説明します。

“アクション状態通知API”の一覧を以下に示します。

- アクション状態通知API(Mp_GetActionInfo)
- アクション情報領域の解放API(Mp_FreeActionInfo)

旧バージョンのSystemwalker Operation Managerとは、ここでは以下のバージョンを指しています。

- SystemWalker/OperationMGR V4.0

17.4.5.1 アクション状態通知API

アクション状態通知API(Mp_GetActionInfo)について説明します。

記述形式

```
long Mp_GetActionInfo (long handle,
                       MpActionInfo **ActionInfo,
                       int *nActionInfo)
```

機能説明

指定されたアクションの現在の状態を通知します。

パラメタの説明

handle

アクションの管理番号(アクション実行APIの復帰値)を指定します。

ActionInfo

アクション状態情報構造体(MpActionInfo)を指す領域のアドレスを指定します。APIでアクション状態情報構造体が確保されます。本パラメタが指す領域にアクション状態情報構造体が設定されます。アクション状態情報構造体は、アクション情報領域の解放API(Mp_FreeActionInfo)で解放されます。

nActionInfo

アクションの数を格納する領域のアドレスを指定します。本パラメタが指す領域にアクション状態情報構造体の配列数が設定されます。

復帰値

復帰値	意 味	対 処
0	停止処理を行います。	-
MPACT_NOTACTION	指定されたアクションは存在しません(すでに終了している)。	-
MPACT_PARAMERR	パラメタエラーです。	パラメタに誤りがないか確認してください。
MPACT_ALLOCFAIL	領域の獲得に失敗しました。	ページファイルのサイズを拡張するか、メモリを増設してください。
MPACT_NOTMOVE	アクション管理サーバが起動されていません。	サービスSystemwalker MpAosfBが起動されていることを確認してください。
MPACT_COMFAIL	アクション管理サーバとの通信に失敗しました。	サービスSystemwalker MpAosfB、およびSystemwalker MpAosfXが起動されていることを確認してください。
MPACT_SYSERR	アクション管理サーバでエラーが発生しました。	イベントログ/SYSLOGに出力されたメッセージを参照しエラーを取り除いた後、Systemwalkerを再起動してください。エラーを取り除き、Systemwalker を再起動しても再現する場合は、保守情報収集ツールを使用し[イベント監視]の資料を採取して、富士通技術員に連絡してください。
MPACT_NOTDLL	アクション要求用DLLがロードできません。	以下のフォルダにアクション管理に必要なファイルf3crhxac.dllがあるか確認してください。 Systemwalkerのインストールディレクトリ¥mpwalker¥bin 存在しない場合は、Systemwalkerを再インストールするか、保守情報収集ツールを使用し[イベント監視]の資料を採取して、富士通技術員に連絡してください。
MPACT_EXCEPT	不当な領域のアドレスが指定されました。	内部論理異常が発生しています。保守情報収集ツールを使用し資料を採取して、富士通技術員に連絡してください。

アクション状態情報構造体(MpActionInfo)の形式

```
typedef struct MpActionInfo_tag {
```

```
    long no ;
    long action ;
    MpActionAddr *addr ;
    int  naddr ;
    int  status ;
    int  code ;
} MpActionInfo ;
```

no

アクションの管理番号が設定されます。

action

アクションの種類が設定されます。

ACTION_EMAIL:

E-Mailの送信です。

ACTION_MSMAIL:

MS-Mailの送信です。

ACTION_POPUP:

ポップアップメッセージの送信です。

ACTION_PAGER:

ショートメールへのメッセージ送信です。

ACTION_SOUND:

サウンドの再生です。

addr

アクション実行先情報構造体(MpActionAddr)のアドレスを指定します。アクションの実行先のアドレス情報を格納します。

naddr

addrに設定したアドレス数が設定されます。

status

アクションの状態が設定されます。

STATUS_NORMAL:

正常終了しました。

STATUS_EXEC:

実行中です。

STATUS_EXEWAIT:

実行待ちです。

STATUS_ERROR:

異常終了しました。

STATUS_STOP:

一時停止中です。

code

状態が異常終了の場合に詳細コードを設定します。なお、詳細コードについては、後述の“詳細コード”を参照してください。

アクション実行先情報構造体(MpActionAddr)の形式

```
typedef struct MpActionAddr_tag {
    unsigned char *name ;
    char *address ;
} MpActionAddr ;
```

name

アクション実行APIにおいて、宛先名を指定した場合は、指定された宛先名のアドレスが設定されます。アドレスを指定した場合は、NULLが設定されます。

address

送信先(アドレス)のアドレスが設定されます。

メールの場合:

メールアドレス設定されます。

ポップアップメッセージの場合:

ユーザ名またはコンピュータ名が設定されます。

ショートメールの場合:

電話番号が設定されます。

詳細コード

アクション状態通知API(Mp_GetActionInfo)内で獲得されたアクションの状態について記述します。

詳細コード	意 味	アクション
MPACT_D_NOMEMORY	アクション実行中にメモリ不足が発生した	すべて
MPACT_D_FILEACCESS	アクション実行中にファイル操作でエラーが発生した	
MPACT_D_NOFILE	指定されたファイルが存在しない	
MPACT_D_DEFFILE	アクション実行環境が定義されていない	
MPACT_D_FAIL	何らかの理由でアクション実行に失敗した	
MPACT_D_NOTMOVE	アクション実行サーバが起動されていない	
MPACT_D_NOHOST	指定されたアクション実行ホストが存在しない	
MPACT_D_SENDFAIL	メッセージの送信に失敗した	ポップアップ
MPACT_D_NOTSERVER	ＳＭＴＰサーバが見つからない	E-Mail
MPACT_D_NOTMYHOST	自ホスト名が獲得できない	

詳細コード	意味	アクション
MPACT_D_SMTPCOMMAND	ＳＭＴＰサーバでの処理でエラーが発生した	
MPACT_D_SMTPCOMM	ＳＭＴＰサーバとの通信でエラーが発生した	
MPACT_D_NOTCONNECT	ＳＭＴＰサーバと接続できない	
MPACT_D_FROMADDR	送信元アドレスが指定されていない	
MPACT_D_SIGNIN	サインインに失敗した	MS-Mail
MPACT_D_NOSEND	メール送信処理でエラーが発生した	
MPACT_D_COMMOPEN	ＣＯＭポート(モデム)のオープンに失敗した	ショートメール
MPACT_D_COMMFAIL	ＣＯＭポート(モデム)の入出力でエラーが発生した	
MPACT_D_INITCOMM	ＣＯＭポート(モデム)の設定に失敗した	
MPACT_D_LINEBUSY	通話中のため、送信できなかった	
MPACT_D_LINEFAIL	電話回線の異常のため、送信できなかった	
MPACT_D_CONVERTFAIL	メッセージの変換処理に失敗した	
MPACT_D_NOTSOUND	音声再生用の環境が設定されていない	音声通知
MPACT_D_NOTDEVICE	他のアプリケーション音声再生用のドライバを使用中	
MPACT_D_CREATESOUND	音声再生処理に失敗した	

プログラム例

アクション状態通知APIのプログラム例を以下に示します。

```
#include <windows.h>
#include "f3crhxac.h"

long handle = 1 ;                // 状態を獲得するアクションの管理番号
MpActionInfo *pInfo = NULL ;     // 情報の領域のアドレス設定域
int nInfo = 0 ;
long lrc;

lrc = Mp_GetActionInfo(handle,&pInfo,&nInfo);
if ( lrc != 0 ) {
    /* エラー処理 */
}

/* アクション情報(pInfo)の参照 */
```

```
/* アクション情報域(pInfo)の解放 */
Mp_FreeActionInfo(pInfo,nInfo) ;
```

17.4.5.2　アクション情報領域の解放API

アクション情報領域の解放API(Mp_FreeActionInfo)について説明します。

記述形式

```
void Mp_FreeActionInfo (MpActionInfo *ActionInfo,
                        int nAction)
```

機能説明

アクション状態通知API(Mp_GetActionInfo)内で獲得された領域を解放します。

パラメタの説明

ActionInfo

アクション状態通知APIで獲得されたアクション状態情報構造体(MpActionInfo)のアドレスを指定します。

アクション状態情報構造体については、“17.4.5.1 アクション状態通知API”を参照してください。

nAction

ActionInfoの領域に設定されているアクションの数を示します。

プログラム例

アクション情報領域の解放APIのプログラム例を以下に示します。

```
#include <windows.h>
#include "f3crhxac.h"

long handle = 1 ;                // 状態を獲得するアクションの管理番号
MpActionInfo *pInfo = NULL ;     // 情報の領域のアドレス設定域
int nInfo = 0 ;
long lrc;

lrc = Mp_GetActionInfo(handle,&pInfo,&nInfo);
if ( lrc != 0 ) {
    /* エラー処理 */
}

/* アクション情報(pInfo)の参照 */

/* アクション情報域(pInfo)の解放 */
Mp_FreeActionInfo(pInfo,nInfo) ;
```

第18章　Web API【Windows版】【Linux版】

本章では、Systemwalker Operation Managerが提供するWeb APIについて説明します。

18.1　Web APIの概要

Web APIでは、Systemwalker Operation Managerサーバの各リソースのエンドポイントにHTTPリクエストを送信することで、各リソースの操作(参照・追加・変更・削除)を行い、その結果をHTTPレスポンスで受け取ることができます。

Web APIの機能、動作環境および注意事項を以下に示します。

機能一覧

Web APIでは、以下の機能を使用することができます。

ジョブスケジューラ

- プロジェクトに登録されているジョブネットの状態ごとの数の参照
- プロジェクトに登録されているジョブネットの一括操作
- ジョブネット情報の参照
- ジョブネットの操作(強制終了/起動/再起動/停止/停止解除/無効/無効解除/確認/終了取消/持越解除)
- ジョブ情報の参照
- ジョブの操作

ジョブ実行制御

- キュー情報の参照
- キューの操作(停止/開始/定義の一時変更)
- キューに設定したホストグループに登録されている構成ホストの参照
- キューに設定したホストグループに登録されている構成ホストの定義の一時変更
- キューに設定したホストグループへの構成ホストの登録
- キューに設定したホストグループから構成ホストの削除

注意事項

当APIを利用するためにはWebサーバのセットアップが必要です。Webサーバセットアップコマンドにより、Webサーバをアンセットアップしている場合は利用できません。

18.2　APIリファレンス

18.2.1　API形式

API形式は、プログラム言語に依存しないREST形式のインタフェースとなります。

- リクエストおよびレスポンスのデータフォーマットはJSONを使用します。
- 文字コードはUTF-8を使用します。
- HTTPプロトコルはバージョン HTTP/1.1を使用します

URL

URLの形式は以下になります。

http://Systemwalker Operation Managerサーバのホスト名:9900/Systemwalker-omgr/api/[各リソースのエンドポイント]

HTTPSを使用する場合は、以下になります。

https://Systemwalker Operation Managerサーバのホスト名:9900/Systemwalker-omgr/api/[各リソースのエンドポイント]

Webサーバ用ポート番号をデフォルト値“9900”から変更している場合は、URL内の“9900”を、変更後のポート番号に置き換えて指定してください。

18.2.2 APIー覧

提供するAPIの一覧は、以下のとおりです。

- エンドポイント列に記載の{}は、各リソースの識別子を示しています。
- エンドポイントには、サブシステム番号を指定することができます。なお、表中のエンドポイントはサブシステム番号を省略しています。詳細については、“18.2.3.5 サブシステム番号”を参照してください。
- HTTPメソッド列には、各リソースのエンドポイントで指定可能なHTTPメソッドを示しています。(○：指定可、×：指定不可)

リソース	エンドポイント	HTTPメソッド				参照先
		GET	POST	PUT	DELETE	
プロジェクト	/v1/projects	○	×	×	×	18.3.1 projectsリソース
	/v1/projects/{projectname}	○	○	×	×	
ジョブネット	/v1/jobnets	○	×	×	×	18.3.2 jobnetsリソース
	/v1/projects/{projectname}/jobnets/{jobnetname}	○	○	×	×	
ジョブ	/v1/jobs	○	×	×	×	18.3.3 jobsリソース
	/v1/projects/{projectname}/jobnets/{jobnetname}/jobs/{jobnumber}	○	○	×	×	
キュー	/v1/queues	○	×	×	×	18.3.4 queuesリソース
	/v1/queues/{queuename}	○	○	○	×	
構成ホスト	/v1/contained-hosts	○	○	×	×	18.3.5 contained-hostsリソース
	/v1/host-groups/{hostgroupname}/contained-hosts/{hostname}	○	×	○	○	

18.2.3 共通仕様

Web APIを利用する場合の、各APIの共通仕様について説明します。

18.2.3.1 データ型

Web APIで利用するデータ型は下表のとおりです。

リクエストでの各パラメータの指定や、レスポンスボディのフィールド値の参照はデータ型に合わせてください。

型名	対応するJSONの型	説明
文字列	string	テキストデータです。 文字コードはUTF-8を使用します。 ・ パスパラメータ URLに利用可能な文字以外が含まれる場合、URLエンコード後の文字列をリクエスト行に指定してください。 ・ リクエストボディ JSONで指定します。ダブルクォーテーションやバックスラッシュ、および、制御文字などを含める場合、JSONのフォーマットに従いエスケープした文字列をリクエストボディに指定してください。
数値	number	数値データです。
リスト	string	規定値を扱うデータです。 規定値はフィールドごとに異なります。詳細は各フィールドの説明を参照してください。
日時	string	日付と時刻のデータです。 日付と時刻のフォーマットは以下のとおりです。 ・ YYYY/MM/DD hh:mm:ss ・ YYYY/MM/DD YYYY：年を4桁で0000～9999で表します。 MM：月を2桁で01～12で表します。 DD：日を2桁で01～指定月の最終日で表します。 hh：時を2桁で00～23で表します。 mm：分を2桁で00～59で表します。 ss：秒を2桁で00～59で表します。
真偽値	true, false	真偽値のデータです。 trueまたはfalseが使用できます。

18.2.3.2 リクエストの形式

リクエスト行

リクエスト行は、利用するリソースに合わせて指定します。

リクエスト行はASCII文字で指定します。パス名の一部(パスパラメータ)にマルチバイト文字が含まれる場合は、UTF-8でURLエンコードしてください。

リクエスト行	指定内容
メソッド	各リソースで使用可能なHTTPメソッドを指定します。
パス名	各リソースのURIおよびクエリ文字列を指定します。
HTTPバージョン	“HTTP/1.1”を指定します。

各リソースの一覧取得をする場合、絞り込みやレスポンスの制御のためクエリパラメータを使用します。一覧取得で使用できるクエリパラメータの共通仕様は、以下のとおりです。

パラメータ名	省略値	説明
subsystem	“18.2.3.5 サブシステム番	一覧取得する対象のサブシステムを指定します。指定可能な値は“18.2.3.5 サブシステム番号”を参照してください。

パラメータ名	省略値	説明
	号”を参照	
offset	0	返却するレコードの開始位置を指定します。 指定された値の分だけ、返却するオブジェクト配列の要素をスキップします。
limit	15000	返却するレコードの最大数を指定します。 指定された値よりもオブジェクト配列の要素が多い場合、リクエストボディのnextフィールド値にtrueを設定したうえで、指定値までのレコードを返却します。 指定可能な値は0～15000です。

リクエストヘッダ

以下のリクエストヘッダを指定します。

リクエストヘッダ	指定内容
Host	接続先のSystemwalker Operation Managerサーバのホスト名またはIPアドレスとポート番号を以下の形式で指定します。 host:port
Authorization	Basic認証の認証情報を指定します。 詳細は、“18.2.3.4 ユーザ認証”を参照してください。
Content-Type	リクエストボディにパラメータを指定する場合に指定します。 “application/json”を指定します。

リクエストボディ

JSONでリクエストパラメータを指定します。

指定可能なパラメータは各リソースで異なります、詳細は、各リソースの“リクエスト”の項目を参照してください。

18.2.3.3 レスポンスの形式

ステータス行

リクエストの処理結果をHTTPステータスコードで返却します。

- 返却するHTTPステータスコードは、各リソースの“レスポンス”を参照してください。
- 正常時と異常時でレスポンスボディの形式が異なります。レスポンスボディは、HTTPステータスコードを確認してから利用してください。

レスポンスヘッダ

以下のレスポンスヘッダを設定して、返却します。

レスポンスヘッダ	設定内容	説明
Content-Type	application/json;charset=UTF-8	レスポンスボディのメディアタイプです。

レスポンスボディ

レスポンスヘッダ“Content-Type”のメディアタイプで返却します。

レスポンスボディの形式を以下に示します。

正常終了時の応答

各リソースのリクエストを正常に処理できた場合の形式は、以下のとおりです。

```
{
  "next": false,
  "total: n,
  "data": [                ←----+
    { … },                      |
    …                        各リソースのオブジェクト配列
    { … }                       |
  ]                          ←----+
}
```

n: 返却対象のオブジェクト配列の要素数

対象のオブジェクトが1件以上の場合は、dataフィールドに対象のオブジェクト配列が格納されます。nextおよびtotalフィールドの値で対象のオブジェクトがすべて返却されたかどうかを判断できます。

対象のオブジェクトが0件の場合、nextはfalse、totalは0、dataは空配列となります。

フィールド名	型	説明
next	真偽値	返却対象のオブジェクトのうち、dataフィールドに格納していないオブジェクトが存在するかを真偽値で示します。返却対象のオブジェクト配列の要素数がクエリパラメータのlimitで指定した値以上である場合に、trueとなります。 true: 未返却のオブジェクトが存在する false: 未返却のオブジェクトは存在しない
total	数値	返却対象のオブジェクト配列の要素数です。 dataフィールドに格納されたオブジェクト配列の要素数ではありません。nextフィールドがtrueの場合、この値とdataフィールドに格納されたオブジェクト配列の要素数は異なります。
data	－	返却対象のオブジェクト配列です。 クエリパラメータのlimitで指定した値までの要素数を格納します。

異常終了時の応答

注意

異常時のレスポンスボディでtext/htmlのようにJSON以外のデータフォーマットが返却される場合があります。レスポンスヘッダ“Content-Type”を確認し、適切なデータフォーマットで参照してください。

各リソースのリクエストの処理で異常が発生した場合の形式は、以下のとおりです。

```
{
  "code": "エラーコード",
  "message": "メッセージ文"
}
```

HTTPステータスコードやエラーコード、メッセージ文によりエラー情報を返却します。エラーコードとメッセージ文の詳細は“Systemwalker Operation Manager メッセージ集”の“Web APIの出力メッセージ【Windows版】【Linux版】”を参照してください。

フィールド名	型	説明
code	文字列	エラー情報を示すコードです。
message	文字列	エラー情報を示すメッセージです。

18.2.3.4　ユーザ認証

APIのリクエストごとに認証が必要です。

リクエストヘッダーにBasic認証の情報を設定します。

ユーザIDとパスワードは、接続先のSystemwalker Operation Managerサーバ導入マシンの認証情報です。

拡張ユーザ管理機能が有効な場合は、Operation Managerユーザを指定してください。【Linux版】

指定したユーザIDの権限で各リソースを操作します。必要な権限は各リソースの説明を参照してください。

```
Authorization: Basic <“ユーザID:パスワード” をBase64エンコードした文字列>
```

18.2.3.5　サブシステム番号

サブシステム番号の指定方法は以下のとおりです。

SE版

- パスパラメータ

  /subsystems/{subsystem}は指定不要です。

  サブシステム番号0として指定することはできます。1～9の値は指定できません。

- クエリパラメータ

  ?subsystem={subsystem}は指定不要です。

  サブシステム番号0として指定することはできます。1～9の値を指定した場合、一覧取得の絞り込み結果は0件となります。

EE版

- パスパラメータ

  /subsystems/{subsystem}で指定できます。

  サブシステム番号は、リソースを一意に特定するために必要です。

  指定した場合はそのサブシステム番号を、省略した場合はサブシステム番号0を対象とします。

- クエリパラメータ

  ?subsystem={subsystem}で指定できます。

  省略した場合、すべてのサブシステム番号を対象に処理を行います。そのため、意図して起動させていないサブシステムがある場合は、対象とならないよう、対象のサブシステムだけを指定してください。

リクエスト例 (/Systemwalker-omgr/api 以降)

- jobnetsリソースでジョブネットの操作をする場合
  - サブシステム番号を指定する場合

    /v1/subsystem/1/projects/{projectname}/jobnets/{jobnetname} 【EE】

    /v1/subsystem/0/projects/{projectname}/jobnets/{jobnetname}

  - サブシステム番号を省略する場合

    /v1/projects/{projectname}/jobnets/{jobnetname}

- jobnetsリソースでジョブネット情報の一覧取得をする場合
  - サブシステム番号を指定する場合

    /v1/jobnets?subsystem=1

    /v1/jobnets?subsystem=0
  - サブシステム番号を省略する場合

    /v1/jobnets

18.3 各APIのリファレンス

18.3.1 projectsリソース

Systemwalker Operation Managerに登録済みのプロジェクトを操作します。

以下の操作が可能です。

エンドポイント (/Systemwalker-omgr/api 以降)	HTTP メソッド	操作内容
/v1/projects/{projectname}	GET	指定したプロジェクトの情報を取得します。
	POST	指定したプロジェクトに存在するジョブネットを一括操作します。
/v1/projects	GET	プロジェクト情報の一覧を取得します。

projectsリソースのオブジェクトのフィールドは、以下のとおりです。

なお、システムで一意のプロジェクトを指定する場合、SE版では、プロジェクト名を指定します。EE版およびGEE版では、サブシステム番号とプロジェクト名を指定します。

フィールド名	型	説明
host	文字列	接続先のホスト名
subsystem	数値	サブシステム番号
projectname	文字列	プロジェクト名
normal	数値	正常終了のジョブネット数
abend	数値	異常終了のジョブネット数
forced	数値	強制終了のジョブネット数
warning	数値	警告のジョブネット数
execute	数値	実行中のジョブネット数
wait	数値	実行待ちのジョブネット数
pause	数値	一時停止のジョブネット数
disable	数値	無効状態のジョブネット数
coped	数値	確認済のジョブネット数
pseudo	数値	疑似正常のジョブネット数

18.3.1.1 プロジェクト情報の取得

機能説明

projectsリソースを指定したGETメソッドのリクエストを送ることで、対象のプロジェクト情報を取得します。

実行に必要な権限/実行環境

指定のプロジェクトに対して、参照権以上の権限が必要です。

リクエスト

HTTPメソッド

GET

リクエストURI

/v1/subsystems/{subsystem}/projects/{projectname}

/v1/projects/{projectname}（サブシステム番号を省略する場合)

リクエストパラメータ

パスパラメータで、以下のパラメータを指定します。

パラメータ	指定方法	型	説明
subsystem	パス	数値	対象のプロジェクトが存在するサブシステム番号(0～9)を指定します。サブシステム番号を省略する場合は0になります。
projectname(*)	パス	文字列	対象のプロジェクト名を指定します。

(*)：必須のパラメータ

パス：パスパラメータ

レスポンス

ステータスコード

以下のHTTPステータスコードを返却します。

コード	説明
200	リクエストの処理が正常終了しました。
401	認証情報の指定に誤りがあります。
404	対象のプロジェクトは存在しません。 ・ システムに対象のプロジェクトが存在しない ・ パスパラメータの指定内容が誤っている
500	リクエストの処理中にサーバで異常が発生しました。

レスポンスボディ

対象のprojectsリソースのオブジェクトを返却します。

使用例

サブシステム番号0、プロジェクト名が“SampleProject”のプロジェクトを取得する場合のAPIの使用例を以下に示します。なお、レスポンス例は、dataフィールドに格納されるprojectsリソースのオブジェクトを示しており、レスポンスヘッダおよび他のフィールドは省略しています。

リクエスト例

```
GET /Systemwalker-omgr/api/v1/subsystems/0/projects/SampleProject HTTP/1.1
Host: Systemwalker Operation Managerサーバのホスト名:9900
Authorization: Basic ********************
```

レスポンス例

```
{
  "host": "Systemwalker Operation Managerサーバのホスト名",
  "subsystem": 0,
  "projectname": "SampleProject",
  "normal": 0,
  "abend": 0,
  "forced": 0,
  "warning": 0,
  "execute": 0,
  "wait": 0,
  "pause": 0,
  "disable": 0,
  "coped": 0,
  "pseudo": 0
}
```

18.3.1.2　プロジェクトの操作

機能説明

projectsリソースを指定したPOSTメソッドのリクエストを送ることで、対象のプロジェクトに存在するジョブネットを一括操作(すべて停止/すべて停止解除/すべて無効/すべて無効解除)します。

実行に必要な権限/実行環境

指定のプロジェクトに対して、操作権以上の権限が必要です。

リクエスト

HTTPメソッド

POST

リクエストURI

/v1/subsystems/{subsystem}/projects/{projectname}

/v1/projects/{projectname}（サブシステム番号を省略する場合)

リクエストパラメータ

パスパラメータおよびリクエストボディで、以下のパラメータを指定します。

パラメータ	指定方法	型	説明
subsystem	パス	数値	対象のプロジェクトが存在するサブシステム番号(0～9)を指定します。サブシステム番号を省略する場合は0になります。
projectname(*)	パス	文字列	対象のプロジェクト名を指定します。
control(*)	ボディ	リスト	指定可能な値は以下のとおりです。 pause　：すべて停止 continue ：すべて停止解除 disable　：すべて無効 enable　：すべて無効解除

(*)：必須のパラメータ

パス：パスパラメータ

ボディ：リクエストボディ

レスポンス

ステータスコード

以下のHTTPステータスコードを返却します。

コード	説明
202	リクエストの処理が正常終了しました。
400	リクエストパラメータに誤りがあります。 ・ 必須パラメータを指定していない ・ 指定不要のパラメータを指定している ・ パラメータの指定値に誤りがある
401	認証情報の指定に誤りがあります。
403	対象のプロジェクトへの権限がありません。
404	対象のプロジェクトは存在しません。
409	対象のプロジェクトへの指定の操作はできません。 ・ プロジェクトにジョブネットが存在しない
500	リクエストの処理中にサーバで異常が発生しました。

レスポンスボディ

指定したprojectsリソースのオブジェクトを返却します。

注意

- Web APIによるプロジェクトに存在するジョブネットへの操作は、Systemwalker Operation Managerサーバの処理と同期を取っていません。そのため、Web APIのレスポンスはサーバでの処理の完了を保証するものではありません。レスポンスボディで返却するオブジェクトは、操作が反映されていない場合があります。
- 対象のプロジェクトに存在するジョブネットの中に操作できないジョブネットがあった場合、処理は中断されず、操作できるジョブネットはすべて処理されます。この場合、異常終了時の応答の形式で応答します。

使用例

サブシステム番号0、プロジェクト名が“SampleProject”のプロジェクトに対して、“すべて停止”の操作をする場合のAPIの使用例を次に示します。

リクエスト例

```
POST /Systemwalker-omgr/api/v1/subsystems/0/projects/SampleProject HTTP/1.1
Host: Systemwalker Operation Managerサーバのホスト名:9900
Authorization: Basic **********
Content-Type: application/json

{
  ”control”: ”pause”
}
```

レスポンス例

“18.3.1.1 プロジェクト情報の取得”のレスポンス例を参照してください。

18.3.1.3 プロジェクト情報の一覧取得

機能説明

projectsリソースの一覧を取得します。

対象は、認証情報に指定したユーザが参照権以上を持つプロジェクトとなります。

リクエスト

HTTPメソッド

GET

リクエストURI

/v1/projects

リクエストパラメータ

共通仕様のクエリパラメータが使用できます。

詳細については、“18.2.3.2 リクエストの形式”を参照してください。

レスポンス

ステータスコード

以下のHTTPステータスコードを返却します。

コード	説明
200	リクエストの処理が正常終了しました。
400	リクエストパラメータに誤りがあります。 ・ 指定不要のパラメータを指定している ・ 同じクエリパラメータを複数指定している
401	認証情報の指定に誤りがあります。
500	リクエストの処理中にサーバで異常が発生しました。

レスポンスボディ

projectsリソースのオブジェクトを返却します。

使用例

サブシステム番号0に所属するプロジェクトの一覧を取得する場合の、APIの使用例を以下に示します。

リクエスト例

```
GET /Systemwalker-omgr/api/v1/projects?subsystem=0 HTTP/1.1
Host: Systemwalker Operation Managerサーバのホスト名:9900
Authorization: Basic **********
```

レスポンス例

dataフィールドのオブジェクト配列として、projectsリソースのオブジェクトを返却します。projectsリソースのオブジェクトについては、“18.3.1.1 プロジェクト情報の取得”のレスポンス例を参照してください。

18.3.2 jobnetsリソース

Systemwalker Operation Managerに登録済みのジョブネットを操作します。

以下の操作が可能です。

エンドポイント (/Systemwalker-omgr/api 以降)	HTTP メソッド	操作内容
/v1/projects/{projectname}/ jobnets/{jobnetname}	GET	指定したジョブネットの情報を取得します。
	POST	指定したジョブネットを操作します。
/v1/jobnets	GET	ジョブネット情報の一覧を取得します。

jobnetsリソースのオブジェクトのフィールドは、以下のとおりです。

なお、システムで一意のプロジェクトを指定する場合、SE版では、プロジェクト名とジョブネット名を指定します。
EE版およびGEE版では、サブシステム番号とプロジェクト名、ジョブネット名を指定します。

フィールド名	型	説明
host	文字列	接続先のホスト名
subsystem	数値	サブシステム番号
projectname	文字列	プロジェクト名
jobnetname	文字列	ジョブネット名
jobnetcomment	文字列	ジョブネット名称
parentname	文字列	親ジョブネット名 子ジョブネットまたはリンクジョブネットの場合、親ジョブネット名を返却します。また、グループに登録されたジョブネットの場合、グループ名を返却します。 上記のいずれでもない場合、空文字を返却します。
layertype	リスト	階層の種別(注1)
linktype	リスト	リンクジョブネットの種別(注2)
copystarttype	リスト	複写起動の種別(注3)
execcondition	リスト	ジョブネットの起動条件(注4)
defaulthost	文字列	デフォルトホスト名 複数指定されている場合は、,(カンマ)区切りで返却します。
execpattern	文字列	起動日雛形名 複数指定されている場合は、半角空白区切りで返却します。
holidaycalendar	文字列	カレンダ名 複数指定されている場合は、,(カンマ)区切りで返却します。
messagename	文字列	メッセージ事象名 複数指定されている場合は、,(カンマ)区切りで返却します。
status	リスト	ジョブネットの状態(注5)
remarks	文字列	ジョブネットメモ
starttime	日時	前回の開始日時
endtime	日時	前回の終了日時
scheduledtime	日時	次回の起動予定日時

フィールド名	型	説明
estimatedendtime	日時	次回の終了予定日時

注1)

layertypeフィールドの返却値は以下のとおりです。

フィールド値	階層の種別
None	なし
Parent	親ジョブネット
Child	子ジョブネット
Grouped	グループに登録されたジョブネット

注2)

linktypeフィールドの返却値は以下のとおりです。

フィールド値	リンクジョブネットの種別
None	なし
Linked	リンクジョブネット
MasterLinked	マスタリンクジョブネット

注3)

copystarttypeフィールドの返却値は以下のとおりです。

フィールド値	複写起動の種別
None	なし
CopySource	複写元ジョブネット
Copied	複写先ジョブネット

注4)

execconditionフィールドの返却値は以下のとおりです。

フィールド値	ジョブネットの起動条件
None	なし
SingleTime	時刻起動(単一起動)
MultiTime	時刻起動(複数起動)
Interval	間隔起動
ServerStartup	サーバ起動時に起動
MessageEvent	メッセージ事象発生時のみ起動

注5)

statusフィールドの返却値は以下のとおりです。

フィールド値	ジョブネットの状態
Abend	異常終了
Warning	警告
Forced	強制終了

フィールド値	ジョブネットの状態
CarryOver	持ち越し
StartDelay	開始遅延
EndDelay	終了遅延
Pause	一時停止
Execute	実行中
Wait	実行待ち
Pseudo	疑似正常
Normal	正常終了
Coped	確認済
Disable	無効状態

18.3.2.1 ジョブネット情報の取得

機能説明

jobnetsリソースを指定したGETメソッドのリクエストを送ることで、対象のジョブネット情報を取得します。

実行に必要な権限/実行環境

指定のジョブネットが存在するプロジェクトに対して、参照権以上の権限が必要です。

リクエスト

HTTPメソッド

GET

リクエストURI

/v1/subsystems/{subsystem}/projects/{projectname}/jobnets/{jobnetname}

/v1/projects/{projectname}/jobnets/{jobnetname}（サブシステム番号を省略する場合)

リクエストパラメータ

パスパラメータで、以下のパラメータを指定します。

パラメータ	指定方法	型	説明
subsystem	パス	数値	対象のジョブネットが存在するサブシステム番号(0～9)を指定します。サブシステム番号を省略する場合は0になります。
projectname(*)	パス	文字列	対象のジョブネットが存在するプロジェクト名を指定します。
jobnetname(*)	パス	文字列	対象のジョブネット名を指定します。

(*)：必須のパラメータ

パス：パスパラメータ

レスポンス

ステータスコード

以下のHTTPステータスコードを返却します。

コード	説明
200	リクエストの処理が正常終了しました。
401	認証情報の指定に誤りがあります。
404	対象のジョブネットは存在しません。 ・ システムに対象のジョブネットが存在しない ・ パスパラメータの指定内容が誤っている
500	リクエストの処理中にサーバで異常が発生しました。

レスポンスボディ

対象のjobnetsリソースのオブジェクトを返却します。

使用例

サブシステム番号0、プロジェクト名が“SampleProject”、ジョブネット名が“SampleJobnet”のジョブネットを取得する場合のAPIの使用例を以下に示します。なお、レスポンス例は、dataフィールドに格納されるjobnetsリソースのオブジェクトを示しており、レスポンスヘッダおよび他のフィールドは省略しています。

リクエスト例

```
GET /Systemwalker-omgr/api/v1/subsystems/0/projects/SampleProject/jobnets/SampleJobnet HTTP/1.1
Host: Systemwalker Operation Managerサーバのホスト名:9900
Authorization: Basic ********************
```

レスポンス例

```
{
  ”host”: ”Systemwalker Operation Managerサーバのホスト名”,
  ”subsystem”: 0,
  ”projectname”: ”SampleProject”,
  ”jobnetname”: ”SampleJobnet”,
  ”jobnetcomment”: ”SampleJobnet_Comment”,
  ”parentname”: ””,
  ”layertype”: ”None”,
  ”linktype”: ”None”,
  ”copystarttype”: ”None”,
  ”execcondition”: ”SingleTime”,
  ”defaulthost”: ”SampleHost”,
  ”execpattern”: ”execpattern1”,
  ”holidaycalendar”: ”holidaycalendar1”,
  ”messagename”: ”message1”,
  ”status”: ”Normal”,
  ”remarks”: ”SampleJobnet_Memo”,
  ”starttime”: ”2019/12/31 12:00:00”,
  ”endtime”: ”2019/12/31 12:00:01”,
  ”scheduledtime”: ”2020/01/01 12:00:00”,
  ”estimatedendtime”: ”2020/01/01 12:00:01”
}
```

18.3.2.2 ジョブネットの操作

機能説明

jobnetsリソースを指定したPOSTメソッドのリクエストを送ることで、対象のジョブネットを操作(強制終了/起動/再起動/停止/停止解除/無効/無効解除/確認/終了取消/持越解除)します。

実行に必要な権限/実行環境

指定のジョブネットが存在するプロジェクトに対して、操作権以上の権限が必要です。

リクエスト

HTTPメソッド

POST

リクエストURI

/v1/subsystems/{subsystem}/projects/{projectname}/jobnets/{jobnetname}

/v1/projects/{projectname}/jobnets/{jobnetname}（サブシステム番号を省略する場合)

リクエストパラメータ

パスパラメータおよびリクエストボディで、以下のパラメータを指定します。

パラメータ	指定方法	型	説明
subsystem	パス	数値	対象のジョブネットが存在するサブシステム番号(0～9)を指定します。サブシステム番号を省略する場合は0になります。
projectname(*)	パス	文字列	対象のジョブネットが存在するプロジェクト名を指定します。
jobnetname(*)	パス	文字列	対象のジョブネット名を指定します。
control(*)	ボディ	リスト	ジョブネットへの操作種別を指定します。(注1)
vparam	ボディ	文字列	可変パラメタを指定します。 controlパラメータがstartの場合に指定可能です。

(*)：必須のパラメータ

パス：パスパラメータ

ボディ：リクエストボディ

注1)

メッセージボディのcontrolパラメータで指定可能な値は、以下のとおりです。

指定値	説明
start	ジョブネットを起動します。 可変パラメタを指定することも可能です、詳細はvparamパラメータを参照してください。
cancel	ジョブネットを強制終了します。
cope	異常終了したジョブネットを確認済状態にします。
reinstate	ジョブネットの終了取消を行います。
revoke	ジョブネットの持越解除を行います。
pause	ジョブネットを停止中の状態にします。
continue	ジョブネットの“停止”を行う前の状態にします。
disable	ジョブネットを無効状態にします。
enable	ジョブネットの“無効”を行う前の状態にします。
rerun	異常終了、強制終了または確認済にしたジョブネットを再起動します。

レスポンス

ステータスコード

以下のHTTPステータスコードを返却します。

コード	説明
202	リクエストの処理が正常終了しました。
400	リクエストパラメータに誤りがあります。 ・ 必須パラメータを指定していない ・ 指定不要のパラメータを指定している ・ パラメータの指定値に誤りがある
401	認証情報の指定に誤りがあります。
403	対象のプロジェクトへの権限がありません。
404	対象のジョブネットは存在しません。 ・ システムに対象のジョブネットが存在しない ・ パスパラメータの指定内容が誤っている
409	対象のジョブネットへの指定の操作はできません。 ・ 指定の操作ができるジョブネットの状態ではない
500	リクエストの処理中にサーバで異常が発生しました。

レスポンスボディ

指定したjobnetsリソースのオブジェクトを返却します。

注意

- Web APIによるジョブネットへの操作は、Systemwalker Operation Managerサーバの処理と同期を取っていません。そのため、Web APIのレスポンスはサーバでの処理の完了を保証するものではありません。レスポンスボディで返却するオブジェクトは、操作が反映されていない場合があります。
- 複写元ジョブネットの起動操作をした場合、複写元ジョブネットの情報を返却します。複写先ジョブネットの情報は、ジョブネット情報の一覧取得から取得してください。

使用例

サブシステム番号0、プロジェクト名“SampleProject”、ジョブネット名“SampleJobnet”のジョブネットを可変パラメタ“SampleParam”で起動する場合のAPIの使用例を次に示します。

リクエスト例

```
POST /Systemwalker-omgr/api/v1/subsystems/0/projects/SampleProject/jobnets/SampleJobnet HTTP/1.1
Host: Systemwalker Operation Managerサーバのホスト名:9900
Authorization: Basic ********************
Content-Type: application/json

{
  "control": "start",
  "vparam": "SampleParam"
}
```

レスポンス例

“18.3.2.1 ジョブネット情報の取得”のレスポンス例を参照してください。

18.3.2.3　ジョブネット情報の一覧取得

機能説明

jobnetsリソースの一覧を取得します。

対象は、認証情報に指定したユーザが参照権以上を持つプロジェクトに存在するジョブネットとなります。

リクエスト

HTTPメソッド

GET

リクエストURI

/v1/jobnets

リクエストパラメータ

共通仕様のクエリパラメータが使用できます。

詳細については、“18.2.3.2 リクエストの形式”を参照してください。

レスポンス

ステータスコード

以下のHTTPステータスコードを返却します。

コード	説明
200	リクエストの処理が正常終了しました。
400	リクエストパラメータに誤りがあります。 ・ 指定不要のパラメータを指定している ・ 同じクエリパラメータを複数指定している
401	認証情報の指定に誤りがあります。
500	リクエストの処理中にサーバで異常が発生しました。

レスポンスボディ

jobnetsリソースのオブジェクトを返却します。

使用例

サブシステム0に存在するジョブネットの一覧を取得する場合の、APIの使用例を次に示します。

リクエスト例

```
GET /Systemwalker-omgr/api/v1/jobnets?subsystem=0 HTTP/1.1
Host: Systemwalker Operation Managerサーバのホスト名:9900
Authorization: Basic ********************
```

レスポンス例

dataフィールドのオブジェクト配列として、jobnetsリソースのオブジェクトを返却します。jobnetsリソースのオブジェクトについては、“18.3.2.1 ジョブネット情報の取得”のレスポンス例を参照してください。

18.3.3　jobsリソース

Systemwalker Operation Managerに登録済みのジョブを操作します。

以下の操作が可能です。

エンドポイント (/Systemwalker-omgr/api 以降)	HTTP メソッド	操作内容
/v1/projects/{projectname}/ jobnets/{jobnetname}/jobs/ {jobnumber}	GET	指定したジョブの情報を取得します。
	POST	指定したジョブを操作します。
/v1/jobs	GET	ジョブ情報の一覧を取得します。

jobsリソースのオブジェクトのフィールドは、以下のとおりです。

なお、システムで一意のプロジェクトを指定する場合、SE版では、プロジェクト名とジョブネット名、ジョブ番号を指定します。EE版およびGEE版では、サブシステム番号とプロジェクト名、ジョブネット名、ジョブ番号を指定します。

フィールド名	型	説明
host	文字列	接続先のホスト名
subsystem	数値	サブシステム番号
projectname	文字列	プロジェクト名
jobnetname	文字列	ジョブネット名
jobnumber	数値	ジョブ番号
jobname_jes	文字列	ジョブ実行制御でのジョブ名
jobiconex	文字列	アイコン種別の説明文
jobicon	リスト	アイコン種別 返却値は“4.12.2 ジョブ制御文”のアイコン種別を参照してください。
jobname	文字列	アイコン種別によって返却値が異なります。 JNET：子ジョブネットのジョブネット名 LNET：リンクジョブネットのジョブネット名 BJOB：バッチジョブ定義名 INTS：ワークユニット名 上記以外：ジョブとして起動するバッチプログラム名、コマンド名、またはプログラム名
jobparam	文字列	ジョブのパラメータ
directory	文字列	ディレクトリ
jobcomment	文字列	アイコン種別によって返却値が異なります。 JNET：子ジョブネットのジョブネット名称 LNET：リンクジョブネットのジョブネット名称 上記以外：ジョブへのコメント
req_host	文字列	依頼ホスト名 複数指定されている場合は、,(カンマ)区切りで返却します。
status	リスト	ジョブの状態(注1)
starttime	日時	前回の開始日時
endtime	日時	前回の終了日時
endcode	数値	終了コード 未実行のジョブの終了コードは0です。

注1)

statusフィールドの返却値は以下のとおりです。

フィールド値	ジョブの状態
Abend	異常終了
Forced	強制終了
Pause	停止中
Execute	実行中
Wait	実行待ち
Pseudo	疑似正常
Normal	正常終了
Disable	無効状態
Delay	実行遅延
Warning	警告
Pass & W	パス状態(実行待ち)
Pass & P	パス状態(停止中)
Pass & D	パス状態(無効状態)

18.3.3.1 ジョブ情報の取得

機能説明

jobsリソースを指定したGETメソッドのリクエストを送ることで、対象のジョブ情報を取得します。

実行に必要な権限/実行環境

指定のジョブが存在するプロジェクトに対して、参照権以上の権限が必要です。

リクエスト

HTTPメソッド

GET

リクエストURI

/v1/subsystems/{subsystem}/projects/{projectname}/jobnets/{jobnetname}/jobs/{jobnumber}

/v1/projects/{projectname}/jobnets/{jobnetname}/jobs/{jobnumber} (サブシステム番号を省略する場合)

リクエストパラメータ

パスパラメータで、以下のパラメータを指定します。

パラメータ	指定方法	型	説明
subsystem	パス	数値	対象のジョブが存在するサブシステム番号(0～9)を指定します。サブシステム番号を省略する場合は0になります。
projectname(*)	パス	文字列	対象のジョブが存在するプロジェクト名を指定します。
jobnetname(*)	パス	文字列	対象のジョブが存在するジョブネット名を指定します。
jobnumber(*)	パス	数値	対象のジョブ番号(注1)を指定します。

(*)：必須のパラメータ

パス：パスパラメータ

注1)

ジョブ番号は、ジョブ制御文で指定した番号です。“18.3.3.3 ジョブ情報の一覧取得”のWeb APIで取得し、そのジョブネットに登録されているジョブのジョブ番号を確認できます。

レスポンス

ステータスコード

以下のHTTPステータスコードを返却します。

コード	説明
200	リクエストの処理が正常終了しました。
401	認証情報の指定に誤りがあります。
404	対象のジョブは存在しません。 ・ システムに対象のジョブが存在しない ・ パスパラメータの指定内容が誤っている
500	リクエストの処理中にサーバで異常が発生しました。

レスポンスボディ

対象のjobsリソースのオブジェクトを返却します。

使用例

サブシステム番号0、プロジェクト名が“SampleProject”、ジョブネット名が“SampleJobnet”、ジョブ番号が1のジョブを取得する場合のAPIの使用例を以下に示します。なお、レスポンス例は、dataフィールドに格納されるjobsリソースのオブジェクトを示しており、レスポンスヘッダおよび他のフィールドは省略しています。

リクエスト例

```
GET /Systemwalker-omgr/api/v1/subsystems/0/projects/SampleProject/jobnets/SampleJobnet/jobs/1 HTTP/1.1
Host: Systemwalker Operation Managerサーバのホスト名:9900
Authorization: Basic **********
```

レスポンス例

```
{
  "host": "Systemwalker Operation Managerサーバのホスト名",
  "subsystem": 0,
  "projectname": "SampleProject",
  "jobnetname": "SampleJobnet",
  "jobnumber": 1,
  "jobname_jes": "SampleJob",
  "jobiconex": "Execute command",
  "jobicon": "CMND",
  "jobname": "SampleJob",
  "jobparam": "",
  "directory": "",
  "jobcomment": "コマンド実行",
  "req_host": "",
  "status": "Normal",
  "starttime": "2019/12/31 12:00:00",
  "endtime": "2019/12/31 12:00:01",
  "endcode": 0
}
```

18.3.3.2 ジョブの操作

機能説明

jobsリソースを指定したPOSTメソッドのリクエストを送ることで、対象のジョブを操作します。

実行に必要な権限/実行環境

指定のジョブが存在するプロジェクトに対して、操作権以上の権限が必要です。

リクエスト

HTTPメソッド

POST

リクエストURI

/v1/subsystems/{subsystem}/projects/{projectname}/jobnets/{jobnetname}/jobs/{jobnumber}

/v1/projects/{projectname}/jobnets/{jobnetname}/jobs/{jobnumber} (サブシステム番号を省略する場合)

リクエストパラメータ

パスパラメータおよびリクエストボディで、以下のパラメータを指定します。

パラメータ	指定方法	型	説明
subsystem	パス	数値	対象のジョブが存在するサブシステム番号(0～9)を指定します。サブシステム番号を省略する場合は0になります。
projectname(*)	パス	文字列	対象のジョブが存在するプロジェクト名を指定します。
jobnetname(*)	パス	文字列	対象のジョブが存在するジョブネット名を指定します。
jobnumber(*)	パス	数値	対象のジョブ番号(注1)を指定します。
control(*)	ボディ	リスト	ジョブへの操作種別を指定します。(注2)

(*)：必須のパラメータ

パス：パスパラメータ

ボディ：リクエストボディ

注1)

ジョブ番号は、ジョブ制御文で指定した番号です。“18.3.3.3 ジョブ情報の一覧取得”のWeb APIで任意のジョブネットに絞り込むことで、そのジョブネットに登録されているジョブのジョブ番号を確認できます。

注2)

メッセージボディのcontrolパラメータで指定可能な値は、以下のとおりです。

指定値	説明
cancel	実行中のジョブや実行遅延のジョブを強制的に終了します。
pause	実行待ちのジョブや終了したジョブを停止状態にします。
continue	停止中のジョブを停止解除します。
disable	実行待ち/停止中/終了したジョブを無効状態にします。
enable	無効状態のジョブを無効解除します。
rerun	異常終了または強制終了したジョブを再起動します。

レスポンス

ステータスコード

以下のHTTPステータスコードを返却します。

コード	説明
202	リクエストの処理が正常終了しました。
400	リクエストパラメータに誤りがあります。 ・ 必須パラメータを指定していない ・ 指定不要のパラメータを指定している ・ パラメータの指定値に誤りがある
401	認証情報の指定に誤りがあります。
403	対象のプロジェクトへの権限がありません。
404	対象のジョブは存在しません。 ・ システムに対象のジョブが存在しない ・ パスパラメータの指定内容が誤っている
409	対象のジョブへの指定の操作はできません。 ・ 指定の操作ができるジョブの状態ではない
500	リクエストの処理中にサーバで異常が発生しました。

レスポンスボディ

指定したjobsリソースのオブジェクトを返却します。

注意

Web APIによるジョブネットへの操作は、Systemwalker Operation Managerサーバの処理と同期を取っていません。そのため、Web APIのレスポンスはサーバでの処理の完了を保証するものではありません。レスポンスボディで返却するオブジェクトは、操作が反映されていない場合があります。

使用例

サブシステム番号0、プロジェクト名が“SampleProject、ジョブネット名が“SampleJobnet”、ジョブ番号が5のジョブを再起動する場合のAPIの使用例を次に示します。

リクエスト例

```
POST /Systemwalker-omgr/api/v1/subsystems/0/projects/SampleProject/jobnets/SampleJobnet/jobs/5 HTTP/1.1
Host: Systemwalker Operation Managerサーバのホスト名:9900
Authorization: Basic **********
Content-Type: application/json

{
  ”control”: ”rerun”
}
```

レスポンス例

“18.3.3.1 ジョブ情報の取得”のレスポンス例を参照してください。

18.3.3.3 ジョブ情報の一覧取得

機能説明

jobsリソースの一覧を取得します。

対象は、認証情報に指定したユーザが参照権以上を持つプロジェクトに存在するジョブネットのジョブとなります。

リクエスト

HTTPメソッド

GET

リクエストURI

/v1/jobs

リクエストパラメータ

共通仕様のクエリパラメータが使用できます。

詳細については、“18.2.3.2 リクエストの形式”を参照してください。

レスポンス

ステータスコード

以下のHTTPステータスコードを返却します。

コード	説明
200	リクエストの処理が正常終了しました。
400	リクエストパラメータに誤りがあります。 ・ 指定不要のパラメータを指定している ・ 同じクエリパラメータを複数指定している
401	認証情報の指定に誤りがあります。
500	リクエストの処理中にサーバで異常が発生しました。

レスポンスボディ

jobsリソースのオブジェクトを返却します。

使用例

サブシステム0に登録されたジョブの一覧を取得する場合のAPIの使用例を次に示します。

リクエスト例

```
GET /Systemwalker-omgr/api/v1/jobs?subsystem=0 HTTP/1.1
Host: Systemwalker Operation Managerサーバのホスト名:9900
Authorization: Basic **********
```

レスポンス例

dataフィールドのオブジェクト配列として、jobsリソースのオブジェクトを返却します。jobsリソースのオブジェクトについては、“18.3.3.1 ジョブ情報の取得”のレスポンス例を参照してください。

18.3.4 queuesリソース

Systemwalker Operation Managerに登録済みのキューを操作します。

以下の操作が可能です。

エンドポイント (/Systemwalker-omgr/api 以降)	HTTP メソッド	操作内容
/v1/queues/{queuename}	GET	指定したキューの情報を取得します。

エンドポイント (/Systemwalker-omgr/api 以降)	HTTP メソッド	操作内容
	POST	指定したキューを操作します。
	PUT	指定したキューの定義を一時変更します。
/v1/queues	GET	プロジェクト情報の一覧を取得します。

queuesリソースのオブジェクトのフィールドは、以下のとおりです。

なお、システムで一意のキューを指定する場合、SE版では、キュー名を指定します。EE版およびGEE版では、サブシステム番号とキュー名を指定します。

フィールド名	型	説明
host	文字列	接続先のホスト名
subsystem	数値	サブシステム番号
queuename	文字列	キュー名
status	リスト	キューの状態 ACT：キューは動作中です INACT：キューは停止中です
maxexec	数値	多重度 無制限の場合、-1を返却します。
run	数値	実行中ジョブ数
que	数値	実行待ちジョブ数
hold	数値	実行保留ジョブ数
qprty	数値	実行優先順位
dprty	数値	実行優先度
maxjob	数値	投入可能ジョブ数 無制限の場合、-1を返却します。
elapse	数値	経過時間制限 無制限の場合、0を返却します。
hostgroup	文字列	ホストグループ名 指定されていない場合、“*”を返却します。

18.3.4.1 キュー情報の取得

機能説明

queuesリソースを指定したGETメソッドのリクエストを送ることで、対象のキュー情報を取得します。

実行に必要な権限/実行環境

Systemwalker Operation Manager環境設定でSystemwalker Operation Manager利用者の限定をしている場合、swadminグループに所属するユーザの権限が必要です。Systemwalker Operation Manager利用者の限定をしていない場合は、一般ユーザの権限で使用できます。

リクエスト

HTTPメソッド

GET

リクエストURI

/v1/subsystems/{subsystem}/queues/{queuename}

/v1/queues/{queuename} (サブシステム番号を省略する場合)

リクエストパラメータ

パスパラメータで、以下のパラメータを指定します。

パラメータ	指定方法	型	説明
subsystem	パス	数値	対象のキューが存在するサブシステム番号(0～9)を指定します。サブシステム番号を省略する場合は0になります。
queuename(*)	パス	文字列	対象のキュー名を指定します。

(*)：必須のパラメータ

パス：パスパラメータ

レスポンス

ステータスコード

以下のHTTPステータスコードを返却します。

コード	説明
200	リクエストの処理が正常終了しました。
401	認証情報の指定に誤りがあります。
403	swadminグループに所属するユーザではありません。
404	対象のキューは存在しません。 ・ システムに対象のキューが存在しない ・ パスパラメータの指定内容が誤っている
500	リクエストの処理中にサーバで異常が発生しました。

レスポンスボディ

対象のqueuesリソースのオブジェクトを返却します。

使用例

サブシステム番号0、キュー名が“queue1”のキューを取得する場合のAPIの使用例を以下に示します。なお、レスポンス例は、dataフィールドに格納されるqueuesリソースのオブジェクトを示しており、レスポンスヘッダおよび他のフィールドは省略しています。

リクエスト例

```
GET /Systemwalker-omgr/api/v1/subsystems/0/queues/queue1 HTTP/1.1
Host: Systemwalker Operation Managerサーバのホスト名:9900
Authorization: Basic ********************
```

レスポンス例

```
{
  "host": "Systemwalker Operation Managerサーバのホスト名",
  "subsystem": 0,
  "queuename": "queue1",
  "status": "ACT",
  "maxexec": 256,
  "run": 0,
  "que": 0,
  "hold": 0,
```

```
  "qprty": 31,
  "dprty": 2,
  "maxjob": -1,
  "elapse": 0,
  "hostgroup": "hostgroup1"
}
```

18.3.4.2　キューの定義の一時変更

機能説明

queuesリソースを指定したPUTメソッドのリクエストを送ることで、対象のキューの定義を一時変更します。

実行に必要な権限/実行環境

システム管理者(Administratorsグループに属するユーザ/スーパーユーザ)権限が必要です。

リクエスト

HTTPメソッド

PUT

リクエストURI

/v1/subsystems/{subsystem}/queues/{queuename}

/v1/queues/{queuename}（サブシステム番号を省略する場合)

リクエストパラメータ

パスパラメータおよびリクエストボディで、以下のパラメータを指定します。

パラメータ	指定方法	型	説明
subsystem	パス	数値	対象のキューが存在するサブシステム番号(0～9)を指定します。サブシステム番号を省略する場合は0になります。
queuename(*)	パス	文字列	対象のキュー名を指定します。
maxexec	ボディ	数値	多重度(同時実行可能数)をします。範囲は1～999です。
qprty	ボディ	数値	実行優先順位をします。範囲は0～63です。
dprty	ボディ	数値	実行優先度をします。範囲は、Windows版の場合は0～4、Linux版の場合は-20～19です。
maxjob	ボディ	数値	投入可能ジョブ数を指定します。範囲は1～999です。
elapse	ボディ	数値	経過時間制限値をします。範囲は1～99999999です。

(*)：必須のパラメータ

パス：パスパラメータ

ボディ：リクエストボディ

レスポンス

ステータスコード

以下のHTTPステータスコードを返却します。

コード	説明
200	リクエストの処理が正常終了しました。
400	リクエストパラメータに誤りがあります。 ・ 必須パラメータを指定していない ・ 指定不要のパラメータを指定している ・ パラメータの指定値に誤りがある
401	認証情報の指定に誤りがあります。
403	システム管理者の権限ではありません。
404	対象のキューは存在しません。 ・ システムに対象のキューが存在しない ・ パスパラメータの指定内容が誤っている
500	リクエストの処理中にサーバで異常が発生しました。

レスポンスボディ

指定したqueuesリソースのオブジェクトを返却します。

注意事項

- キューの属性変更により変更した内容は、ジョブ実行制御を初期化モードで起動するまでの間、保持されます。ジョブ実行制御を初期化モードで起動すると、分散実行の設定は、初期化ファイルに定義した内容に変更されます。
- リクエストボディには、少なくとも1つパラメータを指定してください。

使用例

サブシステム番号0、キュー名が“queue1”のキューの多重度を99に、投入可能ジョブ数を200にする場合のAPIの使用例を以下に示します。

リクエスト例

```
PUT /Systemwalker-omgr/api/v1/subsystems/0/queues/queue1 HTTP/1.1
Host: Systemwalker Operation Managerサーバのホスト名:9900
Authorization: Basic ********************
Content-Type: application/json

{
  "maxexec": 99,
  "maxjob": 200
}
```

レスポンス例

多重度が99に、投入可能ジョブ数が200に変更されたqueuesリソースのオブジェクトが返ります。“18.3.4.1 キュー情報の取得”のレスポンス例を参照してください。

18.3.4.3 キューの操作

機能説明

queuesリソースを指定したPOSTメソッドのリクエストを送ることで、対象のキューを操作(停止/開始)します。

実行に必要な権限/実行環境

システム管理者(Administratorsグループに属するユーザ/スーパーユーザ)権限が必要です。

リクエスト

HTTPメソッド

POST

リクエストURI

/v1/subsystems/{subsystem}/queues/{queuename}

/v1/queues/{queuename}（サブシステム番号を省略する場合)

リクエストパラメータ

パスパラメータおよびリクエストボディで、以下のパラメータを指定します。

パラメータ	指定方法	型	説明
subsystem	パス	数値	対象のキューが存在するサブシステム番号(0～9)を指定します。サブシステム番号を省略する場合は0になります。
queuename(*)	パス	文字列	対象のキュー名を指定します。
control(*)	ボディ	リスト	キューへの操作種別を指定します。 start: キューを起動します stop: キューを停止します

(*)：必須のパラメータ

パス：パスパラメータ

ボディ：リクエストボディ

レスポンス

以下のHTTPステータスコードを返却します。

コード	説明
200	リクエストの処理が正常終了しました。
400	リクエストパラメータに誤りがあります。 ・ 必須パラメータを指定していない ・ 指定不要のパラメータを指定している ・ パラメータの指定値に誤りがある
401	認証情報の指定に誤りがあります。
403	システム管理者の権限ではありません。
404	対象のキューは存在しません。 ・ システムに対象のキューが存在しない ・ パスパラメータの指定内容が誤っている
500	リクエストの処理中にサーバで異常が発生しました。

レスポンスボディ

指定したqueuesリソースのオブジェクトを返却します。

使用例

サブシステム番号0、キュー名が“queue2”のキューを停止する場合のAPIの使用例を以下に示します。

リクエスト例

```
POST /Systemwalker-omgr/api/v1/subsystems/0/queues/queue2 HTTP/1.1
Host: Systemwalker Operation Managerサーバのホスト名:9900
Authorization: Basic ********************
Content-Type: application/json

{
  ”control”: ”stop”
}
```

レスポンス例

“18.3.4.1 キュー情報の取得”のレスポンス例を参照してください。

18.3.4.4　キュー情報の一覧取得

機能説明

queuesリソースの一覧を取得します。

実行に必要な権限/実行環境

Systemwalker Operation Manager環境設定でSystemwalker Operation Manager利用者の限定をしている場合、swadminグループに所属するユーザの権限が必要です。Systemwalker Operation Manager利用者の限定をしていない場合は、一般ユーザの権限で使用できます。

リクエスト

HTTPメソッド

GET

リクエストURI

/v1/queues

リクエストパラメータ

共通仕様のクエリパラメータが使用できます。

詳細については、“18.2.3.2 リクエストの形式”を参照してください。

レスポンス

ステータスコード

以下のHTTPステータスコードを返却します。

コード	説明
200	リクエストの処理が正常終了しました。
400	リクエストパラメータに誤りがあります。 ・ 指定不要のパラメータを指定している ・ 同じクエリパラメータを複数指定している
401	認証情報の指定に誤りがあります。

コード	説明
403	swadminグループに所属するユーザの権限ではありません。
500	リクエストの処理中にサーバで異常が発生しました。

レスポンスボディ

queuesリソースのオブジェクトを返却します。

使用例

サブシステム番号0のキューの一覧を取得する場合の、APIの使用例を次に示します。

リクエスト例

```
GET /Systemwalker-omgr/api/v1/queues?subsystem=0 HTTP/1.1
Host: Systemwalker Operation Managerサーバのホスト名:9900
Authorization: Basic ********************
```

レスポンス例

dataフィールドのオブジェクト配列として、queuesリソースのオブジェクトを返却します。queuesリソースのオブジェクトについては、“18.3.4.1 キュー情報の取得”のレスポンス例を参照してください。

18.3.5 contained-hostsリソース

Systemwalker Operation Managerに登録済みのホストグループに対して、構成ホストを操作します。

以下の操作が可能です。

エンドポイント (/Systemwalker-omgr/api 以降)	HTTP メソッド	操作内容
/v1/host-groups/{hostgroupname}/contained-hosts/{hostname}	GET	指定した構成ホストの情報を取得します。
	POST	指定した構成ホストの属性を変更します。
	DELETE	指定した構成ホストをホストグループから削除します。
/v1/contained-hosts	POST	ホストグループに構成ホストを追加します。
	GET	構成ホストの情報の一覧を取得します。

contained-hostsリソースのオブジェクトのフィールドは、以下のとおりです。

なお、システムで一意の構成ホストを指定する場合、SE版では、ホストグループ名と構成ホストのホスト名を指定します。EE版およびGEE版では、サブシステム番号とホストグループ、構成ホストのホスト名を指定します。

フィールド名	型	説明
host	文字列	接続先のホスト名
subsystem	数値	サブシステム番号
hostgroupname	文字列	ホストグループ名
hostname	文字列	構成ホストのホスト名
status	リスト	構成ホストの状態 ACT：構成ホストがダウンしていない DOWN：構成ホストがダウンしている NA：構成ホストの状態が取得できない
maxexec	数値	構成ホストの最大多重度
run	数値	構成ホストで実行中のジョブ数

18.3.5.1　構成ホスト情報の取得

機能説明

contained-hostsリソースを指定したGETメソッドのリクエストを送ることで、対象の構成ホスト情報を取得します。

実行に必要な権限/実行環境

Systemwalker Operation Manager環境設定でSystemwalker Operation Manager利用者の限定をしている場合、swadminグループに所属するユーザの権限が必要です。Systemwalker Operation Manager利用者の限定をしていない場合は、一般ユーザの権限で使用できます。

リクエスト

HTTPメソッド

GET

リクエストURI

/v1/subsystems/{subsystem}/host-groups/{hostgroupname}/contained-hosts/{hostname}

/v1/host-groups/{hostgroupname}/contained-hosts/{hostname}（サブシステム番号を省略する場合）

リクエストパラメータ

パスパラメータで、以下のパラメータを指定します。

パラメータ	指定方法	型	説明
subsystem	パス	数値	対象の構成ホストが存在するサブシステム番号(0～9)を指定します。サブシステム番号を省略する場合は0になります。
hostgroupname (*)	パス	文字列	対象の構成ホストが存在するホストグループを指定します。(注1)
hostname(*)	パス	文字列	対象の構成ホスト名を指定します。

(*)：必須のパラメータ

パス：パスパラメータ

注1)

hostgroupnameには、いずれかのキューに登録されたホストグループを指定できます。“18.3.4.4 キュー情報の一覧取得”のWeb APIでqueuesリソースの一覧を取得し、そのhostgroupフィールド値からホストグループを確認できます。

レスポンス

以下のHTTPステータスコードを返却します。

コード	説明
200	リクエストの処理が正常終了しました。
401	認証情報の指定に誤りがあります。
403	swadminグループに所属するユーザの権限ではありません。
404	対象の構成ホストは存在しません。 ・ システムに対象の構成ホストが存在しない ・ パスパラメータの指定内容が誤っている

コード	説明
500	リクエストの処理中にサーバで異常が発生しました。

レスポンスボディ

対象のcontained-hostsリソースのオブジェクトを返却します。

使用例

サブシステム番号0、ホストグループ名が“hostgroup1”、構成ホスト名が“host1”の構成ホストを取得する場合のAPIの使用例を以下に示します。なお、レスポンス例は、dataフィールドに格納されるcontained-hostsリソースのオブジェクトを示しており、レスポンスヘッダおよび他のフィールドは省略しています。

リクエスト例

```
GET /Systemwalker-omgr/api/v1/subsystems/0/host-groups/hostgroup1/contained-hosts/host1  HTTP/1.1
Host: Systemwalker Operation Managerサーバのホスト名:9900
Authorization: Basic ********************
```

レスポンス例

```
{
  ”host”: ”Systemwalker Operation Managerサーバのホスト名”,
  ”subsystem”: 0,
  ”hostgroupname”: ”hostgroup1”,
  ”hostname”: ”host1”,
  ”status”: ”ACT”,
  ”maxexec”: 10,
  ”run”: 0
}
```

18.3.5.2　構成ホストの定義の一時変更

機能説明

contained-hostsリソースを指定したPUTメソッドのリクエストを送ることで、対象の構成ホストの定義を一時変更します。

実行に必要な権限/実行環境

システム管理者(Administratorsグループに属するユーザ/スーパーユーザ)権限が必要です。

リクエスト

HTTPメソッド

PUT

リクエストURI

/v1/subsystems/{subsystem}/host-groups/{hostgroupname}/contained-hosts/{hostname}

/v1/host-groups/{hostgroupname}/contained-hosts/{hostname} (サブシステム番号を省略する場合)

リクエストパラメータ

パスパラメータおよびリクエストボディで、以下のパラメータを指定します。

パラメータ	指定方法	型	説明
subsystem	パス	数値	対象の構成ホストが存在するサブシステム番号(0～9)を指定します。サブシステム番号を省略する場合は0になります。

パラメータ	指定方法	型	説明
hostgroupname(*)	パス	文字列	対象の構成ホストが存在するホストグループを指定します。
hostname(*)	パス	文字列	対象の構成ホスト名を指定します。
maxexec	ボディ	数値	対象の構成ホストのジョブの多重度（同時実行可能数）を指定します。範囲は1～999です。

(*)：必須のパラメータ

パス：パスパラメータ

ボディ：リクエストボディ

レスポンス

ステータスコード

以下のHTTPステータスコードを返却します。

コード	説明
200	リクエストの処理が正常終了しました。
400	リクエストパラメータに誤りがあります。 ・ 必須パラメータを指定していない ・ 指定不要のパラメータを指定している ・ パラメータの指定値に誤りがある
401	認証情報の指定に誤りがあります。
403	システム管理者の権限ではありません。
404	対象の構成ホストは存在しません。 ・ システムに対象の構成ホストが存在しない ・ パスパラメータの指定内容が誤っている
500	リクエストの処理中にサーバで異常が発生しました。

レスポンスボディ

指定したcontained-hostsリソースのオブジェクトを返却します。

注意事項

構成ホストの属性変更により変更した内容は、ジョブ実行制御を初期化モードで起動するまでの間、保持されます。
ジョブ実行制御を初期化モードで起動すると、分散実行の設定は、初期化ファイルに定義した内容に変更されます。

使用例

サブシステム番号0、ホストグループ名が“hostgroup1”、ホスト名が“host1”の構成ホストのジョブの多重度(同時実行可能数)を2に変更する場合のAPIの使用例を以下に示します。

リクエスト例

```
PUT /Systemwalker-omgr/api/v1/subsystems/0/host-groups/hostgroup1/contained-hosts/host1  HTTP/1.1
Host: Systemwalker Operation Managerサーバのホスト名:9900
Authorization: Basic ********************
Content-Type: application/json

{
```

```
  "maxexec": 2
}
```

レスポンス例

ジョブの多重度(同時実行可能数)が2に変更されたcontained-hostsリソースのオブジェクトが返ります。
“18.3.5.1 構成ホスト情報の取得”のレスポンス例を参照してください。

18.3.5.3 構成ホストの削除

機能説明

contained-hostsリソースを指定したDELETEメソッドのリクエストを送ることで、対象の構成ホストをホストグループから削除します。

実行に必要な権限/実行環境

システム管理者(Administratorsグループに属するユーザ/スーパーユーザ)権限が必要です。

リクエスト

HTTPメソッド

DELETE

リクエストURI

/v1/subsystems/{subsystem}/host-groups/{hostgroupname}/contained-hosts/{hostname}

/v1/host-groups/{hostgroupname}/contained-hosts/{hostname}（サブシステム番号を省略する場合)

リクエストパラメータ

パスパラメータで、以下のパラメータを指定します。

パラメータ	指定方法	型	説明
subsystem	パス	数値	対象の構成ホストが存在するサブシステム番号(0～9)を指定します。サブシステム番号を省略する場合は0になります。
hostgroupname(*)	パス	文字列	対象の構成ホストが存在するホストグループを指定します。
hostname(*)	パス	文字列	対象の構成ホスト名を指定します。

(*)：必須のパラメータ

パス：パスパラメータ

レスポンス

ステータスコード

以下のHTTPステータスコードを返却します。

コード	説明
200	リクエストの処理が正常終了しました。
401	認証情報の指定に誤りがあります。
403	システム管理者の権限ではありません。
404	対象の構成ホストは存在しません。 ・ システムに対象の構成ホストが存在しない

コード	説明
	・ パスパラメータの指定内容が誤っている
409	対象の構成ホストへの指定の操作はできません。 ・ ホストグループの構成ホストの登録数が1つしかない
500	リクエストの処理中にサーバで異常が発生しました。

レスポンスボディ

dataフィールドが空のcontained-hostsリソースのオブジェクトを返します。

注意事項

ホストグループに登録する構成ホストの最小数は1です。そのため、対象となるホストグループの構成ホストの登録数が1の場合、削除により構成ホストの登録数が0になるため、削除に失敗します。

使用例

サブシステム番号0、ホストグループ名が“hostgroup1”のホストグループから、ホスト名が“host1”の構成ホストを削除する場合のAPIの使用例を以下に示します。

リクエスト例

```
DELETE /Systemwalker-omgr/api/v1/subsystems/0/host-groups/hostgroup1/contained-hosts/host1 HTTP/1.1
Host: Systemwalker Operation Managerサーバのホスト名:9900
Authorization: Basic ********************
```

レスポンス例

```
{
  ”next”: false,
  ”total”: 0,
  ”data”: []
}
```

18.3.5.4　構成ホストの追加

機能説明

リクエストボディに構成ホスト情報を指定したPOSTメソッドのリクエストを送ることで、対象のホストグループに構成ホストを追加します。

なお、対象の構成ホストがすでに存在していた場合は、上書き更新します。

実行に必要な権限/実行環境

システム管理者(Administratorsグループに属するユーザ/スーパーユーザ)権限が必要です。

リクエスト

HTTPメソッド

POST

リクエストURI

/v1/contained-hosts

リクエストパラメータ

リクエストボディで、以下のパラメータを指定します。

パラメータ	指定方法	型	説明
subsystem	ボディ	数値	対象の構成ホストが存在するサブシステム番号(0～9)を指定します。サブシステム番号を省略する場合は0になります。
hostgroupname(*)	ボディ	文字列	対象の構成ホストが存在するホストグループ名を指定します。
hostname(*)	ボディ	文字列	対象の構成ホスト名を指定します。 ・ ホスト名に自ホストを指定しないでください。 ・ 実行サブシステム名は使用できません。 Web APIから指定する場合、半角英数字および“-”(半角ハイフン)、“_”(半角アンダーバー)、“:”(半角コロン)、“.”(半角ドット)、“%”(半角パーセント)のみ使用できます。
maxexec	ボディ	数値	対象の構成ホストのジョブの多重度(同時実行可能数)を指定します。 指定可能な値は1～999です。省略値は10です。

(*)：必須のパラメータ

ボディ：リクエストボディ

レスポンス

ステータスコード

以下のHTTPステータスコードを返却します。

コード	説明
200	リクエストの処理が正常終了しました。 (構成ホストの上書き更新)
201	リクエストの処理が正常終了しました。 (構成ホストを追加)
400	リクエストパラメータに誤りがあります。 ・ 必須パラメータを指定していない ・ 指定不要のパラメータを指定している ・ パラメータの指定値に誤りがある
401	認証情報の指定に誤りがあります。
403	システム管理者の権限ではありません。
409	対象のホストグループに構成ホストは追加できません。 ・ ホストグループの構成ホストの登録数が100である ・ 対象のホストグループが存在しない
500	リクエストの処理中にサーバで異常が発生しました。

レスポンスボディ

追加したcontained-hostsリソースのオブジェクトを返却します。

注意事項

- 構成ホストの属性変更により変更した内容は、ジョブ実行制御を初期化モードで起動するまでの間、保持されます。ジョブ実行制御を初期化モードで起動すると、分散実行の設定は、初期化ファイルに定義した内容に変更されます。
- 1つのホストグループ内に登録できる構成ホストは100個までです。そのため、対象となるホストグループの構成ホストの登録数が100の場合、追加に失敗します。

使用例

サブシステム番号0、ホストグループ名が“hostgroup1”のホストグループに、構成ホスト名が“host2”、多重度(同時実行可能数)が4の構成ホストを追加する場合のAPIの使用例を以下に示します。

リクエスト例

```
POST /Systemwalker-omgr/api/v1/contained-hosts HTTP/1.1
Host: Systemwalker Operation Managerサーバのホスト名:9900
Authorization: Basic **********
Content-Type: application/json

{
  "subsystem": 0,
  "hostgroupname": "hostgroup1",
  "hostname": "host2",
  "maxexec": 4
}
```

レスポンス例

“18.3.5.1 構成ホスト情報の取得”のレスポンス例を参照してください。

18.3.5.5　構成ホスト情報の一覧取得

機能説明

contained-hostsリソースの一覧を取得します。

実行に必要な権限/実行環境

Systemwalker Operation Manager環境設定でSystemwalker Operation Manager利用者の限定をしている場合、swadminグループに所属するユーザの権限が必要です。Systemwalker Operation Manager利用者の限定をしていない場合は、一般ユーザの権限で使用できます。

リクエスト

HTTPメソッド

GET

リクエストURI

/v1/contained-hosts

リクエストパラメータ

共通仕様のクエリパラメータが使用できます。

詳細については、“18.2.3.2 リクエストの形式”を参照してください。

レスポンス

ステータスコード

以下のHTTPステータスコードを返却します。

コード	説明
200	リクエストの処理が正常終了しました。
400	リクエストパラメータに誤りがあります。 ・ 指定不要のパラメータを指定している ・ 同じクエリパラメータを複数指定している
401	認証情報の指定に誤りがあります。
403	swadminグループに所属するユーザの権限ではありません。
500	リクエストの処理中にサーバで異常が発生しました。

レスポンスボディ

contained-hostsリソースのオブジェクトを返却します。

使用例

サブシステム番号0に存在する構成ホストの一覧を取得する場合の、APIの使用例を次に示します。

リクエスト例

```
GET /Systemwalker-omgr/api/v1/contained-hosts?subsystem=0 HTTP/1.1
Host: Systemwalker Operation Managerサーバのホスト名:9900
Authorization: Basic ********************
```

レスポンス例

dataフィールドのオブジェクト配列として、contained-hostsリソースのオブジェクトを返却します。
contained-hostsリソースのオブジェクトについては、“18.3.5.1 構成ホスト情報の取得”のレスポンス例を参照してください。

第19章　出口情報

本章では、Systemwalker Operation Managerが提供する出口の情報について説明します。

注意

出口プログラムの配置には、システム管理者(Administratorsグループに属するユーザ/スーパーユーザ)権限が必要です。

19.1　ジョブネット正常終了/異常終了出口

ジョブスケジューラに登録してあるジョブネットが正常終了、疑似正常、異常終了または強制終了した場合に呼び出される、出口プログラムについて説明します。

【Windows版】

ジョブネットが正常終了または疑似正常終了したときは、normalexit.bat、normalexit.exeの順に出口プログラムが検索され、見つかった出口プログラムが呼び出されます。normalexit.batが見つかった場合、normalexit.exeが存在しても呼び出されません。

ジョブネットが異常終了または強制終了したときは、jobschexit.bat、jobschexit.exeの順に出口プログラムが検索され、見つかった出口プログラムが呼び出されます。jobschexit.batが見つかった場合、jobschexit.exeが存在しても呼び出されません。

利用者がこれらの出口を利用して独自の処理を行いたい場合は、同名の出口プログラムを作成して、以下のディレクトリに格納してください。

```
Systemwalkerインストールディレクトリ¥MpWalker.JM¥bin
```

【UNIX版】

ジョブネットが正常終了または疑似正常したときは、jobsch.exit.normalの出口プログラムが検索され、見つかった出口プログラムが呼び出されます。

ジョブネットが異常終了または強制終了したときは、jobsch.exitの出口プログラムが検索され、見つかった出口プログラムが呼び出されます。

利用者がこれらの出口を利用して独自の処理を行いたい場合は、同名の出口プログラムを作成して、以下のディレクトリに格納してください。

Solaris/Linux版：

```
/opt/FJSVJOBSC/bin
```

HP-UX版：

```
/opt/FHPJOBSCH/bin
```

AIX版：

```
/usr/FAIXJOBSC/bin
```

出口プログラムには、実行権が設定されている必要があります。実行権が設定されていないと呼び出されません。

出口プログラムに渡されるパラメタ

出口プログラムに渡されるパラメタは、以下のとおりです。

```
argc ： パラメタの数(7)                    (注1)
argv[0] ： 出口プログラム名
```

```
argv[1] ： ジョブネットの登録されているプロジェクト名
argv[2] ： ジョブネット名
argv[3] ： ジョブネット名称
argv[4] ： ジョブネット内ジョブの終了コードの最大値(文字列)
argv[5] ： ジョブネット内ジョブ数(文字列)
argv[6] ： 一時ファイル名
argv[7] ： サブシステム番号(文字列)              (注2)
```

注1)

Systemwalker Operation Manager EEの場合、パラメタの数は8になります。

注2)

Systemwalker Operation Manager EEのパラメタです。

一時ファイルには、ジョブネット内の個々のジョブについての情報が、以下のように格納されます。ジョブ名/終了コード/状態がジョブ1個分の情報です。

```
ジョブ名
終了コード
状態
ジョブ名
終了コード
状態
……
ジョブ名
終了コード
状態
```

上図の各情報について説明します。

ジョブ名:

実行属性によって、以下の内容が格納されます。

実行属性が“ジョブ実行制御”の場合:

ジョブ名です。ジョブ名が省略されている場合は、コマンド名です。

実行属性が“Interstage”の場合:

ジョブのワークユニット名です。

実行属性が“PowerAIM”の場合:【UNIX版】

ジョブのワークユニット名です。

実行属性が“旧バージョン互換(旧・標準)”の場合:【UNIX版】

コマンド名です。

終了コード:

ジョブの終了コード(0～256までの文字)です。未実行のジョブの終了コードは0です。

状態:

ジョブの状態を以下の7つのうちいずれかの文字列で表します。

normal_end:

正常終了したことを示します。

pseudo-normal:

疑似正常終了したことを示します。

abnormal_end:

異常終了したことを示します。

canceled:

強制終了したことを示します。

not_executed:

未実行であったことを示します。

paused:

停止状態であることを示します。

disabled:

無効状態であることを示します。

一時ファイルは、以下のディレクトリに作成されます。

Windows版:

ジョブスケジューラのデータベースディレクトリ配下のworkディレクトリ。正常終了出口のファイル名はNMLnnn.tmp(nnnはシステムが付加)、異常終了出口のファイル名はNETnnn.tmp(nnnはシステムが付加)の形式で作成されます。

Solaris版、HP-UX版:

/var/tmpディレクトリ

AIX版、Linux版:

/tmpディレクトリ

一時ファイルは、ジョブスケジューラ側では削除しないので、出口プログラムで削除する必要があります。

出口が登録されていない場合は、一時ファイルは作成されません。

パス状態のジョブについて

先行ジョブの終了コードによる後続ジョブの起動条件を指定している場合に、パス状態で終了したジョブは、以下のようになります。

- ジョブの情報としては、パスされる以前のジョブの状態(実行待ち/無効状態/停止中)が渡ります。
- 終了コードは0になります。

また、OR条件で待ち合わせている場合に、後続ジョブのOR条件により強制終了されたジョブは、以下のようになります。

- 実際に起動されたかどうかに関わらず、ジョブの状態は強制終了(canceled)になります。
- 終了コードは248または249になります。

注意事項

- ジョブネットやグループの実行が遅延する可能性があるため、時間のかかる処理は行わないでください。
- Windows版の場合、出口の実行ユーザは、ジョブスケジューラサービスのログオンアカウントです。【Windows版】
- UNIX版の場合、出口の実行ユーザは、“root”となります。【UNIX版】
- 出口内で、当該ジョブネットの操作(無効操作、削除操作など)を実施した場合、操作に失敗する場合があります。これは、性能を考慮し、出口呼び出し時点で、ジョブネットの状態が終了状態であることを保証していないためです。出口でジョブネットを操作したい場合は、操作に失敗する場合を考慮し、一定時間リトライするなど考慮してください。

プログラム例

ジョブネット異常終了出口のプログラム例を以下に示します。

Windows版の例

```
/*  ジョブネット異常終了時に、"KANRI"サーバに以下のメッセージを送信する
              ジョブネット名 is abended !! code = 終了コード*/
#include <stdio.h>
#include <process.h>

main(int argc, char *argv[])
{
    char cmdline[256];

    sprintf_s(cmdline,sizeof(cmdline),
            "NET SEND KANRI %s is abended !! code = %s",
            argv[2],    //ジョブネット名
            argv[4]);   //終了コード

    system(cmdline);

    _unlink(argv[6]);    // temp ファイルを削除

    exit(0);
}
```

Solaris版の例

```
/*
  コンソールに、ジョブネットが異常終了したというメッセージが
  ジョブネット名および終了コードと共に出力される、ジョブネット
  異常終了出口のプログラム例(Solaris版)
*/

#include <syslog.h>
#include <stdio.h>

main(int argc, char **argv)
{
    openlog("jobsch", LOG_CONS, LOG_USER);
    syslog(LOG_WARNING,
        "WARNING: jobnet abnormal end. project=%s, net_name=%s, net_comment=%s, net_code=%s",
        argv[1], argv[2], argv[3], argv[4]);
    closelog();

    unlink(argv[6]);
    exit(0);
}
```

19.2　ジョブネット異常終了拡張出口

ジョブネットが異常終了または強制終了した場合に呼び出される出口プログラムについて説明します。

【Windows版】

ジョブネットが異常終了または強制終了したとき、jobschexitex.bat、jobschexitex.exeの順に出口プログラムが検索され、見つかれば呼び出されます。jobschexitex.batが見つかった場合、jobschexitex.exeが存在しても呼び出されません。

利用者がこれらの出口を利用して独自の処理を行いたい場合は、同名の出口プログラムを作成して、以下のディレクトリに格納してください。

Systemwalkerインストールディレクトリ¥MpWalker.JM¥bin

【UNIX版】

ジョブネットが異常終了または強制終了したとき、jobschex.exitの出口プログラムが検索され、見つかった出口プログラムが呼び出されます。

利用者がこの出口を利用して、独自の処理を行いたい場合は、同名の出口プログラムを作成して、以下のディレクトリに格納してください。

Solaris/Linux版：

```
/opt/FJSVJOBSC/bin
```

HP-UX版：

```
/opt/FHPJOBSCH/bin
```

AIX版：

```
/usr/FAIXJOBSC/bin
```

出口プログラムには、実行権が設定されている必要があります。実行権が設定されていないと呼び出されません。

ジョブネット異常終了拡張出口は、ジョブネット異常終了出口と異なり、ジョブ実行制御属性のジョブネットが異常終了した場合、出口プログラムに渡される終了コードはジョブ実行制御から返された値となります。

出口プログラムに渡されるパラメタ

```
argc ： パラメタの数(20)
argv[0] ： 出口プログラム名
argv[1] ： ジョブネットの登録されているプロジェクト名
argv[2] ： ジョブネット名
argv[3] ： ジョブネット名称
argv[4] ： ジョブネットが属しているグループ名
        ただし、グループに属していなければ""です。
argv[5] ： ジョブネット内ジョブの終了コードの最大値
        ジョブ実行制御属性のジョブネットの場合、256以上になることもあります。
        ただし、ジョブネットが強制終了した場合は256、実行中断した場合は239です。
argv[6] ： ジョブネットの開始日時
        フォーマットは"年 月/日 時:分:秒"
argv[7] ： ジョブネットの終了日時
        フォーマットは"年 月/日 時:分:秒"
argv[8] ： ジョブネット内ジョブ数(文字列)
argv[9] ： 一時ファイル名
argv[10] ： SE版の場合は""、EE版の場合はサブシステム番号
argv[11～19] ： 予約済み("")
```

一時ファイルには、ジョブネット内の個々のジョブについての情報が、以下のように格納されます。ジョブ名/終了コード/状態がジョブ1個分の情報です。

```
ジョブ名
終了コード
状態
ジョブ名
終了コード
状態
……
ジョブ名
終了コード
状態
```

上図の各情報について説明します。

ジョブ名:

実行属性によって、以下の内容が格納されます。

実行属性が“ジョブ実行制御”の場合:

ジョブ名です。ジョブ名が省略されている場合は、コマンド名です。

実行属性が“Interstage”の場合:

ジョブのワークユニット名です。

実行属性が“PowerAIM”の場合:【UNIX版】

ジョブのワークユニット名です。

実行属性が“旧バージョン互換(旧・標準)”の場合:【UNIX版】

コマンド名です。

終了コード:

ジョブの終了コードです。未実行のジョブの終了コードは0です。

ジョブ実行制御属性のジョブネットの場合、256以上になることもあります。

状態:

ジョブの状態を以下の7つのうちいずれかの文字列で表します。

normal_end:

正常終了したことを示します。

pseudo-normal:

疑似正常終了したことを示します。

abnormal_end:

異常終了したことを示します。

canceled:

強制終了したことを示します。

not_executed:

未実行であったことを示します。

paused:

停止状態であることを示します。

disabled:

無効状態であることを示します。

一時ファイルは、以下のディレクトリに作成されます。

Windows版:

ジョブスケジューラのデータベースディレクトリ配下のworkディレクトリ。ファイル名はNEXnnn.tmp(nnnはシステムが付加)の形式で作成されます。

Solaris版、HP-UX版:

/var/tmpディレクトリ

AIX版、Linux版:

/tmpディレクトリ

一時ファイルは、ジョブスケジューラ側では削除しないので、出口プログラムで削除する必要があります。

出口が登録されていない場合は、一時ファイルは作成されません。

パス状態のジョブについて

先行ジョブの終了コードによる後続ジョブの起動条件を指定している場合に、パス状態で終了したジョブは、以下のようになります。

- ジョブの情報としては、パスされる以前のジョブの状態(実行待ち/無効状態/停止中)が渡ります。
- 終了コードは0になります。

また、OR条件で待ち合わせている場合に、後続ジョブのOR条件により強制終了されたジョブは、以下のようになります。

- 実際に起動されたかどうかに関わらず、ジョブの状態は強制終了(canceled)になります。
- 終了コードは248または249になります。

出口プログラム格納場所

Windows	Systemwalker Operation Managerインストールディレクトリ¥MpWalker.JM¥bin
Solaris	/opt/FJSVJOBSC/bin
HP-UX	/opt/FHPJOBSCH/bin
AIX	/usr/FAIXJOBSC/bin
Linux	/opt/FJSVJOBSC/bin

注意事項

- ジョブネットやグループの実行が遅延する可能性があるため、時間のかかる処理は行わないでください。
- Windows版の場合、出口の実行ユーザは、ジョブスケジューラサービスのログオンアカウントです。【Windows版】
- UNIX版の場合、出口の実行ユーザは、“root”となります。【UNIX版】

プログラム例

ジョブネット異常終了拡張出口のプログラム例を以下に示します。

Windows版の例

```
/*  ジョブネット異常終了時に、"KANRI"サーバに以下のメッセージを送信する
       ジョブネット名(グループ名) is abended !! code = 終了コード*/
#include <stdio.h>
#include <process.h>

main(int argc, char *argv[])
{
    char cmdline[256];

    sprintf_s(cmdline,sizeof(cmdline),
            "NET SEND KANRI %s(%s) is abended !! code = %s",
            argv[2],    //ジョブネット名
            argv[4],    //グループ名
            argv[5]);   //終了コード

    system(cmdline);

    _unlink(argv[9]);    // temp ファイルを削除

    exit(0);
}
```

Solaris版の例

```
/*
  コンソールに、ジョブネットが異常終了したというメッセージが
  グループ名、ジョブネット名および終了コードと共に出力される、
  ジョブネット異常終了拡張出口のプログラム例(Solaris版)
*/

#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <syslog.h>

main(int argc, char **argv)
{
    openlog("jobsch", LOG_CONS, LOG_USER);
    syslog(LOG_WARNING,
        "WARNING: jobnet abnormal end. project=%s, group_name=%s, net_name=%s, net_comment=%s, net_code=%s",
       argv[1], argv[4], argv[2], argv[3], argv[5]);
    closelog();

    unlink(argv[9]);
    exit(0);
}
```

19.3　ジョブ終了出口

ジョブスケジューラに登録してあるジョブが終了した場合に呼び出される、出口プログラムについて説明します。

【Windows版】

jobexit.bat、jobexit.exeの順に出口プログラムが検索され、見つかった出口プログラムが呼び出されます。
jobexit.batが見つかった場合、jobexit.exeが存在しても呼び出されません。

利用者がこれらの出口を利用して、独自の処理を行いたい場合は、同名の出口プログラムを作成して、以下のディレクトリに格納してください。

Systemwalkerインストールディレクトリ¥MpWalker.JM¥bin

【UNIX版】

jobsch.job.exitの出口プログラムが検索され、見つかった出口プログラムが呼び出されます。

利用者がこれらの出口を利用して、独自の処理を行いたい場合は、同名の出口プログラムを作成して、以下のディレクトリに格納してください。

Solaris/Linux版：

/opt/FJSVJOBSC/bin

HP-UX版：

/opt/FHPJOBSCH/bin

AIX版：

/usr/FAIXJOBSC/bin

出口プログラムには、実行権が設定されている必要があります。実行権が設定されていないと呼び出されません。

出口プログラムに渡されるパラメタ

出口プログラムに渡されるパラメタは、以下のとおりです。

```
argc   ： パラメタの数(5)                    (注1)
argv[0] ： 出口プログラム名
argv[1] ： ジョブネットの登録されているプロジェクト名
argv[2] ： ジョブネット名
argv[3] ： ジョブネット名称
argv[4] ： 一時ファイル名
argv[5] ： サブシステム番号(文字列)              (注2)
```

注1)

Systemwalker Operation Manager EEの場合、パラメタの数は6になります。

注2)

Systemwalker Operation Manager EEのパラメタです。

一時ファイルには、ジョブ情報が以下のように格納されます。

```
ジョブ名
終了コード
状態
```

上図の各情報について説明します。

ジョブ名:

実行属性によって、以下の内容が格納されます。

ジョブとして子ジョブネットが登録されている場合、子ジョブネットのジョブネット名が格納されます。

ジョブとしてリンクジョブネットが登録されている場合、リンクジョブネットのジョブネット名が格納されます。

実行属性が“ジョブ実行制御”の場合:

ジョブ名です。ジョブ名が省略されている場合は、コマンド名です。

実行属性が“Interstage”の場合:

ジョブのワークユニット名です。

実行属性が“PowerAIM”の場合:【UNIX版】

ジョブのワークユニット名です。

実行属性が“旧バージョン互換(旧・標準)”の場合:【UNIX版】

コマンド名です。

終了コード:

ジョブの終了コード(0～256までの文字)です。

状態:

ジョブの状態を以下の4つのうちいずれかの文字列で表します。

normal_end:

正常終了したことを示します。

pseudo-normal:

疑似正常終了したことを示します。

abnormal_end:

異常終了したことを示します。

canceled:

強制終了したことを示します。

一時ファイルは、以下のディレクトリに作成されます。

Windows版:

ジョブスケジューラのデータベースディレクトリ配下のworkディレクトリ。ファイル名はJOBnnn.tmp (nnnはシステムが付加)の形式で作成されます。

Solaris版、HP-UX版:

/var/tmpディレクトリ

AIX版、Linux版:

/tmpディレクトリ

一時ファイルは、ジョブスケジューラ側では削除しないので、出口プログラムで削除する必要があります。出口が登録されていない場合は、一時ファイルは作成されません。

パス状態のジョブについて

先行ジョブの終了コードによる後続ジョブの起動条件を指定している場合に、パス状態で終了したジョブは、ジョブ終了出口が起動されません。

また、OR条件で待ち合わせている場合に、後続ジョブのOR条件により強制終了されたジョブは、以下のようになります。

- 起動されてから強制終了されたジョブは、ジョブの状態は強制終了(canceled)、終了コードは248または249になります。
- 起動されずに強制終了されたジョブは、ジョブ終了出口が起動されません。

注意事項

- ジョブネットやグループの実行が遅延する可能性があるため、時間のかかる処理は行わないでください。
- Windows版の場合、出口の実行ユーザは、ジョブスケジューラサービスのログオンアカウントです。【Windows版】
- UNIX版の場合、出口の実行ユーザは、“root”となります。【UNIX版】

プログラム例

ジョブ終了出口のプログラム例を以下に示します。

Windows版の例

```
/*  ジョブ終了時に、"KANRI"サーバに以下のメッセージを送信する
      ジョブ名 is ended !! code = 終了コード
*/
#include <stdio.h>
#include <string.h>
#include <process.h>

main(int argc, char *argv[])
{
    FILE *fp;
    char jobname[100];
    char jobcode[100];
    char cmdline[256];

    errno_t err;
    err = fopen_s(&fp,argv[4], "r");
    fgets(jobname, 99, fp);
    jobname[strlen(jobname)] = '¥0';
    fgets(jobcode, 99, fp);
    jobcode[strlen(jobcode)] = '¥0';
    fclose(fp);
```

```
    sprintf_s(cmdline,sizeof(cmdline),
            "NET SEND KANRI %s is ended !! code = %s",
            jobname,    //ジョブ名
            jobcode);   //終了コード

    system(cmdline);

    _unlink(argv[4]);    // temp ファイルを削除

    exit(0);
}
```

Solaris版の例

```
/*
  コンソールに、ジョブが終了したというメッセージが、ジョブ名および
  終了コードと共に出力される、ジョブ終了出口のプログラム例(Solaris版)
*/

#include <syslog.h>
#include <stdio.h>

main(int argc, char **argv)
{
    FILE      *fp;
    char      jobname[100];
    char      jobcode[100];

    memset(jobname, 0x00, sizeof(jobname));
    memset(jobcode, 0x00, sizeof(jobcode));

    fp = fopen(argv[4], "r");
    fgets(jobname, sizeof(jobname), fp);
    jobname[strlen(jobname)-1] = '¥0';
    fgets(jobcode, sizeof(jobcode), fp);
    jobcode[strlen(jobcode)-1] = '¥0';
    fclose(fp);

    openlog("jobsch", LOG_CONS, LOG_USER);
    syslog(LOG_WARNING,
        "WARNING: job end. project=%s, net_name=%s, net_comment=%s, job_name=%s, job_code=%s",
        argv[1], argv[2], argv[3], jobname, jobcode);
    closelog();

    unlink(argv[4]);
    exit(0);
}
```

19.4　ログファイル切換え出口

ジョブスケジューラのログファイル(jobdb1.log,jobdb2.log,jobdb3.log)が切り換わった場合に呼び出される、出口プログラムについて説明します。

【Windows版】

ログファイルが切り換わると、jobdbexit.bat、jobdbexit.exeの順に出口プログラムが検索され、見つかった出口プログラムが呼び出されます。jobdbexit.batが見つかった場合、jobdbexit.exeが存在しても呼び出されません。

利用者がこの出口を利用して、独自の処理を行いたい場合は、同名の出口プログラムを作成して、以下のディレクトリに格納してください。

Systemwalkerインストールディレクトリ¥MpWalker.JM¥bin

【UNIX版】

ログファイルが切り換わると、jobdb.exitの出口プログラムが検索され、見つかった出口プログラムが呼び出されます。

利用者がこの出口を利用して、独自の処理を行いたい場合は、同名の出口プログラムを作成して、以下のディレクトリに格納してください。

Solaris/Linux版：

/opt/FJSVJOBSC/bin

HP-UX版：

/opt/FHPJOBSCH/bin

AIX版：

/usr/FAIXJOBSC/bin

出口プログラムには、実行権が設定されている必要があります。実行権が設定されていないと呼び出されません。

出口プログラムに渡されるパラメタ

出口プログラムに渡されるパラメタは、以下のとおりです。

argc　：パラメタの数(2)　(注1) argv[0]：出口プログラム名 argv[1]：切り換えられる前のログファイル名 argv[2]：サブシステム番号(文字列)　(注2)

注1)

Systemwalker Operation Manager EEの場合、パラメタの数は3になります。

注2)

Systemwalker Operation Manager EEのパラメタです。

注意事項

- ジョブネットやグループの実行が遅延する可能性があるため、時間のかかる処理は行わないでください。
- Windows版の場合、出口の実行ユーザは、ジョブスケジューラサービスのログオンアカウントです。【Windows版】
- UNIX版の場合、出口の実行ユーザは、“root”となります。【UNIX版】

プログラム例

ログファイル切換え出口のプログラム例を以下に示します。

Windows版の例

```
/*
   DB-CHANGEというメッセージ事象を発生させる
*/

#include <process.h>

main()
```

```
{
    system(”jobschmsgevent DB-CHANGE”);

    exit(0);
}
```

Solaris版の例

```
/*
  コンソールにログが切り換わったことを知らせるメッセージが出力される、
  ログファイル切換え出口のプログラム例(Solaris版)
*/

#include <syslog.h>
#include <stdio.h>

main(int argc, char **argv)
{
    openlog(”jobsch”, LOG_CONS, LOG_USER);
    syslog(LOG_WARNING, ”WARNING: Log file was switched. file=%s”, argv[1]);
    closelog();

    exit(0);
}
```

19.5　日変わり時刻到来出口

日変わり時刻が到来した時に呼び出される、出口プログラムについて説明します。

【Windows版】

日変わり時刻が到来すると、timeexit.bat、timeexit.exeの順に出口プログラムが検索され、見つかった出口プログラムが呼び出されます。timeexit.batが見つかった場合、timeexit.exeが存在しても呼び出されません。

利用者がこの出口を利用して、独自の処理を行いたい場合は、同名の出口プログラムを作成して、以下のディレクトリに格納してください。

```
Systemwalkerインストールディレクトリ¥MpWalker.JM¥bin
```

【UNIX版】

日変わり時刻が到来すると、jobsch.time.exitの出口プログラムが検索され、見つかった出口プログラムが呼び出されます。

利用者がこの出口を利用して、独自の処理を行いたい場合は、同名の出口プログラムを作成して、以下のディレクトリに格納してください。

Solaris/Linux版：

```
/opt/FJSVJOBSC/bin
```

HP-UX版：

```
/opt/FHPJOBSCH/bin
```

AIX版：

```
/usr/FAIXJOBSC/bin
```

出口プログラムには、実行権が設定されている必要があります。実行権が設定されていないと呼び出されません。

出口プログラムに渡されるパラメタ

出口プログラムに渡されるパラメタは、以下のとおりです。

argc ： パラメタの数(1)	(注1)
argv[0] ： 出口プログラム名	
argv[1] ： サブシステム番号(文字列)	(注2)

注1)

Systemwalker Operation Manager EEの場合、パラメタの数は2になります。

注2)

Systemwalker Operation Manager EEのパラメタです。

注意事項

- ジョブネットやグループの実行が遅延する可能性があるため、時間のかかる処理は行わないでください。
- Windows版の場合、出口の実行ユーザは、ジョブスケジューラサービスのログオンアカウントです。【Windows版】
- UNIX版の場合、出口の実行ユーザは、“root”となります。【UNIX版】
- 複数サブシステム運用の場合、サブシステムごとに呼び出されます。

プログラム例

日変わり時刻到来出口のプログラム例を以下に示します。

Windows版の例

```
/*
   JOBSCH-TIMEというメッセージ事象を発生させる
*/

#include <stdio.h>
#include <string.h>
#include <process.h>

main(int argc, char *argv[])
{
    char msg[256];
    memset(msg,0x00,sizeof(msg));
    sprintf_s(msg,sizeof(msg),”jobschmsgevent JOBSCH-TIME-sys%s”,argv[1]);
    system(msg);

    exit(0);
}
```

UNIX版の例

```
/*
  日変わり時刻が到来したことを知らせるメッセージがコンソール
  に出力される、日変わり時刻到来出口のプログラム例(UNIX版)
*/

#include <syslog.h>
#include <stdio.h>
#include <string.h>

main(int argc, char *argv[])
{
```

```
    char msg[256];
    memset(msg,0x00,sizeof(msg));
    sprintf(msg, "WARNING: Day change time has come. sys%s",argv[1]);
    openlog("jobsch", LOG_CONS, LOG_USER);
    syslog(LOG_WARNING, msg);
    closelog();

    exit(0);
}
```

19.6　正常終了/異常終了シャットダウン出口【Windows版】

終了処理ジョブネットの実行が終わると、ジョブスケジューラは、終了処理ジョブネットの実行結果(正常終了または疑似正常/異常終了)に応じて、以下に示すシャットダウン出口プログラムを呼び出します。

終了処理ジョブネットの実行結果に応じて、何らかの処理を行いたい場合には、同名の出口プログラムを作成して、以下のディレクトリに格納してください。

```
Systemwalkerインストールディレクトリ¥MpWalker.JM¥bin
```

なお、インストール時には、バッチファイル(jobschendexit.batおよびjobschnoendexit.bat)が提供されています。

正常終了または疑似正常の時

jobschendexit.bat、jobschendexit.exeの順に出口プログラムが検索され、見つかった出口プログラムが呼び出されます。jobschendexit.batが見つかった場合、jobschendexit.exeが存在しても呼び出されません。

インストール時に提供されるjobschendexit.batでは、ジョブスケジューラおよびジョブ実行制御のサービスを停止し、待ち合わせ終了通知"system"で待ち合わせ終了通知コマンド(f3crheet.exe)を呼び出します。

マシンをシャットダウンするためには、待ち合わせ終了通知"system"を[待ち合わせ終了通知定義]ダイアログボックスで定義しておいてください。

[待ち合わせ終了通知定義]ダイアログボックスについては、“Systemwalker Operation Manager オンラインヘルプ”を参照してください。

異常終了した時

jobschnoendexit.bat、jobschnoendexit.exeの順に出口プログラムが検索され、見つかった出口プログラムが呼び出されます。jobschnoendexit.batが見つかった場合、jobschnoendexit.exeが存在しても呼び出されません。

インストール時に提供されるjobschnoendexit.batでは、何の処理も行いません。

したがって、終了処理ジョブネットが異常終了した時には、シャットダウンは行われません。

注意事項

- 出口の実行ユーザは、ジョブスケジューラサービスのログオンアカウントです。
- EE 複数サブシステム運用を行っている場合は、引数としてサブシステム番号が渡されます。運用形態に合わせて必ずシャットダウン出口の変更を行ってください。

EE サブシステム運用時の使用例

サブシステム運用を行っている場合は、すべてのサブシステム終了後にシステムをシャットダウンする必要があります。その場合の出口の例を以下に示します。

出口プログラムに渡されるパラメタは、%6(サブシステム番号)です。

```
echo off

echo shutdown job net exit routine start
```

```
if "%6" == "0" goto zero
if not "%6" == "0" goto notzero

:zero
rem --- Systemwalker MpJobsch ---
net stop "Fujitsu MpWalker MpJobsch"
rem --- Systemwalker MpMjes ---
mjstop -sys %6
net stop "Fujitsu MpWalker MpMjes"
goto done

:notzero
rem --- Systemwalker MpJobsch ---
net stop "Fujitsu MpWalker MpJobsch%6"
rem --- Systemwalker MpMjes ---
mjstop -sys %6
net stop "Fujitsu MpWalker MpMjes%6"

:done
f3crheet.exe system%6
```

なお、使用例のような設定を行う場合は、f3crheetコマンドで指定する待ち合わせ終了通知を、あらかじめ[待ち合わせ終了通知定義]ダイアログボックスで定義しておいてください。それにより、f3crheetコマンドで指定した待ち合わせ終了通知が出力されるまでシャットダウン処理が延期されます。

19.7　ジョブネット起動遅延出口

ジョブスケジューラに登録してあるジョブネットが起動予定時刻になっても起動されない場合に呼び出される、出口プログラムについて説明します。

【Windows版】

ジョブネットが起動予定時刻になっても起動されないときは、startlateexit.bat、startlateexit.exeの順に出口プログラムが検索され、見つかった出口プログラムが呼び出されます。startlateexit.batが見つかった場合、startlateexit.exeが存在しても呼び出されません。

また、[メッセージ事象発生時のみ起動]で[一日一回]と[開始予定時刻]が設定されているジョブネットで、開始予定時刻が到来してもメッセージ事象が発生していない、または開始予定時刻の時点で無効状態であった場合も呼び出されます。

利用者がこれらの出口を利用して、独自の処理を行いたい場合は、同名の出口プログラムを作成して、以下のディレクトリに格納してください。

```
Systemwalkerインストールディレクトリ¥MpWalker.JM¥bin
```

【UNIX版】

startlate.exitの出口プログラムが検索され、見つかった出口プログラムが呼び出されます。

また、[メッセージ事象発生時のみ起動]で[一日一回]と[開始予定時刻]が設定されているジョブネットで、開始予定時刻が到来してもメッセージ事象が発生していない、または開始予定時刻の時点で無効状態であった場合も呼び出されます。

利用者がこれらの出口を利用して、独自の処理を行いたい場合は、同名の出口プログラムを作成して、以下のディレクトリに格納してください。

Solaris/Linux版：

```
/opt/FJSVJOBSC/bin
```

HP-UX版：

```
/opt/FHPJOBSCH/bin
```

AIX版：

```
/usr/FAIXJOBSC/bin
```

出口プログラムには、実行権が設定されている必要があります。実行権が設定されていないと呼び出されません。

出口プログラムに渡されるパラメタ

出口プログラムに渡されるパラメタは、以下のとおりです。

```
argc   ： パラメタの数(4)                    (注1)
argv[0] ： 出口プログラム名
argv[1] ： ジョブネットの登録されているプロジェクト名
argv[2] ： ジョブネット名
argv[3] ： ジョブネット名称
argv[4] ： サブシステム番号(文字列)              (注2)
```

注1)

Systemwalker Operation Manager EEの場合、パラメタの数は5になります。

注2)

Systemwalker Operation Manager EEのパラメタです。

注意事項

- ジョブネットやグループの実行が遅延する可能性があるため、時間のかかる処理は行わないでください。
- Windows版の場合、出口の実行ユーザは、ジョブスケジューラサービスのログオンアカウントです。【Windows版】
- UNIX版の場合、出口の実行ユーザは、“root”となります。【UNIX版】

プログラム例

ジョブネット起動遅延出口のプログラム例を以下に示します。

Windows版の例

```
/*ジョブネット起動遅延時に、”KANRI” サーバに以下のメッセージを送信する
            ジョブネット名 is delayed
*/
#include <stdio.h>
#include <process.h>

main(int argc, char *argv[])
{
        char cmdline[256];

        sprintf_s(cmdline,sizeof(cmdline),
                        ”NET SEND KANRI %s is delayed”,
                        argv[2]);       //ジョブネット名

        system(cmdline);

        exit (0);
}
```

Solaris版の例

```
/*
  コンソールに、遅れているというメッセージが、プロジェクト名およびジョブネット名
  と共に出力される、ジョブネット起動遅延出口のプログラム例(Solaris版)
*/

#include <syslog.h>
#include <stdio.h>

main(int argc, char **argv)
{
    openlog("jobsch", LOG_CONS, LOG_USER);
    syslog(LOG_WARNING,
        "WARNING: %s %s is delayed.",
        argv[1], argv[2]);
    closelog();

    exit(0);
}
```

19.8　ジョブネット終了遅延出口

ジョブスケジューラに登録してあるジョブネットが終了予定時刻になっても終了しない場合に呼び出される、出口プログラムについて説明します。

【Windows版】

ジョブネットが終了予定時刻になっても終了しないときは、endlateexit.bat、endlateexit.exeの順に出口プログラムが検索され、見つかった出口プログラムが呼び出されます。endlateexit.batが見つかった場合、endlateexit.exeが存在しても呼び出されません。

利用者がこれらの出口を利用して、独自の処理を行いたい場合は、同名の出口プログラムを作成して、以下のディレクトリに格納してください。

```
Systemwalkerインストールディレクトリ¥MpWalker.JM¥bin
```

【UNIX版】

endlate.exitの出口プログラムが検索され、見つかった出口プログラムが呼び出されます。

利用者がこれらの出口を利用して、独自の処理を行いたい場合は、同名の出口プログラムを作成して、以下のディレクトリに格納してください。

Solaris/Linux版：

```
/opt/FJSVJOBSC/bin
```

HP-UX版：

```
/opt/FHPJOBSCH/bin
```

AIX版：

```
/usr/FAIXJOBSC/bin
```

出口プログラムには、実行権が設定されている必要があります。実行権が設定されていないと呼び出されません。

出口プログラムに渡されるパラメタ

出口プログラムに渡されるパラメタは、以下のとおりです。

argc　：パラメタの数(5)　　(注1) argv[0]：出口プログラム名 argv[1]：ジョブネットの登録されているプロジェクト名 argv[2]：ジョブネット名 argv[3]：ジョブネット名称 argv[4]：ジョブネットの状態 argv[5]：サブシステム番号(文字列)　　(注2)

注1)

Systemwalker Operation Manager EEの場合、パラメタの数は6になります。

注2)

Systemwalker Operation Manager EEのパラメタです。

第5パラメタ“ジョブネットの状態”に格納される文字列の意味を以下に示します。

not_executed:

ジョブネットが“実行待ち”であることを示します。

executing:

ジョブネットが“実行中”であることを示します。

warning:

ジョブネットが“警告状態”であることを示します。

注意事項

- ジョブネットやグループの実行が遅延する可能性があるため、時間のかかる処理は行わないでください。
- Windows版の場合、出口の実行ユーザは、ジョブスケジューラサービスのログオンアカウントです。【Windows版】
- UNIX版の場合、出口の実行ユーザは、“root”となります。【UNIX版】

プログラム例

ジョブネット終了遅延出口のプログラム例を以下に示します。

Windows版の例

```
/*ジョブネット終了遅延時に、"KANRI"サーバに以下のメッセージを送信する
            ジョブネット名 is delayed
*/
#include <stdio.h>
#include <process.h>

main(int argc, char *argv[])
{
        char cmdline[256];

        sprintf_s(cmdline,sizeof(cmdline),
                        "NET SEND KANRI %s is delayed",
                        argv[2]);       //ジョブネット名

        system(cmdline);

        exit (0);
}
```

Solaris版の例

```
/*
  コンソールに、遅れているというメッセージが、プロジェクト名およびジョブネット名
  と共に出力される、ジョブネット終了遅延出口のプログラム例(Solaris版)
*/

#include <syslog.h>
#include <stdio.h>

main(int argc, char **argv)
{
    openlog("jobsch", LOG_CONS, LOG_USER);
    syslog(LOG_WARNING,
        "WARNING: %s %s is delayed.",
        argv[1], argv[2]);
    closelog();

    exit(0);
}
```

19.9 サポート外のショートメールのメッセージ変換用出口【Windows版】

ショートメールへメッセージを送信するとき、メッセージテキストをコードに変換する必要があります。

Systemwalker Operation Managerが標準でサポートしていない会社のショートメールを使用する場合、メッセージを変換する出口(DLL)をあらかじめ用意する必要があります。

作成したDLLは、DLL名を“f3crhxpc.dll”とし、以下のディレクトリに格納してください。

Systemwalkerインストールディレクトリ¥MpWalker.JM¥bin

出口プログラムとのインタフェース

メッセージを変換する関数は以下の形式で呼び出されます。

```
int f3crhx_convert(char *Msg_Text,char *MsgNo,int MsgNoSize,char *Vender)

Msg_Text  ： ショートメールへ送信する文字(変換するメッセージテキストの文字列)
         が設定されているアドレスが設定されています。
MsgNo     ： 変換結果を格納する領域が用意されています。出口で指定してください。
          Msg_Textに格納されている文字をダイヤルする番号の文字列に変換して指定
          してください。MsgNoSize に設定された長さの範囲で指定し、文字列の終わ
          りには'¥0'を指定してください。
MsgNoSize ： 変換結果を格納する領域(MsgNo)のサイズが設定されています。
Vender    ： ショートメール会社名が設定されています。
```

復帰コード

メッセージの変換結果を関数の復帰コードにより通知してください。

正常終了:

0以上の整数を通知します。ショートメールの送信を行います。

異常終了:

0未満の整数を通知します。ショートメールの送信は行われません。

注意事項

- 本DLLを作成すると共に[アクション環境設定]ダイアログボックスの[ショートメール]シートで、利用するショートメールの会社を登録する必要があります。
- 本DLLが異常終了した場合、Systemwalkerのサービスが停止します。
- 本DLLから制御が戻らない場合、その後のショートメール送信も行われません。

プログラム例

サポート外のショートメールのメッセージ変換用出口のプログラム例を以下に示します。

```
int f3crhx__convert(char *MsgText,char *MsgNo,int MsgNoSize,char *Vender)
{
    int i ;
    int NoLen = 0 ;

    if ( strcmp(Vender,"ven01") != 0 )  {      //  サポート外の会社
        return -1 ;
    }

    //  メッセージ変換処理
    for ( i = 0 ; i < strlen(MsgText) ; i++ ) {
        /* MsgNoに変換結果を格納する */
        /* NoLenには、MsgNo に格納した文字列の数を格納する */
    }
    MsgNo[NoLen] = '¥0' ;                      //  終端コードの設定

    return 0 ;                                 //  正常終了
}
```

19.10　デマンドジョブ終了出口

デマンドジョブが終了した場合に呼び出される、出口プログラムについて説明します。

【Windows版】

mjesexit.bat、mjesexit.exeの順に出口プログラムが検索され、見つかった出口プログラムが呼び出されます。mjesexit.batが見つかった場合、mjesexit.exeが存在しても呼び出されません。

利用者がこれらの出口を利用して、独自の処理を行いたい場合は、同名の出口プログラムを作成して、以下のディレクトリに格納してください。

```
Systemwalkerインストールディレクトリ¥ MpWalker.JM¥bin
```

【UNIX版】

mjes.job.exitの出口プログラムが検索され、見つかった出口プログラムが呼び出されます。

利用者がこれらの出口を利用して、独自の処理を行いたい場合は、同名の出口プログラムを作成して、以下のディレクトリに格納してください。

Solaris/Linux：

```
/opt/FJSVMJS/usr/lib/mjes
```

HP-UX：

```
/opt/FHPMJS/usr/lib/mjes
```

AIX：

```
/opt/FAIXMJS/usr/lib/mjes
```

出口プログラムには、実行権が設定されている必要があります。実行権が設定されていないと呼び出されません。

出口プログラムに渡されるパラメタ

出口プログラムに渡されるパラメタは、以下のとおりです。

```
argc ： パラメタの数(9)
argv[0] ： 出口プログラム名
argv[1] ： ジョブ名
argv[2] ： ジョブ番号(文字列)
argv[3] ： ジョブが実行された実行サーバのホスト名
argv[4] ： ジョブが実行されたキュー名
argv[5] ： ジョブに指定されたコメント
argv[6] ： ジョブの終了コード(文字列)
argv[7] ： ジョブの標準出力ファイルのファイル名
argv[8] ： ジョブの標準エラー出力ファイルのファイル名
argv[9] ： サブシステム番号(文字列)              (注1)
```

注1)

Systemwalker Operation Manager EEのパラメタです。SE版では必ず 0 が設定されます。

注意事項

- UNIX版はジョブ所有者の権限でデマンドジョブ終了出口が呼び出されます。qsubコマンドにおいて、クライアントユーザ名を指定する-cuオプションの指定がなかった場合、qsubコマンドを起動したユーザがジョブ所有者です。Windows版はサービスの起動アカウントの権限でデマンドジョブ終了出口が呼び出されます。
- 出口プログラムは、新規プロセスとして起動され、その終了は待ち合わせません。本出口処理が終了しなくても、当該ジョブは終了したと見なされ、後続のキュー待ちジョブが順次起動されます。
- 本出口処理の実行結果異常に対して、Systemwalker Operation Managerにおいては、何も行いません。
- 何らかの理由によりデマンドジョブ終了出口が存在するのに呼び出せなかった場合、SYSLOG/イベントログにその旨のメッセージが出力されます。

プログラム例

デマンドジョブ終了出口のプログラム例を以下に示します。

Windows版の例

```
/*
   ジョブの終了コードが"0"以外の場合に、"KANRI"サーバに以下の
   メッセージを送信します。
     ジョブ名 is ended !! code = 終了コード
 */
#include <stdio.h>
#include <string.h>
#include <process.h>

main(int argc, char *argv[])
{
    char cmdline[256];
    int job_code = 0;

    job_code = atoi(argv[6]);

    if (job_code != 0) {
```

```
        sprintf_s(cmdline,sizeof(cmdline),
                "NET SEND KANRI %s is ended !! code = %s",
                argv[1],  //ジョブ名
                argv[6]); //終了コード

        system(cmdline);
    }

    exit(0);
}
```

Solaris版の例

```
/*
   ジョブの終了コードが“0”以外の場合に、コンソールにメッセージを
   出力します。出力する情報は、ジョブ名、ジョブ番号、およびジョブの
   終了コードの3つです。
 */

#include <syslog.h>
#include <stdio.h>

main(int argc, char **argv)
{
    int job_code = 0;

    job_code = atoi(argv[6]);

    if (job_code != 0) {
        openlog("mjes.job.exit", LOG_CONS, LOG_USER);
        syslog(LOG_WARNING, "WARNING: job end. job_name=%s, job_number=%s,
                            job_code=%s", argv[1], argv[2], argv[6]);
        closelog();
    }
    exit(0);
}
```

EE

19.11　出口情報(災対用)【Windows版】【Linux版】

本節では、ジョブ/ジョブネットの実行状況を通知するための以下の災害対策用(略して災対用と呼びます)の出口について説明します。

なお、以下の出口は、ジョブスケジューラの起動パラメタファイル(jobschev.txt)でEXIT_CALLをONに設定した場合に使用できます。起動パラメタファイルの設定については、“Systemwalker Operation Manager 運用ガイド”の“災対運用の準備作業(現用環境・災対環境)”を参照してください。

- **ジョブ開始出口(災対用)**
- **ジョブ終了出口(災対用)**
- **ジョブネット開始出口(災対用)**
- **ジョブネット終了出口(災対用)**

  上記出口で、状態情報ファイルを作成し、必要に応じて災害対策環境(以降、災対環境と呼びます)に転送する処理を行います。

  状態情報ファイルのファイル名は、プロジェクト名、ジョブネット名、ジョブ名および日付情報を付加するなどして、各出口で重複しないようにしてください。

  上記出口は、同期で呼び出されます。ただし、以下の場合には非同期で呼び出されます。

- 旧バージョン互換(旧・標準)属性のジョブネットにおいて、ジョブネット自身を管理するプロセスのfork/execに失敗した場合(ジョブネット終了出口)
- ジョブ/ジョブネットが実行中の状態でジョブスケジューラが停止したため、サーバの再起動時に状態を異常終了(実行中断)に変更する場合(ジョブ終了出口/ジョブネット終了出口)
- ジョブネットの強制終了時(ジョブネット終了出口)

また、メッセージ事象の発生状況を通知するための以下の出口について説明します。

- **メッセージ事象発生出口(災対用)**

  上記出口で、発生したメッセージ事象の情報からメッセージ事象情報ファイルを作成し、必要に応じて災対環境に転送する処理を行います。

  メッセージ事象情報ファイルのファイル名は、メッセージ事象名、サブシステム番号および日付情報を付加するなどして、メッセージ事象ごとに重複しないようにしてください。

  本出口は、非同期で呼び出されます。

19.11.1 ジョブ開始出口(災対用)

ジョブスケジューラに登録してあるジョブを開始する場合に呼び出される、出口プログラムについて説明します。

【Windows版】

job_start_exit.bat、job_start_exit.exeの順に出口プログラムが検索され、見つかった出口プログラムが呼び出されます。

job_start_exit.batが見つかった場合、job_start_exit.exeが存在しても呼び出されません。

利用者がこれらの出口を利用して、独自の処理を行いたい場合は、同名の出口プログラムを作成して、以下のディレクトリに格納してください。

```
Systemwalkerインストールディレクトリ¥MpWalker.JM¥bin
```

【Linux版】

job_start.exitの出口プログラムが検索され、見つかった出口プログラムが呼び出されます。

利用者がこれらの出口を利用して、独自の処理を行いたい場合は、同名の出口プログラムを作成して、以下のディレクトリに格納してください。

```
/opt/FJSVJOBSC/bin
```

出口プログラムには、実行権が設定されている必要があります。実行権が設定されていないと呼び出されません。

出口プログラムに渡されるパラメタ

出口プログラムに渡されるパラメタは、以下のとおりです。

```
argc ： パラメタの数(9)
argv[0] ： 出口プログラム名
argv[1] ：自ホスト名(ネットワークジョブの場合はジョブの投入先サーバ名)
argv[2] ：プロジェクト名
argv[3] ：ジョブネット名
argv[4] ：ジョブネット開始日時(形式：YYYY/MM/DD hh:mm:ss)
argv[5] ：ジョブ名
argv[6] ：ジョブ番号(文字列)
argv[7] ：ジョブ開始日時(形式：YYYY/MM/DD hh:mm:ss)
argv[8] ：サブシステム番号(文字列)
```

注意事項

- ジョブ実行制御属性のジョブネットの場合、ジョブの起動時に状態が一度、実行遅延になった後、実行中に変化します。ジョブ開始出口は、ジョブが実行遅延になるタイミングで呼び出されます。
- ジョブネットの実行が遅延する可能性があるため、時間のかかる処理は行わないでください。
- Windows版の場合、出口の実行ユーザは、ジョブスケジューラサービスのログオンアカウントです。【Windows版】
- Linux版の場合、出口の実行ユーザは、“root”となります。【Linux版】

プログラム例

ジョブ開始出口のプログラム例を以下に示します。

転送処理については、お客様の運用にあわせて組み込んでください。

Windows版の例

状態情報ファイル(C:¥state¥プロジェクト名_ジョブネット名_ジョブ番号_サブシステム番号.jobstart)を作成し、災対環境に転送するようにします。

```
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char *argv[]) {
    if (argc != 9) {
        return EXIT_FAILURE;
    }
    const char *program_name = argv[0];
    const char *host_name = argv[1];
    const char *project_name = argv[2];
    const char *jobnet_name = argv[3];
    const char *jobnet_start_datetime = argv[4];
    const char *job_name = argv[5];
    const char *job_number = argv[6];
    const char *job_start_datetime = argv[7];
    const char *subsystem_number = argv[8];
    char file_path[256];
    snprintf(file_path, sizeof(file_path), "C:¥¥state¥¥%s_%s_%s_%s.jobstart", project_name, jobnet_name,
job_number,  subsystem_number);
    FILE *file = fopen(file_path, "w");
    if (file == NULL) {
        perror("Failed to open file");
        return EXIT_FAILURE;
    }
    fprintf(file, "JOB,%s,%s,execute,%s¥n", jobnet_name, job_number, job_start_datetime);
    fclose(file);

    // 次にお客様の運用にあわせて転送処理を組み込みます
    ・・・転送処理・・・

    return EXIT_SUCCESS;
}
```

Linux版の例

状態情報ファイル(/var/tmp/state/プロジェクト名_ジョブネット名_ジョブ番号_サブシステム番号.jobstart)を作成し、災対環境に転送するようにします。

```
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char *argv[]) {
    if (argc != 9) {
```

```
        return EXIT_FAILURE;
    }
    const char *program_name = argv[0];
    const char *host_name = argv[1];
    const char *project_name = argv[2];
    const char *jobnet_name = argv[3];
    const char *jobnet_start_datetime = argv[4];
    const char *job_name = argv[5];
    const char *job_number = argv[6];
    const char *job_start_datetime = argv[7];
    const char *subsystem_number = argv[8];
    char file_path[256];
   snprintf(file_path, sizeof(file_path), "/var/tmp/state/%s_%s_%s_%s.jobstart", project_name, jobnet_name,
job_number, subsystem_number);
    FILE *file = fopen(file_path, "w");
    if (file == NULL) {
        perror("Failed to open file");
        return EXIT_FAILURE;
    }
    fprintf(file, "JOB,%s,%s,execute,%s¥n", jobnet_name, job_number, job_start_datetime);
    fclose(file);

    // 次にお客様の運用にあわせて転送処理を組み込みます
    ・・・転送処理・・・

    return EXIT_SUCCESS;
}
```

19.11.2 ジョブ終了出口(災対用)

ジョブスケジューラに登録してあるジョブが終了した場合に呼び出される、出口プログラムについて説明します。

【Windows版】

job_end_exit.bat、job_end_exit.exeの順に出口プログラムが検索され、見つかった出口プログラムが呼び出されます。

job_end_exit.batが見つかった場合、job_end_exit.exeが存在しても呼び出されません。

利用者がこれらの出口を利用して、独自の処理を行いたい場合は、同名の出口プログラムを作成して、以下のディレクトリに格納してください。

```
Systemwalkerインストールディレクトリ¥MpWalker.JM¥bin
```

【Linux版】

job_end.exitの出口プログラムが検索され、見つかった出口プログラムが呼び出されます。

利用者がこれらの出口を利用して、独自の処理を行いたい場合は、同名の出口プログラムを作成して、以下のディレクトリに格納してください。

```
/opt/FJSVJOBSC/bin
```

出口プログラムには、実行権が設定されている必要があります。実行権が設定されていないと呼び出されません。

出口プログラムに渡されるパラメタ

出口プログラムに渡されるパラメタは、以下のとおりです。

```
argc ：パラメタの数(11)
argv[0] ：出口プログラム名
argv[1] ：自ホスト名(ネットワークジョブの場合はジョブの投入先サーバ名)
argv[2] ：プロジェクト名
```

argv[3] ：ジョブネット名
argv[4] ：ジョブネット開始日時(形式：YYYY/MM/DD hh:mm:ss)
argv[5] ：ジョブ名
argv[6] ：ジョブ番号(文字列)
argv[7] ：ジョブ終了日時(形式：YYYY/MM/DD hh:mm:ss)
argv[8] ：ジョブの終了コード(文字列)
argv[9] ：ジョブの状態(normal/notice/abend/cancel)　※ notice:疑似正常
argv[10] ：サブシステム番号(文字列)

パス状態のジョブについて

先行ジョブの終了コードによる後続ジョブの起動条件を指定している場合に、パス状態で終了したジョブは、ジョブ終了出口が起動されません。

また、OR条件で待ち合わせている場合に、後続ジョブのOR条件により強制終了されたジョブは、以下のようになります。

- 起動されてから強制終了されたジョブは、ジョブの状態は強制終了(cancel)、終了コードは248または249になります。
- 起動されずに強制終了されたジョブは、ジョブ終了出口が起動されません。

注意事項

- ジョブネットの実行が遅延する可能性があるため、時間のかかる処理は行わないでください。
- Windows版の場合、出口の実行ユーザは、ジョブスケジューラサービスのログオンアカウントです。【Windows版】
- Linux版の場合、出口の実行ユーザは、“root”となります。【Linux版】

プログラム例

ジョブ終了出口のプログラム例を以下に示します。

転送処理については、お客様の運用にあわせて組み込んでください。

Windows版の例

状態情報ファイル(C:¥state¥プロジェクト名_ジョブネット名_ジョブ番号_サブシステム番号.jobend)を作成し、災対環境に転送するようにします。

```
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char *argv[]) {
    if (argc != 11) {
        return EXIT_FAILURE;
    }
    const char *program_name = argv[0];
    const char *host_name = argv[1];
    const char *project_name = argv[2];
    const char *jobnet_name = argv[3];
    const char *jobnet_start_datetime = argv[4];
    const char *job_name = argv[5];
    const char *job_number = argv[6];
    const char *job_end_datetime = argv[7];
    const char *job_end_code = argv[8];
    const char *job_status = argv[9];
    const char *subsystem_number = argv[10];
    char file_path[256];
    snprintf(file_path, sizeof(file_path), "C:¥¥state¥¥%s_%s_%s_%s.jobend", project_name, jobnet_name,
job_number, subsystem_number);
    FILE *file = fopen(file_path, "w");
    if (file == NULL) {
```

```
        perror("Failed to open file");
        return EXIT_FAILURE;
    }
    fprintf(file, "JOB,%s,%s,%s,%s,%s¥n", jobnet_name, job_number, job_status, job_end_datetime,
job_end_code);
    fclose(file);

    // 次にお客様の運用にあわせて転送処理を組み込みます
    ・・・転送処理・・・

    return EXIT_SUCCESS;
}
```

Linux版の例

状態情報ファイル(/var/tmp/state/プロジェクト名_ジョブネット名_ジョブ番号_サブシステム番号.jobend)を作成し、災対環境に転送するようにします。

```
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char *argv[]) {
    if (argc != 11) {
        return EXIT_FAILURE;
    }
    const char *program_name = argv[0];
    const char *host_name = argv[1];
    const char *project_name = argv[2];
    const char *jobnet_name = argv[3];
    const char *jobnet_start_datetime = argv[4];
    const char *job_name = argv[5];
    const char *job_number = argv[6];
    const char *job_end_datetime = argv[7];
    const char *job_end_code = argv[8];
    const char *job_status = argv[9];
    const char *subsystem_number = argv[10];
    char file_path[256];
    snprintf(file_path, sizeof(file_path), "/var/tmp/state/%s_%s_%s_%s.jobend", project_name, jobnet_name,
job_number, subsystem_number);
    FILE *file = fopen(file_path, "w");
    if (file == NULL) {
        perror("Failed to open file");
        return EXIT_FAILURE;
    }
    fprintf(file, "JOB,%s,%s,%s,%s,%s¥n", jobnet_name, job_number, job_status, job_end_datetime,
job_end_code);
    fclose(file);

    // 次にお客様の運用にあわせて転送処理を組み込みます
    ・・・転送処理・・・

    return EXIT_SUCCESS;
}
```

19.11.3　ジョブネット開始出口(災対用)

ジョブスケジューラに登録してあるジョブネットを開始する場合に呼び出される、出口プログラムについて説明します。

【Windows版】

jobnet_start_exit.bat、jobnet_start_exit.exeの順に出口プログラムが検索され、見つかった出口プログラムが呼び出されます。

jobnet_start_exit.batが見つかった場合、jobnet_start_exit.exeが存在しても呼び出されません。

利用者がこれらの出口を利用して、独自の処理を行いたい場合は、同名の出口プログラムを作成して、以下のディレクトリに格納してください。

Systemwalkerインストールディレクトリ¥MpWalker.JM¥bin

【Linux版】

jobnet_start.exitの出口プログラムが検索され、見つかった出口プログラムが呼び出されます。

利用者がこれらの出口を利用して、独自の処理を行いたい場合は、同名の出口プログラムを作成して、以下のディレクトリに格納してください。

/opt/FJSVJOBSC/bin

出口プログラムには、実行権が設定されている必要があります。実行権が設定されていないと呼び出されません。

出口プログラムに渡されるパラメタ

出口プログラムに渡されるパラメタは、以下のとおりです。

argc ：パラメタの数(7) argv[0] ：出口プログラム名 argv[1] ：自ホスト名 argv[2] ：プロジェクト名 argv[3] ：ジョブネット名 argv[4] ：ジョブネット開始日時(形式：YYYY/MM/DD hh:mm:ss) argv[5] ：再起動フラグ(起動の場合は“NO”、再起動の場合は“YES”) argv[6] ：サブシステム番号(文字列)

注意事項

- ジョブネットの実行が遅延する可能性があるため、時間のかかる処理は行わないでください。
- Windows版の場合、出口の実行ユーザは、ジョブスケジューラサービスのログオンアカウントです。【Windows版】
- Linux版の場合、出口の実行ユーザは、“root”となります。【Linux版】

プログラム例

ジョブネット開始出口のプログラム例を以下に示します。

再起動フラグが“NO”の場合は“execute”、再起動フラグが“YES”の場合は“rerun”にします。

転送処理については、お客様の運用にあわせて組み込んでください。

Windows版の例

状態情報ファイル(C:¥state¥プロジェクト名_ジョブネット名_サブシステム番号.netstart)を作成し、災対環境に転送するようにします。

```
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char *argv[]) {
    if (argc != 7) {
        return EXIT_FAILURE;
    }
    const char *program_name = argv[0];
    const char *host_name = argv[1];
    const char *project_name = argv[2];
    const char *jobnet_name = argv[3];
```

```
    const char *start_datetime = argv[4];
    const char *restart_flag = argv[5];
    const char *subsystem_number = argv[6];
    char file_path[256];
    snprintf(file_path, sizeof(file_path), "C:¥¥state¥¥%s_%s_%s.netstart", project_name, jobnet_name,
subsystem_number);
    FILE *file = fopen(file_path, "w");
    if (file == NULL) {
        perror("Failed to open file");
        return EXIT_FAILURE;
    }
    int exit_code = 0;
    if (strcmp(restart_flag, "NO") == 0) {
        fprintf(file, "NET,%s,execute,%s,%d¥n", jobnet_name, start_datetime, exit_code);
    } else {
        fprintf(file, "NET,%s,rerun,%s,%d¥n", jobnet_name, start_datetime, exit_code);
    }
    fclose(file);

    // 次にお客様の運用にあわせて転送処理を組み込みます
    ・・・転送処理・・・

    return EXIT_SUCCESS;
}
```

Linux版の例

状態情報ファイル(/var/tmp/state/プロジェクト名_ジョブネット名_サブシステム番号.netstart)を作成し、災対環境に転送するようにします。

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

int main(int argc, char *argv[]) {
    if (argc != 7) {
        return EXIT_FAILURE;
    }
    const char *program_name = argv[0];
    const char *host_name = argv[1];
    const char *project_name = argv[2];
    const char *jobnet_name = argv[3];
    const char *start_datetime = argv[4];
    const char *restart_flag = argv[5];
    const char *subsystem_number = argv[6];
    char file_path[256];
    snprintf(file_path, sizeof(file_path), "/var/tmp/state/%s_%s_%s.netstart", project_name, jobnet_name,
subsystem_number);
    FILE *file = fopen(file_path, "w");
    if (file == NULL) {
        perror("Failed to open file");
        return EXIT_FAILURE;
    }
    int exit_code = 0;
    if (strcmp(restart_flag, "NO") == 0) {
        fprintf(file, "NET,%s,execute,%s,%d¥n", jobnet_name, start_datetime, exit_code);
    } else {
        fprintf(file, "NET,%s,rerun,%s,%d¥n", jobnet_name, start_datetime, exit_code);
    }
    fclose(file);

    // 次にお客様の運用にあわせて転送処理を組み込んでください
    ・・・転送処理・・・
```

```
    return EXIT_SUCCESS;
}
```

19.11.4　ジョブネット終了出口(災対用)

ジョブスケジューラに登録してあるジョブネットが終了した場合に呼び出される、出口プログラムについて説明します。

【Windows版】

jobnet_end_exit.bat、jobnet_end_exit.exeの順に出口プログラムが検索され、見つかった出口プログラムが呼び出されます。

jobnet_end_exit.batが見つかった場合、jobnet_end_exit.exeが存在しても呼び出されません。

利用者がこれらの出口を利用して、独自の処理を行いたい場合は、同名の出口プログラムを作成して、以下のディレクトリに格納してください。

```
Systemwalkerインストールディレクトリ¥MpWalker.JM¥bin
```

【Linux版】

jobnet_end.exitの出口プログラムが検索され、見つかった出口プログラムが呼び出されます。

利用者がこれらの出口を利用して、独自の処理を行いたい場合は、同名の出口プログラムを作成して、以下のディレクトリに格納してください。

```
/opt/FJSVJOBSC/bin
```

出口プログラムには、実行権が設定されている必要があります。実行権が設定されていないと呼び出されません。

出口プログラムに渡されるパラメタ

出口プログラムに渡されるパラメタは、以下のとおりです。

```
argc ：パラメタの数(8)
argv[0] ：出口プログラム名
argv[1] ：自ホスト名
argv[2] ：プロジェクト名
argv[3] ：ジョブネット名
argv[4] ：ジョブネット終了日時(形式：YYYY/MM/DD hh:mm:ss)
argv[5] ：ジョブネットの終了コード(文字列)
argv[6] ：ジョブネットの状態(normal/notice/abend/cancel)  ※ notice:疑似正常
argv[7] ：サブシステム番号(文字列)
```

注意事項

- ジョブネットの実行が遅延する可能性があるため、時間のかかる処理は行わないでください。
- Windows版の場合、出口の実行ユーザは、ジョブスケジューラサービスのログオンアカウントです。【Windows版】
- Linux版の場合、出口の実行ユーザは、“root”となります。【Linux版】

プログラム例

ジョブネット終了出口のプログラム例を以下に示します。

転送処理については、お客様の運用にあわせて組み込んでください。

Windows版の例

状態情報ファイル(C:¥state¥プロジェクト名_ジョブネット名_サブシステム番号.netend)を作成し、災対環境に転送するようにします。

```
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char *argv[]) {
    if (argc != 8) {
        return EXIT_FAILURE;
    }
    const char *program_name = argv[0];
    const char *host_name = argv[1];
    const char *project_name = argv[2];
    const char *jobnet_name = argv[3];
    const char *end_datetime = argv[4];
    const char *exit_code = argv[5];
    const char *jobnet_status = argv[6];
    const char *subsystem_number = argv[7];
    char file_path[256];
    snprintf(file_path, sizeof(file_path), " C:¥¥state¥¥%s_%s_%s.netend", project_name, jobnet_name,
subsystem_number);
    FILE *file = fopen(file_path, "w");
    if (file == NULL) {
        perror("Failed to open file");
        return EXIT_FAILURE;
    }
    fprintf(file, "NET,%s,%s,%s,%s¥n", jobnet_name, jobnet_status, end_datetime, exit_code);
    fclose(file);

    // 次にお客様の運用にあわせて転送処理を組み込みます
    ・・・転送処理・・・

    return EXIT_SUCCESS;
}
```

Linux版の例

状態情報ファイル(/var/tmp/state/プロジェクト名_ジョブネット名_サブシステム番号.netend)を作成し、災対環境に転送するようにします。

```
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char *argv[]) {
    if (argc != 8) {
        return EXIT_FAILURE;
    }
    const char *program_name = argv[0];
    const char *host_name = argv[1];
    const char *project_name = argv[2];
    const char *jobnet_name = argv[3];
    const char *end_datetime = argv[4];
    const char *exit_code = argv[5];
    const char *jobnet_status = argv[6];
    const char *subsystem_number = argv[7];
    char file_path[256];
    snprintf(file_path, sizeof(file_path), "/var/tmp/state/%s_%s_%s.netend", project_name, jobnet_name,
subsystem_number);
    FILE *file = fopen(file_path, "w");
    if (file == NULL) {
        perror("Failed to open file");
        return EXIT_FAILURE;
    }
    fprintf(file, "NET,%s,%s,%s,%s¥n", jobnet_name, jobnet_status, end_datetime, exit_code);
    fclose(file);
```

```
    // 次にお客様の運用にあわせて転送処理を組み込みます
    ・・・転送処理・・・

    return EXIT_SUCCESS;
}
```

19.11.5　メッセージ事象発生出口(災対用)

メッセージ事象が発生した場合に呼び出される、出口プログラムについて説明します。

【Windows版】

msg_event_exit.bat、msg_event_exit.exeの順に出口プログラムが検索され、見つかった出口プログラムが呼び出されます。

msg_event_exit.batが見つかった場合、msg_event_exit.exeが存在しても呼び出されません。

利用者がこれらの出口を利用して、独自の処理を行いたい場合は、同名の出口プログラムを作成して、以下のディレクトリに格納してください。

```
Systemwalkerインストールディレクトリ¥MpWalker.JM¥bin
```

【Linux版】

msg_event.exitの出口プログラムが検索され、見つかった出口プログラムが呼び出されます。

利用者がこれらの出口を利用して、独自の処理を行いたい場合は、同名の出口プログラムを作成して、以下のディレクトリに格納してください。

```
/opt/FJSVJOBSC/bin
```

出口プログラムには、実行権が設定されている必要があります。実行権が設定されていないと呼び出されません。

出口プログラムに渡されるパラメタ

出口プログラムに渡されるパラメタは、以下のとおりです。

```
argc ： パラメタの数(11)
argv[0] ： 出口プログラム名
argv[1] ： 自ホスト名
argv[2] ： メッセージ事象発生日時(形式: YYYY MM/DD hh:mm:ss)
argv[3] ： メッセージ事象名
argv[4] ： プロジェクト名
argv[5] ： ジョブネット名
argv[6] ： 可変パラメタ
argv[7] ： イベントログのソース名
argv[8] ： イベントログのイベントID
argv[9] ： メールファイル名
argv[10] ： サブシステム番号(文字列)
```

以下に各パラメタについて説明します。

メッセージ事象発生日時：

メッセージ事象発生日時が格納されます。

形式: YYYY MM/DD hh:mm:ss

メッセージ事象名：

発生したメッセージ事象名です。必ず格納されます。

プロジェクト名：

プロジェクトが指定されてメッセージ事象が発生した場合のプロジェクト名です。

プロジェクト名が指定されていない場合は、空文字が格納されます。

ジョブネット名：

ジョブネット名が指定されてメッセージ事象が発生した場合のジョブネット名です。

ジョブネット名が指定されていない場合は、空文字が格納されます。

可変パラメタ：

可変パラメタが指定されてメッセージ事象が発生した場合の可変パラメタです。

可変パラメタが指定されていない場合は、空文字が格納されます。

イベントログのソース名：

メッセージテーブルの定義により発生したメッセージ事象の場合のイベントログのソース名です。

メッセージテーブルの定義以外で発生したメッセージ事象の場合は、空文字が格納されます。

イベントログのイベントID：

メッセージテーブルの定義により発生したメッセージ事象の場合のイベントログのイベントIDです。

メッセージテーブルの定義以外で発生したメッセージ事象の場合は、空文字が格納されます。

メールファイル名：

メール受信により発生したメッセージ事象の場合のメールファイル名です。

メール受信時に保存されるメール全体が格納されたファイル名です。メール受信によるメッセージ事象以外の場合は、空文字が格納されます。

サブシステム番号：

サブシステム番号が文字列で格納されます。

注意事項

- Windows版の場合、出口の実行ユーザは、ジョブスケジューラサービスのログオンアカウントです。【Windows版】
- Linux版の場合、出口の実行ユーザは、“root”となります。【Linux版】

プログラム例

メッセージ事象発生出口のプログラム例を以下に示します。

転送処理については、お客様の運用にあわせて組み込んでください。

Windows版の例

メッセージ事象情報ファイル(C:¥msg¥メッセージ事象名_サブシステム番号)をメッセージ事象発生履歴ファイルの形式にあわせて作成し、災対環境に転送するようにします。

```
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char *argv[]) {
    if (argc != 11) {
        return EXIT_FAILURE;
    }
    char filepath[256];
    snprintf(filepath, sizeof(filepath), "C:¥¥msg¥¥%s_%s", argv[3], argv[10]);

    FILE *fp = fopen(filepath, "w");
    if (fp == NULL) {
        perror("Failed to open file");
        return EXIT_FAILURE;
    }

    int i;
```

```
    for (i = 2; i <= 9; i++) {
        fprintf(fp, "%s", argv[i]);
        if (i < 9) {
            fprintf(fp, ",");
        }
    }
    fprintf(fp, "¥n");

    fclose(fp);

    // 次にお客様の運用にあわせて転送処理を組み込みます
    ・・・転送処理・・・

    return EXIT_SUCCESS;
}
```

Linux版の例

メッセージ事象情報ファイル(/var/tmp/msg/メッセージ事象名_サブシステム番号)をメッセージ事象発生履歴ファイルの形式にあわせて作成し、災対環境に転送するようにします。

```
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char *argv[]) {
    if (argc != 11) {
        return EXIT_FAILURE;
    }
    char filepath[256];
    snprintf(filepath, sizeof(filepath), "/var/tmp/msg/%s_%s", argv[3], argv[10]);

    FILE *fp = fopen(filepath, "w");
    if (fp == NULL) {
        perror("Failed to open file");
        return EXIT_FAILURE;
    }

    int i;
    for (i = 2; i <= 9; i++) {
        fprintf(fp, "%s", argv[i]);
        if (i < 9) {
            fprintf(fp, ",");
        }
    }
    fprintf(fp, "¥n");

    fclose(fp);

    // 次にお客様の運用にあわせて転送処理を組み込みます
    ・・・転送処理・・・

    return EXIT_SUCCESS;
}
```

第20章　ファイルリファレンス

本章では、Systemwalker Operation Managerが提供するファイルについて説明します。

20.1　ログファイル

本節では、Systemwalker Operation Managerが出力するログファイルについて説明します。

ログファイルの種類と内容

Systemwalker Operation Managerでは、下記のログファイルを出力します。

デフォルトで出力されるもの	デフォルトで出力されないもの
・ ジョブスケジューラのログ ・ ジョブ実行履歴情報のログ ・ 監査ログ	・ 稼働実績情報のログ

各ファイルについて説明します。

ジョブスケジューラのログ

ログの内容	ジョブネットおよびジョブの実行履歴(開始時刻、終了時刻、終了コード)
ログの格納先	ジョブスケジューラのデータベースディレクトリ配下 Windows版の場合：Systemwalker Operation Managerインストールディレクトリ¥MpWalker.JM¥mpjobsch¥jobdb Linux版の場合：/var/opt/FJSVJOBSC Solaris版の場合：/var/opt/FJSVJOBSC HP-UX版の場合：/opt/FHPJOBSCH/db AIX版の場合：/usr/FAIXJOBSC/db ※複数サブシステム運用の場合は、配下のJOBDBn [n:数字で1-9のサブシステム番号]
ファイル名	jobdb1.log/jobdb2.log/jobdb3.log
ログファイルのサイズ、または保存期間	1～99MB(デフォルトは3MB) ※[ジョブスケジューラ起動パラメタの定義]ウィンドウ-[データベース]シートの[ログファイルサイズ]で設定できます。
サイクリックの場合のファイルの個数	3ファイル(jobdb1.log/jobdb2.log/jobdb3.log）で変更はできません。 jobdb1.log→jobdb2.log→jobdb3.log→jobdb1.log と、ファイルサイズによりサイクリックにローテーション使用されます。 日数で保持はされません。

ジョブ実行履歴情報のログ

ログの内容	ジョブの投入/開始/終了/削除の日付、時刻およびジョブの終了コード
ログの格納先	デフォルトは以下のディレクトリ配下 【Windows版】 Systemwalker Operation Managerインストールディレクトリ¥MpWalker.JM¥mpmjessv¥hist 【UNIX版】 /var/spool/mjes/hist ※[運用情報の定義]ウィンドウ-[ログ]シートで変更できます。

	※複数サブシステム運用の場合、サブシステム番号0の場合は上記ディレクトリ、1～9の場合は配下のサブシステム番号nディレクトリ
ファイル名	yyyymmdd.log (ログファイルが作成された日付.log)
ファイル形式	バイナリ属性を含んでいるため、直接参照できません。mjsviewコマンドを使って参照してください。
ログファイルのサイズ、または保存期間	1～31日(デフォルトは7日) ※[運用情報の定義]ウィンドウ-[ログ]シートで設定できます。
保存期間を超えた場合の扱い	指定した日数を超えたファイルは削除されます。

稼働実績情報のログ

(Systemwalker Operation Managerの[環境設定]-[運用情報]で設定している場合)

ログの種類	実行されたジョブおよびジョブネットの稼働情報
ログの格納先	デフォルトは無し ※[運用情報の定義]ウィンドウ-[ログ]シートで設定できます。 ※複数サブシステム運用の場合、サブシステム番号0の場合は設定ディレクトリ、1～9の場合は配下のサブシステム番号nディレクトリ
ファイル名	yyyymmdd.csv (稼働実績情報ファイルが作成された日付.csv)
ファイル形式	CSV形式
ログファイルのサイズ、または保存期間	1～31日 ※[運用情報の定義]ウィンドウ-[ログ]シートで設定できます。
保存期間を超えた場合の扱い	指定した日数を超えたファイルは削除されます。

監査ログ

ログの種類	Systemwalker Operation Managerクライアントからの操作の記録情報
ログの格納先	デフォルトは以下のディレクトリ配下 【Windows版】 Systemwalkerインストールディレクトリ¥MpWalker.JM¥mpcmtool¥audit 【UNIX版】 /var/opt/FJSVftlo/audit ※mpsetlogsend_omgrコマンドで変更できます。
ファイル名	mp_omgr_auditYYMMDD.log (YYMMDDは操作が記録された日付)
ファイル形式	テキスト(CSV)形式
ログファイルのサイズ、または保存期間	1～999日または“unlimit”[無制限](デフォルトは31日) ※mpsetlogsend_omgrコマンドで設定できます。
保存期間を超えた場合の扱い	指定した日数を超えたファイルは削除されます。ただし、“unlimit(無制限)”を指定した場合は削除されません。

20.1.1　ジョブ実行制御の稼働実績情報ファイル

ジョブ実行制御が出力する稼働実績情報ファイルについて説明します。

稼働実績情報ファイルとは

稼働実績情報ファイルとは、実行されたジョブおよびジョブネットの稼働情報がCSV形式で出力されるファイルです。稼働実績情報を採取する場合、[運用情報の定義]ウィンドウの[ログ]シートで指定します。

稼働実績情報ファイルに出力されるCSV形式のレコードには、以下の3種類があります。

- サービスレコード
- ジョブレコード
- ステップレコード

各レコードの出力内容については、“20.1.1.1 サービスレコード”、“20.1.1.2 ジョブレコード”および“20.1.1.3 ステップレコード”を参照してください。

注意

稼働実績情報ファイルの参照について

ジョブ実行制御のサービスの実行中に“Excel”などで稼働実績情報ファイルを開いていた場合、書き込みに失敗する場合があります。

注意

ネットワークジョブでのレコード出力について

ネットワークジョブでのレコード出力は、以下のように出力されます。

JCLのネットワークジョブの場合：

実行ホスト名は、ステップ終了イベントのステップレコードに出力されます。

JCL以外のネットワークジョブの場合：

実行ホスト名、サブシステム番号は、ジョブ終了イベントのジョブレコードに出力されます。

注意

キュー名およびジョブ名について

キュー名およびジョブ名に出力される内容について、注意事項を以下に示します。

キュー名：

ジョブ投入イベントのジョブレコードおよびジョブ終了イベントのジョブレコードのキュー名には、同じキュー名が出力されます。

ジョブ名：

ジョブフォルダ、フォルダ管理外またはqsubコマンドからJCLを投入した場合、パス名および拡張子を除いたジョブファイル名をジョブ名とします。jobstart文に記述したジョブ名は無視されます。

出力規約

稼働実績情報ファイルの出力規約を以下に示します。

- 1レコードごとに改行が出力され、1レコードが1行として扱われます。
- データとデータの区切りには、“,(コンマ)”が出力されます。

- データ種別が“文字列”の情報は、“”(ダブルクォーテーション)”で囲まれます。文字列内にダブルクォーテーションが存在する場合は、ダブルクォーテーションが連続して2つ出力されます。
- データ種別が“数値”の情報は、そのままのデータが出力されます。(ダブルクォーテーションでは囲まれません。)
- 日付、時刻のデータ種別は、“文字列”として出力されます。
- 稼働実績における1レコードの最大文字数は2048バイト分に制限されています。

なお、出力データが日付および時刻である場合は、具体例を記載してあります。

20.1.1.1 サービスレコード

このレコードは、Systemwalker MpMjes サービスの起動時および停止時に出力されます。

No	データ項目名	値 (固定／可変)	出力内容	データ形式
1	レコード作成日付	可変	"2000/01/01"	文字列
2	レコード作成時刻	可変	"01:01:01"	文字列
3	レコード種別	固定	1	数値
4	ホスト名	可変	"ホスト名"	文字列
5	バージョン情報	可変	"バージョン・レベル名" (注1)	文字列
6	起動／停止種別	可変	"START" : 起動 "STOP" : 停止	文字列
7	起動モード (注2)	可変	"COLD" : 初期化モード "WARM" : 継続モード "RECOVERY" : リカバリモード	文字列
8	動作モード (注2)	可変	“CANCEL” : 実行継続モード無効 “CONTINUE” : 実行継続モード有効	文字列

注1)

現在運用中のバージョンレベル名が出力されます。

注2)

起動モードと動作モードは、起動/停止種別が、”START”の時のみ出力されます。

20.1.1.2 ジョブレコード

このレコードは、固定部と可変部から構成されます。固定部と可変部は結合して1レコードとして出力されます。

可変部は、ジョブに関するイベントの種別により以下の種類があります。

ジョブ投入イベント:

ジョブが投入された時に出力されます。

ジョブ開始イベント:

ジョブの実行が開始した時に出力されます。

ジョブ終了イベント:

ジョブの実行が終了した時に出力されます。

ジョブ消去イベント:

ジョブがSystemwalker Operation Managerを導入しているサーバから消去された時に出力されます。

ジョブ再開始イベント:

ジョブ開始イベントが出力された後、ジョブの実行依頼ができなかったために他のホストにジョブを再度依頼した時に出力されます。

【固定部】

No	データ項目名	値(固定／可変)	出力内容	データ形式
1	レコード作成日	可変	"2000/01/01"	文字列
2	レコード作成時刻	可変	"01:01:01"	文字列
3	レコード種別	固定	2	数値
4	ホスト名	可変	"ホスト名"	文字列
5	バージョン情報	可変	"バージョン・レベル名" (注1)	文字列
6	ジョブ名	可変	"ジョブ名"	文字列
7	ジョブ番号	可変	NN	数値
8	ユーザ名	可変	"ユーザ名"	文字列
9	イベント種別	可変	0 : ジョブ投入イベント 1 : ジョブ開始イベント 2 : ジョブ終了イベント 3 : ジョブ消去イベント 4 : ジョブ再開始イベント	数値
10	投入元ホスト名	可変	“投入元ホスト名”	文字列
11	ジョブ種別	可変	0 : ローカルジョブ(注2) 1 : ネットワークジョブ(注2) 2 : 旧互換負荷分散ジョブ(注3)	数値
12	ジョブ属性	可変	0 : 一般ジョブ 1 : SAP ERPジョブ 2 : NJSジョブ 3 : GLOVIAジョブ 4 : GLOVIA-BO 5 : EBSジョブ 6 : GS連携(メインフレーム)ジョブ 7 : オンライン業務ジョブ 8 : リモートマシン上のコマンド/シェルスクリプト実行コマンド(mjrmtjob) 9 : REST実行ジョブ(jobschrestexec)	数値

注1)

現在運用中のバージョンレベル名が出力されます。

注2)

分散実行の場合のジョブ種別は、ローカルで実行された場合には、“0 ：ローカルジョブ”、他ホストで実行した場合には、“1 ：ネットワークジョブ”として出力します。

注3)

旧互換負荷分散は、Windowsのみ出力されます。

【可変部：ジョブ投入イベント】

No	データ項目名	値(固定／可変)	出力内容	データ形式
1	ジョブ投入日付	可変	"2000/01/01"	文字列
2	ジョブ投入時刻	可変	"01:01:00"	文字列
3	クライアントホスト名	可変	"クライアントホスト名"	文字列
4	キュー名	可変	"キュー名"	文字列
5	コマンド名 (注1)	可変	“コマンド名 引数”	文字列
6	プロジェクト名	可変	“SampleProject”	文字列
7	ジョブネット名	可変	“サンプル”	文字列
8	サブシステム番号(注2)	可変	NN	数値

注1)

コマンド名は、ジョブとして投入したコマンド名と引数が出力されます。

注2)

他の投入元ホストから受け付けたネットワークジョブの場合、投入元ホストのサブシステム番号が出力されます。ネットワークジョブでない場合や、他の実行ホストへ依頼するネットワークジョブの場合、-1が出力されます。

【可変部：ジョブ開始イベント】

No	データ項目名	値(固定／可変)	出力内容	データ形式
1	ジョブ開始日付	可変	"2000/01/01"	文字列
2	ジョブ開始時刻	可変	"01:01:01"	文字列

【可変部：ジョブ終了イベント】

No	データ項目名	値(固定／可変)	出力内容	データ形式
1	ジョブ終了日付	可変	"2000/01/01"	文字列
2	ジョブ終了時刻	可変	"01:01:02"	文字列
3	実行ホスト名	可変	"実行ホスト名"	文字列
4	ジョブ終了コード(注1)	可変	NN	数値
5	ジョブステップ数	可変	NN	数値
6	キュー名	可変	"キュー名"	文字列
7	実行経過時間 (注2)	可変	NN	数値
8	経過時間制限値 (注2) (注3)	可変	NN	数値
9	実行予測時間 (注2) (注3)	可変	NN	数値
10	ジョブの先頭プロセスID(注4)	可変	NN	数値

No	データ項目名	値(固定／可変)	出力内容	データ形式
11	サブシステム番号(注5)	可変	NN	数値

注1)

複数ステップある場合、終了したステップのうち最大終了コードが出力されます。

注2)

秒単位で出力されます。

注3)

キューの定義またはジョブ投入時に指定されていない場合、0が出力されます。

注4)

ネットワークジョブ・JCLの場合-1が出力されます。

注5)

他の実行ホストの任意のサブシステムへ依頼するネットワークジョブの場合、実行ホストのサブシステム番号が出力されます。ネットワークジョブでない場合や、他の投入元ホストから受け付けたネットワークジョブの場合、および任意のサブシステムを指定しないネットワークジョブの場合は、-1が出力されます。

【可変部：ジョブ消去イベント】

No	データ項目名	値(固定／可変)	出力内容	データ形式
1	ジョブ消去日付	可変	"2000/01/01"	文字列
2	ジョブ消去時刻	可変	"01:01:03"	文字列

【可変部：ジョブ再開始イベント】

No	データ項目名	値(固定／可変)	出力内容	データ形式
1	ジョブ再開日付	可変	"2000/01/01"	文字列
2	ジョブ再開時刻	可変	"01:01:03"	文字列

20.1.1.3 ステップレコード

このレコードは、固定部と可変部から構成されます。固定部と可変部は結合して1レコードとして出力されます。

可変部は、ステップに関するイベントの種別により以下の種類があります。

ステップ開始イベント:

JCLファイルのステップが開始された時に出力されます。

ステップ終了イベント:

JCLファイルのステップが終了された時に出力されます。

【固定部】

No	データ項目名	値(固定／可変)	出力内容	データ形式
1	レコード作成日付	可変	"2000/01/01"	文字列
2	レコード作成時刻	可変	"01:01:01"	文字列

No	データ項目名	値(固定／可変)	出力内容	データ形式
3	レコード種別	固定	3	数値
4	ホスト名	可変	"ホスト名"	文字列
5	バージョン情報	可変	"バージョン・レベル名" (注1)	文字列
6	ジョブ名	可変	"ジョブ名"	文字列
7	ジョブ番号	可変	NN	数値
8	イベント種別	可変	0 : ステップ開始イベント 1 : ステップ終了イベント	数値
9	ステップ名	可変	“ステップ名”	文字列
10	ステップ番号	可変	NN	数値

注1)

現在運用中のバージョンレベル名が出力されます。

【可変部：ステップ開始イベント】

No	データ項目名	値(固定／可変)	出力内容	データ形式
1	ステップ開始日付	可変	"2000/01/01"	文字列
2	ステップ開始時刻	可変	"01:01:01"	文字列

【可変部：ステップ終了イベント】

No	データ項目名	値(固定／可変)	出力内容	データ形式
1	ステップ終了日付	可変	"2000/01/01"	文字列
2	ステップ終了時刻	可変	"01:01:02"	文字列
3	実行ホスト名	可変	"実行ホスト名"	文字列
4	ステップ終了コード	可変	NN	数値

20.1.2 監査ログのファイル

Systemwalker Operation Managerが提供する監査ログのファイルについて説明します。

20.1.2.1 監査ログファイル

ファイル名

mp_omgr_auditYYMMDD.log

使用用途

Systemwalker Operation Manager上で行われた操作の記録を監査ログファイルに出力できます。本ファイルはOSの管理者のみが参照可能です。

格納場所

Systemwalker Operation Manager導入時のデフォルトの設定は、以下のとおりです。格納場所は変更が可能です。変更方法は、“Systemwalker Operation Manager 導入ガイド”の“監査ログ出力の定義”を参照してください。

Windows	Systemwalkerインストールディレクトリ ¥MpWalker.JM¥mpcmtool¥audit
Solaris Linux HP-UX AIX	/var/opt/FJSVftlo/audit

ファイル形式

以下の情報が1行1レコードで記録されます。ファイルの形式はCSV形式になっており、各項目はコンマで区切られています。

項目名	最大長(Byte)	値の形式	備考
日付	29	YYYY/MM/DD△ HH:mm:SS.sss△ [+|-]UTC	“△”は半角スペース。 YYYY:西暦4桁 MM:2桁の月(01～12) DD:2桁の日(01～31) HH:2桁の時(00～23) mm:2桁の分(00～59) SS:2桁の秒(00～59) sss:3桁のミリ秒(000～999) +|-UTC:UTCからの時差、 +または-の後は、4桁の数字。
操作場所	1024	操作を行った場所を表す文字列。以下のいずれかの文字列が出力されます。 ・IPアドレス ・ホスト名 Webコンソールから操作を行った場合、“Web-GUI”の文字列を出力します。 Web APIから操作を行った場合、“Web-API”の文字列を出力します。	
実行ホスト	1024	クライアントから要求された操作を実行するホスト名、またはコマンド/API/Web APIを実行するホスト名を表す文字列。	
操作者	255	Systemwalker Operation Managerへログインしているユーザーか、サーバのOSにログインしているユーザーID。 【Windows版】 “コンピュータ名¥ユーザーID”または“ドメイ	以下の場合、“server”という文字列が出力されます。 ・ポリシー適用時 ・ジョブスケジューラのサービス/デーモンの起動および停止 ・Operation Manager共通基盤のサービス/デーモンの

項目名	最大長(Byte)	値の形式	備考
		ン名¥ユーザーID”の形式で出力されます。 なお、複数サーバ監視クライアントで複数サーバ監視を行っている場合、監視対象サーバに出力される操作者は、監視サーバにログインしたユーザにより異なります。 ・ 管理者ユーザでログイン 監視サーバにログインした管理者権限のユーザが出力されます。 ・ 一般ユーザでログイン 監視対象サーバへ接続したユーザが出力されます。	起動および停止 ・カレンダのサービス/デーモンの起動および停止 ・カレンダの自動反映機能利用時 ・電源スケジュールによるサーバの自動切断時 ・年変わり時によるカレンダ/システムカレンダの情報更新時 ・スケジュール/未来のスケジュールの実行時 ・スケジュールサーバの追加/編集/解除 ・マスタスケジュール管理による仮想時間の設定/解除
操作種別	512	操作した内容を分類するための文字列。	操作種別は以下の形式で出力されます。 [操作]_[追加情報] 詳細は“Systemwalker Operation Manager 運用ガイド”の“監査ログの検索キーワード一覧”を参照してください。
操作対象	4096	操作対象を特定する情報が出力されます。 複数の項目が出力される場合は、セミコロンで区切られて出力されます。	詳細は“Systemwalker Operation Manager 運用ガイド”の“監査ログの検索キーワード一覧”を参照してください。
操作内容	4096	実行内容が出力されます。 例:ジョブAを停止します。	コマンド/API/Web APIによる定義変更、ジョブの操作が行われた場合、以下のように出力されます。 コマンド実行の場合: コマンドを実行します。(COMMAND=＜コマンドおよびパラメタ文字列＞) APIの場合: APIを実行します。(API=＜API名＞) Web APIの場合： Web APIを実行します。(Web API=<Web API名>)
実行結果	1	B:BEGIN(操作の開始) S:SUCCESS(成功) F:FAIL(失敗) C:CANCEL	BEGIN(操作の開始)を表す“B”は、コマンドなどで処理に時間がかかる場合に処理開始を意味する情報として出力されます。

項目名	最大長(Byte)	値の形式	備考
			BEGINが出力された場合、そのログの後に処理結果を表すログがさらに1行出力され、SUCCESS、FAILまたはCANCELが出力されます。
コンポーネント	256	操作対象のコンポーネント名(機能名)が出力されます。	出力される情報と対応する機能名は以下のとおりです。 MPCMTOOLO:共通ツール MPFWSEC:ACLマネージャ MPJMCAL:カレンダ・電源制御 MPJOBSCH:ジョブスケジューラ MPJMSRV:Operation Manager共通基盤 MPMJES:ジョブ実行制御 MPSTEM:マスタスケジュール管理 MPNJSOSV:業務連携
追加情報	1024	コマンド実行時のプロセスIDや、接続したクライアントのバージョンなど、付加的な詳細情報がある場合に出力されます。	サーバへのログイン時は、接続したクライアントのログインと対に出力する認証のログにて、以下の情報が出力されます。 V13.0.0またはV13.1.0のクライアントから接続した場合: CL_VERSION=V13.0 or V13.1 V13.2.0以降のクライアントから接続した場合: CL_VERSION=V13.2 or later
リザーブ領域	－	－	使用されません。

[追加情報]

操作種別の補足情報です。主なキーワードは以下のとおりです。

追加情報	説明
CALENDAR	カレンダ情報
CARRYIEDOVERJOBNET	持ち越しジョブネット
DAEMON/SERVICE	デーモン／サービス
GROUP	グループ
JOB	ジョブ
JOBNET	ジョブネット
LGMANAGE	パスワード管理簿
MESSAGE	メッセージ事象

追加情報	説明
OMGR_USER	Operation Managerユーザ
PERMISSION	アクセス権
PROJECT	プロジェクト
QUEUE	キュー
SCHEDULES	スケジュール
SERVER	サーバ

参考

監査ログを検索する場合のキーワードについては、“Systemwalker Operation Manager 運用ガイド”の“監査ログの検索キーワード一覧”を参照してください。

参照

mpsetlogsend_omgr

使用例

```
"2006/09/26 11:25:12.672
+0900","10.90.100.100","host1","user1","ADD_PROJECT","SUBSYSTEM=0;PROJECT=PRJ01;","プロジェクトを追加します。",S,"MPJOBSCH","PID=8045;",
"2006/09/26 11:30:41.835
+0900","10.90.100.100","host1","user1","ADD_JOBNET","SUBSYSTEM=0;PROJECT=PRJ01;JOBNET=JOBNET01;","ジョブネットを追加します。",S,"MPJOBSCH","PID=8045;",
"2006/09/26 11:30:45.390
+0900","10.90.100.100","host1","user1","START_JOBNET","SUBSYSTEM=0;PROJECT=PRJ01;JOBNET=JOBNET01;","ジョブネットを起動します。",S,"MPJOBSCH","PID=8045;",
```

20.2 メッセージ事象発生履歴ファイル

本節では、メッセージ事象発生時の履歴情報ファイルについて説明します。

20.2.1 メッセージ事象発生履歴ファイル

メッセージ事象が発生した場合に、Systemwalker Operation Managerが記録するファイルです。

本ファイルは、ジョブスケジューラサービス/デーモンが起動されている場合に更新されます。

ファイルサイズは最大99MBです。99MBに達した場合は、3ファイルが順番に切り替わり、古いファイルが順番に破棄されます。

ファイル名

```
msgevent.log
msgevent0.log
msgevent1.log
```

使用用途

メッセージ事象の発生状況を確認するために参照します。

システムメンテナンスなどで、メッセージ事象待ち合わせジョブが一時的に停止していたタイミングが発生した場合に、待ち合わせるべきだったメッセージ事象の発生状況を確認し、後続ジョブを実行すべきかどうかを判断することができます。

格納場所

Windows	Systemwalkerインストールディレクトリ¥MpWalker.JM¥mpjobsch¥jobdb 複数サブシステム運用でサブシステム1～9の場合 Systemwalkerインストールディレクトリ¥MpWalker.JM¥mpjobsch¥jobdb*n* ※“*n*”はサブシステム番号
Solaris Linux	/var/opt/FJSVJOBSC 複数サブシステム運用でサブシステム1～9の場合 /var/opt/FJSVJOBSC/JOBDB*n* ※“*n*”はサブシステム番号
HP-UX	/opt/FHPJOBSCH/db 複数サブシステム運用でサブシステム1～9の場合 /opt/FHPJOBSCH/db/JOBDB*n* ※“*n*”はサブシステム番号
AIX	/usr/FAIXJOBSC/db 複数サブシステム運用でサブシステム1～9の場合 /usr/FAIXJOBSC/db/JOBDB*n* ※“*n*”はサブシステム番号

ファイル形式

以下の情報が1行1レコードで記録されます。ファイルの形式はCSV形式になっており、各項目はコンマで区切られています。

項目名	サイズ(Byte)	出力内容	備考
日付	19	メッセージ事象が発生した日付です。 YYYY MM/DD hh:mm:ss 必ず出力されます。	YYYY:西暦4桁 MM:2桁の月(01～12) DD:2桁の日(01～31) hh:2桁の時(00～23) mm:2桁の分(00～59) ss:2桁の秒(00～59)
メッセージ事象名	12	発生したメッセージ事象名です。 必ず出力されます。	
プロジェクト名	50	プロジェクト名が指定されてメッセージ事象が発生した場合のプロジェクト名です。 プロジェクト名が指定されていない場合は、出力されません。	

項目名	サイズ (Byte)	出力内容	備考
ジョブネット名	50	ジョブネット名が指定されてメッセージ事象が発生した場合のジョブネット名です。 ジョブネット名が指定されていない場合は、出力されません。	
可変パラメタ	200	可変パラメタが指定されてメッセージ事象が発生した場合の可変パラメタです。 可変パラメタが指定されていない場合は、出力されません。	
イベントログのソース名	64	メッセージテーブルの定義により発生したメッセージ事象の場合のイベントログのソース名です。 メッセージテーブルの定義以外で発生したメッセージ事象の場合は、出力されません。	
イベントログのイベントID	5	メッセージテーブルの定義により発生したメッセージ事象の場合のイベントログのイベントIDです。 メッセージテーブルの定義以外で発生したメッセージ事象の場合は、出力されません。	
メールファイル名	256	メール受信により発生したメッセージ事象の場合のメールファイル名です。 メール受信時に保存されるメール全体が格納されたファイル名です。 メール受信によるメッセージ事象以外の場合は、出力されません。	

注意事項

本ファイルは、排他オープンするアプリケーションでは開かないでください。排他オープンするアプリケーションで本ファイルを開いたまま新しいメッセージ事象が発生すると、メッセージ事象の発生処理が遅延したり、メッセージ事象の情報が書き込めなくなります。

使用例

プロジェクト名“TEST”、ジョブネット名“TEST01”に対してメッセージ事象“MESSAGE”が発生した場合のレコード内容

```
"2014 01/02 03:04:06","MESSAGE","TEST","TEST01",,,,
```

20.3 ジョブネット変数情報ファイル

本節では、ジョブネット変数情報ファイルについて説明します。

20.3.1 ジョブネット変数情報ファイル

ジョブネット変数の取り込み結果やジョブ実行時の変数情報は、ジョブネット変数情報ファイルに格納されます。
ジョブネット変数情報ファイルには、以下の2種類があります。

- ジョブネット終了時点での変数一覧を格納したファイル [拡張子がtvrのファイル]

・ ジョブネット内の各ジョブが取り込んだ変数と実行時に利用した変数一覧[拡張子がtvhのファイル]

ファイル名

【Windows版】

プロジェクト名_ジョブネット名_ccccccccccccccc.tvr プロジェクト名_ジョブネット名_ccccccccccccccc.tvh

※ccccccccccccccccは任意の文字

【UNIX版】

プロジェクト名_ジョブネット名_nn.tvr プロジェクト名_ジョブネット名_nn.tvh

※nnは任意の数字

使用用途

ジョブネットやジョブが異常終了した場合などに、ジョブネット変数について確認するために参照します。以下の情報を確認することが可能です。

・ 各ジョブがジョブの出力情報から取り込んだ変数

・ 各ジョブがジョブ実行時に利用した変数

・ ジョブネット終了時点における変数の一覧

格納場所

Windows	Systemwalkerインストールディレクトリ¥MpWalker.JM¥mpjobsch¥jobdb¥jobschtvr 複数サブシステム運用でサブシステム1～9の場合 Systemwalkerインストールディレクトリ¥MpWalker.JM¥mpjobsch¥jobdb*n*¥jobschtvr ※“*n*”はサブシステム番号
Solaris Linux	/var/opt/FJSVJOBSC/jobschtvr 複数サブシステム運用でサブシステム1～9の場合 /var/opt/FJSVJOBSC/JOBDB*n*/jobschtvr ※“*n*”はサブシステム番号
HP-UX	/opt/FHPJOBSCH/db/jobschtvr 複数サブシステム運用でサブシステム1～9の場合 /opt/FHPJOBSCH/db/JOBDB*n*/jobschtvr ※“*n*”はサブシステム番号
AIX	/usr/FAIXJOBSC/db/jobschtvr 複数サブシステム運用でサブシステム1～9の場合 /usr/FAIXJOBSC/db/JOBDB*n*/jobschtvr ※“*n*”はサブシステム番号

ファイル形式

本ファイルはテキスト形式であり、テキストエディタなどを用いて参照することができます。

[拡張子がtvrのファイル]

ジョブネット終了時点でのジョブネット変数一覧を格納したファイルです。

階層化されたジョブネットの場合、第2階層以下の子ジョブネット終了時点における変数の一覧は、第1階層ジョブネットの本ファイル内に格納されます。

```
[TRANSFER_VARIABLE]        (1)
JOB.NAME=VALUE             (2)
JOB.A=B                    (2)
```

(1)ジョブネット終了時点における変数の一覧を示すセクション名です。固定の文字列です。
(2)“変数名=値”の形式です。ジョブネット終了時点における変数の一覧です。

[拡張子がtvhのファイル]

ジョブネット内の各ジョブが取り込んだ変数と実行時に利用した変数一覧を格納したファイルです。

```
[VARIABLE_OUTPUT1]         (1)
JOB.NAME=VALUE             (2)
JOB.A=B                    (2)
[VARIABLE_INPUT2]          (3)
JOB.A=B                    (4)
[VARIABLE_OUTPUT2]         (1)
JOB.NAME=VALUE             (2)
JOB.A=C                    (2)
```

(1)各ジョブがジョブの出力情報から取り込んだ変数を示すセクション名です。
“VARIABLE_OUTPUT”までは固定の文字列、後ろの数字はジョブスケジューラが管理しているジョブ番号です。
ジョブが再起動されると、前回の起動(または再起動)のセクションは削除されて、最後尾に新しいセクションが追加されます。
(2)“変数名=値”の形式です。各ジョブがジョブの出力情報から取り込んだ変数の一覧です。
(3)各ジョブがジョブ実行時に利用した変数を示すセクション名です。
“VARIABLE_INPUT”までは固定の文字列、後ろの数字はジョブスケジューラが管理しているジョブ番号です。
ジョブが再起動されると、前回の起動(または再起動)のセクションは削除されて、最後尾に新しいセクションが追加されます。
(4)“変数名=値”の形式です。各ジョブがジョブ実行時に利用した変数の一覧です。

注意事項

- 本ファイルは、ジョブネット起動時に削除され、ジョブネット変数情報は前回の1回分だけ格納されます。
- 以下のジョブネット変数はこのファイルに格納されません。
  - 可変パラメタ(@.VPARAM@)
  - プロジェクト名(@.PROJECT@)
  - ジョブネット名(@.JOBNET@)
  - ジョブ名(@.JOBNAME@)
  - サブシステム番号(@.SUB_SYSTEM@)
  - マスタスケジュール管理機能利用時の、マスタプロジェクト名(@.MASTER_PROJECT@)
- 本ファイルは、ジョブネット実行中にSystemwalker Operation Managerが随時更新するファイルです。実行中のジョブネットに対するファイルに直接アクセスしないでください。ジョブネット実行中に直接アクセスすると、ジョブネットが異常終了する場合があります。

使用例

[拡張子がtvrのファイル]

```
[TRANSFER_VARIABLE]
JOB.CF_FILENAME=/var/tmp/project.txt
JOB.CF_CHKTYPE=fwait
JOB.CF_FILESIZE=1024
```

```
JOB.CF_FILEYEAR=2014
JOB.CF_FILEDATE=0123
JOB.CF_FILETIME=1234
```

[拡張子がtvhのファイル]

```
[VARIABLE_OUTPUT1]
JOB.CF_FILENAME=/var/tmp/project.txt
JOB.CF_CHKTYPE=fwait
JOB.CF_FILESIZE=1024
JOB.CF_FILEYEAR=2014
JOB.CF_FILEDATE=0123
JOB.CF_FILETIME=1234
[VARIABLE_INPUT2]
JOB.CF_FILENAME=/var/tmp/project.txt
JOB.CF_FILESIZE=1024
```

20.4 実行ユーザ制限のファイル【UNIX版】

本節では、Systemwalker Operation Managerが提供する実行ユーザ制限機能のファイルについて説明します。

20.4.1 実行ユーザ制限リスト

ジョブの実行ユーザを制限したい場合に作成する実行ユーザ制限リストについて、viやメモ帳などのエディタを使用して設定する方法を説明します。

ファイル名

```
mjexuser
```

使用用途

本ファイルには、ジョブの実行を許可するユーザ名を記述してください。

設定手順については、“Systemwalker Operation Manager 導入ガイド”を参照してください。

格納場所

Solaris HP-UX AIX Linux	/etc/mjes

EE GEE

複数サブシステム運用を行っている場合は以下の場所に格納します。

/etc/mjes/mjes*n* (*n*：サブシステム番号)

ファイル形式

```
<ユーザ名>
  :
  :
```

パラメタ

<ユーザ名>

ジョブの実行を許可するユーザ名を1行に1ユーザを記述します。

- 大文字と小文字は区別されます。
- 記述できるユーザ数、1行あたりの文字数は以下のとおりです。
  - ユーザ数:256以内(257番目以降は無効になります。)
  - 1行あたりの文字数:1024バイト以内(1025バイト以上指定した場合、1行全体が無効になります。)
- ユーザ名の前には空白(半角スペース、タブ)のみ入力できます。
- ユーザ名の後ろの空白およびタブは無視されます。
- 空行は無視されます。
- コメントを記述する場合は#記号を使用します。#記号以降がコメントとして扱われます。
- 空ファイルを作成するとジョブを実行できるユーザが存在しなくなります。

注意事項

- Systemwalker Operation Managerの再起動時に再実行となるジョブが存在する場合、実行ユーザが実行ユーザ制限リストから削除されていても、ジョブは再実行されます。
- セキュリティ上、安全を確保するため、実行ユーザ制限リストにアクセス権を設定してください。

  所有者：root モード：600
- 以下の例のように実行ユーザ制限リストのユーザ名が重複している場合、username1とusername2がジョブを実行できるユーザとして指定されたことになります。ただし、設定されたユーザ数は重複したユーザ名もカウントされ、ユーザ数は3となります。

```
username1
username2
username1
```

使用例

```
username1
username2
username3
```

20.5 インポート/エクスポート実行結果ファイル

本節では、Systemwalker Operation Managerの定義情報をGUIからインポート/エクスポートした場合に出力される実行結果ファイルについて説明します。

20.5.1 インポート実行結果ファイル

ファイル名

```
impinfo.txt
```

使用用途

インポート実行結果ファイルは、指定されたCSVファイルのインポート結果が出力されるファイルです。

格納場所

Windows	Systemwalker Operation Manager クライアントインストール先ディレクトリ¥MPWALKER.JM¥mpjmcl¥work¥jmimpexp¥

ファイル形式

```
サブシステム番号
プロジェクト名

インポート成功/失敗のメッセージ
インポート情報メッセージ
```

パラメタ

インポート成功/失敗のメッセージ

定義体の登録に成功したか失敗したかのメッセージが出力されます。

インポート情報メッセージ

失敗した場合に、インポートの処理結果が出力されます。

注意事項

本ファイルは、排他オープンするアプリケーションでは開かないでください。排他オープンするアプリケーションで本ファイルを開いたまま、インポートを行うと、インポート情報が書き込めなくなります。

使用例

```
サブシステム番号=0
プロジェクト名=営業

定義体の登録に失敗しました。
MpJobsch:エラー :0350: インポートが異常終了しました ジョブネット名=jobnet1 オペランド=starttime コード=0084
```

20.5.2 インポート実行結果ファイル(V13.9.0以前のサーバに接続した場合)

本項では、以下の場合にインポートをすると出力される実行結果ファイルについて説明します。

- Systemwalker Operation Manager V13.9.0以前のサーバに接続した場合
- クライアント動作環境ファイル (Jobschr.ini)で、キー名“ImportExportLargeScale”の値を0に設定した場合
- [Systemwalker Operation Manager インポート]ウィンドウで[CSVファイルを一括転送して処理を高速化する]のチェックをしない場合

ファイル名

```
impinfo.txt
```

使用用途

インポート実行結果ファイルは、指定されたジョブネット定義CSVファイルまたは、グループ定義CSVファイルのインポート結果が出力されるファイルです。

格納場所

Windows	Systemwalker Operation Manager クライアントインストール先ディレクトリ¥MPWALKER.JM¥mpjmcl¥work¥jmimpexp¥

ファイル形式

```
サブシステム番号
プロジェクト名

ジョブネット定義CSVファイル名を含むパスまたは、グループ定義CSVファイル名を含むパス
インポート情報メッセージ

ジョブネット定義CSVファイル名を含むパスまたは、グループ定義CSVファイル名を含むパス
インポート情報メッセージ
        :
        :
```

※ “インポート情報メッセージ”は複数行の場合があります。

パラメタ

ジョブネット定義CSVファイル名または、グループ定義CSVファイル名を含むパス

インポートしたジョブネット定義CSVファイル名または、グループ定義CSVファイル名が出力されます。

インポート情報メッセージ

指定されたジョブネット定義CSVファイルまたは、グループ定義CSVファイルごとにインポートの処理結果が出力されます。

注意事項

本ファイルは、排他オープンするアプリケーションでは開かないでください。排他オープンするアプリケーションで本ファイルを開いたまま、インポートを行うと、インポート情報が書き込めなくなります。

使用例

```
C:¥temp¥test¥test1.csv
MpJobsch:情報 :2300:jobschsetnet コマンドが正常に終了しました
```

20.5.3　エクスポート実行結果ファイル

ファイル名

```
expinfo.txt
```

使用用途

エクスポート実行結果ファイルは、指定されたジョブネットのエクスポート結果が出力されるファイルです。

格納場所

Windows	Systemwalker Operation Manager クライアントインストール先ディレクトリ¥MPWALKER.JM¥mpjmcl¥work¥jmimpexp¥

ファイル形式

```
サブシステム番号
プロジェクト名
```

```
エクスポート成功/失敗のメッセージ
エクスポート情報メッセージ
```

パラメタ

エクスポート成功/失敗のメッセージ

定義体の取得に成功したか失敗したかのメッセージが出力されます。

エクスポート情報メッセージ

失敗した場合に、エクスポートの処理結果が出力されます。

注意事項

本ファイルは、排他オープンするアプリケーションでは開かないでください。排他オープンするアプリケーションで本ファイルを開いたまま、エクスポートを行うと、エクスポート情報が書き込めなくなります。

使用例

```
サブシステム番号=0
プロジェクト名=営業

定義体の取得に失敗しました。
MpJobsch:エラー :0483:ジョブネット定義CSVファイルの書き込みに失敗しました
```

20.5.4 エクスポート実行結果ファイル(V13.9.0以前のサーバに接続した場合)

本項では、以下の場合にエクスポートをすると出力される実行結果ファイルについて説明します。

- Systemwalker Operation Manager V13.9.0以前のサーバに接続した場合
- クライアント動作環境ファイル(Jobschr.ini)で、キー名“ImportExportLargeScale”の値を0に設定した場合
- [Systemwalker Operation Manager エクスポート]ウィンドウで[CSVファイルを一括転送して処理を高速化する]のチェックをしない場合

ファイル名

```
expinfo.txt
```

使用用途

エクスポート実行結果ファイルは、指定されたジョブネットまたは、グループのエクスポート結果が出力されるファイルです。

格納場所

Windows	Systemwalker Operation Manager クライアントインストール先ディレクトリ¥MPWALKER.JM¥mpjmcl¥work¥jmimpexp¥

ファイル形式

```
サブシステム番号
プロジェクト名

ジョブネット名または、グループ名
```

```
CSVファイル名(CSVファイル名を含むパス)
エクスポート情報メッセージ
        :
        :
ジョブネット名または、グループ名
CSVファイル名(CSVファイル名を含むパス)
エクスポート情報メッセージ

サブシステム番号
プロジェクト名

ジョブネット名または、グループ名
CSVファイル名(CSVファイル名を含むパス)
エクスポート情報メッセージ
        :
        :
```

※ “エクスポート情報メッセージ”は複数行の場合があります。

パラメタ

プロジェクト名

エクスポートで指定したジョブネットまたはグループが属しているプロジェクト名が出力されます。

プロジェクト名フォルダ

エクスポート先に指定したフォルダ名にサブシステム番号とプロジェクト名が付加されたフォルダ名が出力されます。

メッセージ

プロジェクト名フォルダの作成結果が出力されます。

ジョブネット名または、グループ名

エクスポートしたジョブネット名または、グループ名が出力されます。

CSVファイル名

エクスポートで指定したジョブネットまたは、グループに対するCSVファイル名がフルパスで出力されます。

エクスポート情報メッセージ

指定されたジョブネットまたは、グループごとにエクスポートの処理結果が出力されます。

注意事項

本ファイルは、排他オープンするアプリケーションでは開かないでください。排他オープンするアプリケーションで本ファイルを開いたまま、エクスポートを行うと、エクスポート情報が書き込めなくなります。

使用例

```
プロジェクト名=ProjectA
プロジェクト名フォルダ=C:¥CSV¥0¥ProjectA
プロジェクト名フォルダの作成に成功しました。

ジョブネット名=JNetA
CSVファイル名= C:¥CSV¥0¥ProjectA¥JNetA.csv
正常終了しました。
```

20.6　クラスタ運用の定義ファイル

本節では、Systemwalker Operation Managerをクラスタシステムで運用するための定義ファイルについて説明します。

20.6.1　calcphost.def カレンダ反映先ホスト定義ファイル

ファイル名

calcphost.def

使用用途

カレンダ反映先ホスト定義ファイルは、カレンダの自動反映機能が有効の状態でカレンダの変更を行ったときに、カレンダ情報を反映させるホストを定義するファイルです。

格納場所

Windows	Systemwalker Operation Managerインストールディレクトリ¥MPWALKER.JM¥mpjmcal¥etc
Solaris	/var/opt/FJSVjmcal/etc
HP-UX	/opt/FHPjmcal/etc
Linux	/var/opt/FJSVjmcal/etc
AIX	/opt/FAIXjmcal/etc

ファイル形式

本定義ファイルの記述形式は、以下のとおりです。

1行ごとに1つの反映先のホスト名、またはIPアドレスを指定し、改行を入れてください。

IPアドレスで指定することを推奨します。

項目名	最大長(バイト)	備考
ホスト名	128	
IPアドレス	45	物理IPアドレスをIPv4形式またはIPv6形式で指定してください

- 登録できるホスト名またはIPアドレスは10個までです。10個を超えた場合、ファイルの先頭から10個までのホスト名またはIPアドレスが有効となります。
- 自ノードのホスト名、またはIPアドレスを含め、登録してください。
- 定義ファイルの保存時は、Systemwalkerのインストール時の文字コード系と同一の文字コード系で保存してください。
- 初期ファイルの1行目にはファイルフォーマットバージョンが記載されています。この1行目の編集は禁止です。

注意事項

- 本定義ファイルは、Systemwalker Operation Managerの導入時には存在しません。calsetcluster カレンダ自動反映セットアップコマンドで、カレンダの自動反映機能を有効にする設定を行った場合に、初期状態のファイルが作成されます。詳細は、“3.3 calsetcluster カレンダ自動反映セットアップコマンド”を参照してください。
- 本定義ファイルは、ポリシー配付の対象外です。

- 本定義ファイルは、カレンダの同期をとる必要があるすべてのノードに対して、同一のファイルを使用してください。他のホストから反映依頼された場合、自ノードの反映先リスト中のホストに該当しない場合、反映は拒否されます。
- 複数のネットワークカードを使用するなど、複数の物理IPアドレスを使用している場合は、物理IPアドレスを定義ファイルに記載してください。

使用例

カレンダの編集が行われた場合、ホスト名：host1／host2、IPアドレス：192.0.2.0(IPv4形式の場合)／2001:db8::123:4567:89ab:cdef(IPv6形式の場合)のホストにカレンダ情報を自動反映する場合の定義例を以下に示します。

```
@V11L10
#Specify host names or IP addresses on the subsequent lines.
host1
host2
192.0.2.0
2001:db8::123:4567:89ab:cdef
```

1行目：

初期状態から存在します。定義ファイルのバージョンレベルです。編集は禁止です。

2行目：

初期状態から存在します。編集可能です。1カラム目が“#”の場合は、コメントとなります。

20.6.2 cluster_ip.csv 定義ファイル

ファイル名

cluster_ip.csv

使用用途

本ファイルは、Systemwalker Operation Manager クライアントの格納場所に作成することにより、有効になります。

Systemwalker Centric Managerとの画面連携によって、Systemwalkerコンソールから異常終了したジョブネットの監視画面が起動されます。

ジョブネットの実行先サーバがクラスタ運用されている場合、本定義ファイルに論理IPアドレス、物理IPアドレス、サブシステム番号の対応関係を定義しておくことで、ジョブネットの異常終了発生時の実行サーバ(物理IPアドレス)に関係なく、常に現在の運用系のサーバ(論理IPアドレス)に接続して、ジョブネットの監視画面が起動されるようになります。

格納場所

Windows	Systemwalker Operation Managerインストールディレクトリ¥MpWalker.JM¥mpjmcl¥etc

ファイル形式

項目名	最大長(バイト)	値の形式
論理IPアドレス	15	IPアドレス(xxx.xxx.xxx.xxx) (注1、注2、注3)
サブシステム番号	1	0～9(注1、注2)

項目名	最大長(バイト)	値の形式
物理IPアドレス	15	IPアドレス(xxx.xxx.xxx.xxx) (注1、注3)

注1)

先頭が“#”の行は、コメントとなります。

注2)

論理IPアドレスまたはサブシステム番号が、上の行と同じ場合、記述を省略することもできます。

注3)

IPアドレスが2桁、1桁の場合には、先頭に0をつけない形式で記入してください。例えば、“010.003.015.005”でなく、“10.3.15.5”と記入してください。

注意事項

cluster_ip.csv定義ファイルはSystemwalker Operation Managerのバックアップ、リストアの対象になりません。必要な場合は手動でバックアップ、リストアを行ってください。

使用例

[1：1運用待機の場合]

```
#論理IPアドレス,サブシステム番号,物理IPアドレス
172.23.73.10,0,172.23.73.15
,,172.23.73.16
```

[相互待機の場合]

```
#論理IPアドレス,サブシステム番号,物理IPアドレス
172.23.73.10,1,172.23.73.15
,,172.23.73.16
172.23.73.20,2,172.23.73.15
,,172.23.73.16
```

20.7 Webサーバポートの定義ファイル

本節では、Systemwalker Operation ManagerサーバとWebサーバ(Webコンソール/Web API)の通信に利用するポート番号の定義ファイルについて説明します。

20.7.1 Webサーバポート定義ファイル

ファイル名

任意(*.*)

使用用途

mpowebconfig(Webサーバ設定変更コマンド)において、ポート番号の設定を出力/反映させるために使用するファイルです。

本ファイルは、mpowebconfig(Webサーバ設定変更コマンド)の-o filenameオプションまたは-i filenameオプションのfilenameに指定します。

格納場所

任意

ファイル形式

要素名parameterの属性nameに続く各値に、割り当てるポート番号を設定してください。下記のXML形式のファイルです。斜体の箇所が編集可能です。

```
<?xml version="1.0" encoding="UTF-8"?>
<web-server>
    <ahs>
        <parameter name="http.port">9900K</parameter>
    </ahs>
    <fjjavaee>
        <domain>
            <jesetup>
                <parameter name="--domain.instancePort">9901K</parameter>
                <parameter name="--http.ssl.port">9902K</parameter>
                <parameter name="--domain.adminPort">9903K</parameter>
                <parameter name="--jms.port">9904K</parameter>
                <parameter name="--orb.listener.port">9905K</parameter>
                <parameter name="--orb.mutualauth.port">9906K</parameter>
                <parameter name="--orb.ssl.port">9907K</parameter>
                <parameter name="--domain.jmxPort">9908K</parameter>
            </jesetup>
            <setup/>
        </domain>
    </fjjavaee>
</web-server>
```

パラメタ

http.port:

Systemwalker Operation ManagerサーバとWebサーバ(Webコンソール/Web API)が通信するために利用するポート番号の定義です。

--domain.instancePort:
--http.ssl.port:
--domain.adminPort:
--jms.port:
--orb.listener.port:
--orb.mutualauth.port:
--orb.ssl.port:
--domain.jmxPort:

Systemwalker Operation Managerサーバが内部通信に利用するポート番号の定義です。

参照

mpowebconfig(Webサーバ設定変更コマンド)

20.8 実行サブシステム名の定義ファイル

本節では、実行サブシステム名定義ファイルについて説明します。

20.8.1 実行サブシステム名定義ファイル

ファイル名

mjsubsysname.txt

使用用途

ネットワークジョブの実行サーバのホスト名に、実行サブシステム名を指定する場合に定義するファイルです。

格納場所

Windows	Systemwalker Operation Managerインストールディレクトリ ¥MpWalker.JM¥mpmjessv¥user 複数サブシステム運用を行っている場合 Systemwalker Operation Managerインストールディレクトリ ¥MpWalker.JM¥mpmjessv¥mjes*n*¥user (*n*：サブシステム番号)
Soalris Linux	/etc/mjes 複数サブシステム運用を行っている場合 /etc/mjes/mjes*n* (*n*：サブシステム番号)

ファイル形式

```
<実行サブシステム名>,<ホスト名>,<サブシステム番号>
  :
  :
```

- 実行サブシステム名は100個まで定義することが可能です。101個目以降は無視されます。
- 空行は無視されます。
- 形式が不正な行は無視されます。
- コメントを記述する場合は“;”(セミコロン)を使用します。“;”以降がコメントとして扱われます。
- 1行は1024バイト以内で記述してください。1025バイトを超えた場合、1行全体が無視されます。

パラメタ

<実行サブシステム名>

ネットワークジョブの実行サーバのホスト名に指定する実行サブシステム名を記述します。

- 64バイト以内で指定します。
- 半角英数字および、以下を除く記号のみ指定可能です。

  半角空白 タブ / ¥ ; * ? " < > | ( ) ,
- 前後の空白(半角スペース、タブ)は無視されます。

<ホスト名>

ネットワークジョブの実行サーバのホスト名に実行サブシステム名を指定した場合に、ジョブが投入されるホスト名を記述します。

- 64バイト以内で指定します。
- 半角英数字および、以下を除く記号のみ指定可能です。

  半角空白 タブ / ¥ ; * ? " < > | ( ) ,
- 前後の空白(半角スペース、タブ)は無視されます。

<サブシステム番号>

ネットワークジョブの実行サーバのホスト名に実行サブシステム名を指定した場合に、ジョブが投入されるホストのサブシステム番号を記述します。

- 0～9の範囲で指定します。
- 複数サブシステム運用を行っていない場合は0を指定します。
- 前後の空白(半角スペース、タブ)は無視されます。

注意事項

同じ名前の実行サブシステム名を複数行定義した場合、最初に記述された定義が有効になります。定義できる数の上限は、重複する定義もカウントされます。

実行サブシステム名を指定したネットワークジョブは、定義したホスト名およびサブシステム番号に置き換えられてジョブが投入されます。存在しないホスト名を定義するなど、パラメタの形式以外に関する定義の誤りは、ジョブの実行時にエラーになります。

使用例

```
execsvA,vm1,2
execsvB,vm2,3
```

20.9 クラウドサービス連携の定義ファイル【Windows版】【Linux版】

REST実行ジョブがクラウドサービスへ接続するための接続情報ファイルについて説明します。

20.9.1 接続情報ファイル

ファイル名

```
nnnnn.json
```

※nnnnnは任意の文字

使用用途

REST実行ジョブがクラウドサービスに接続するために必要なパラメタを記載する接続情報ファイルです。REST実行ジョブを利用する場合には必須です。REST実行ジョブ登録時には、本ファイルをフルパスで指定します。

格納場所

任意

ファイル形式

本ファイルはJSON形式、文字コードはUTF-8で記載します。string型の値にダブルクォーテーションやバックスラッシュ、および、制御文字などを含める場合、JSONのフォーマットに従いエスケープした文字列を指定してください。

- requestオブジェクト

  実行するAPIに接続するためのパラメタを指定します。

- monitoringオブジェクト

  APIの処理の完了を待ち合わせるために、監視リクエストを送る場合のパラメタを指定します。

- terminationオブジェクト

  REST実行ジョブ強制終了時に、実行中のAPIに強制終了リクエストを送る場合のパラメタを指定します。

```
{
"request": {
    "url": "接続URL",
```

```
    "method": "HTTPメソッド",
    "authName": "認証情報名",
    "proxyAuthName": "プロキシサーバの認証情報名",
    "query": {クエリパラメタ},
    "header": {HTTPリクエストヘッダ},
    "body": "HTTPリクエストメッセージボディ",
    "resHeader": "HTTPレスポンスヘッダ",
    "resBody": "HTTPレスポンスボディ",
    "timeout":  接続タイムアウト時間,
    "takeoverKeys":  {レスポンスボディの値から監視、強制終了リクエストに引継ぐキー},
    "takeoverKeysFromHeader":{レスポンスヘッダの値から監視、強制終了リクエストに引継ぐキー},
    "jobnetValKeys": {ジョブネット変数に設定するキー},
    "tcpKeepAlive": {
      "idleTime": TCP Keep-Aiveパケットを送信するまでの時間,
      "probeInterval": TCP Keep-Aliveパケットを再送するまでの時間,
      "probeCount": TCP Keep-Aliveパケットの最大送信回数
    }
  },
  "monitoring": {
    "url": "接続URL",
    "method": "HTTPメソッド",
    "query": {クエリパラメタ},
    "header": {HTTPリクエストヘッダ},
    "body": "HTTPリクエストメッセージボディ",
    "resHeader": "HTTPレスポンスヘッダ",
    "resBody": "HTTPレスポンスボディ",
    "pollingInterval": 監視のポーリング間隔,
    "pollingCount": 監視のポーリング回数,
    "firstMonitorInterval": 初回の監視リクエスト実行までの間隔,
    "pollingEndKey": "監視の終了判定キー",
    "pollingEndVal": [監視の終了判定の値],
    "checkResultKey": "正常終了判定キー",
    "checkResultVal": [正常終了判定の値],
    "jobnetValKeys": {ジョブネット変数に設定するキー}
  },
  "termination": {
    "url": "接続URL",
    "method": "HTTPメソッド",
    "query": {クエリパラメタ},
    "header": {HTTPリクエストヘッダ},
    "body": "HTTPリクエストメッセージボディ",
    "resHeader": "HTTPレスポンスヘッダ",
    "resBody": "HTTPレスポンスボディ"
  }
}
```

パラメタ

requestオブジェクトのパラメタ

requestオブジェクトの指定は必須です。

パラメタ	キー名・型・デフォルト値		説明
接続URL (*)	**キー名**	url	接続先のURLを以下の形式で指定します。 ・ httpsを使用する場合 https://ホスト[:ポート番号]/リソースパス ・ httpを使用する場合
	型	string	
	デフォルト値	なし	

パラメタ	キー名・型・デフォルト値		説明
			http://ホスト[:ポート番号]/リソースパス URLは2083バイト以内で指定します。URL内で利用可能な文字については接続先のクラウドサービスの規約に従ってください。
HTTPメソッド(*)	**キー名**	method	HTTPメソッドを設定します。 GET/PUT/POST/DELETEのいずれかを指定します。
	型	string	
	デフォルト値	なし	
認証情報名	**キー名**	authName	jobschsetauthinfoコマンドで設定したリクエストの認証情報名を指定します。コマンドの詳細については、“4.9 jobschsetauthinfoコマンド【Windows版】【Linux版】”を参照してください。 省略した場合、認証処理を行いません。
	型	string	
	デフォルト値	なし	
プロキシサーバの認証情報名	**キー名**	proxyAuthName	jobschsetauthinfoコマンドで設定したプロキシサーバの認証情報名を指定します。コマンドの詳細については、“4.9 jobschsetauthinfoコマンド【Windows版】【Linux版】”を参照してください。 省略した場合、プロキシサーバを利用しません。
	型	string	
	デフォルト値	なし	

クエリパラメタ	**キー名**	query	接続先URLにクエリパラメタを指定する場合、“パラメタ名:値”の形式で指定します。以下のようにパラメタ名と値はqueryオブジェクト内で複数指定可能です。値はstring型で指定してください。
			`query: {`
			`  パラメタ名1:値1,`
			`  パラメタ名2:値2,`
			`  …`
			`}`
			クエリパラメタの値にjobschsetauthinfoコマンドで設定した任意のキーの値を設定したい場合は、以下のように値を“@任意のキー名@”の形式で指定することで、設定した値に置きかわります。
			`query: {`
			`  パラメタ名:@任意のキー名@`
			`}`
			パラメタ名および値は1024バイト以内で指定してください。
			省略した場合、URLにクエリパラメタを追加しません。

	型	object	
	デフォルト値	なし	
HTTPリクエストヘッダ	**キー名**	header	追加のHTTPリクエストヘッダを指定する場合、“ヘッダ名:値”の形式で指定します。以下のようにヘッダ名と値はheaderオブ
	型	object	

パラメタ	キー名・型・デフォルト値		説明

	デフォルト値	なし	ジェクト内で複数指定可能です。値はstring型で指定してください。
			
			header: {
			ヘッダ名1:値1,
			ヘッダ名2:値2,
			…
			}
			
			リクエストヘッダの値にjobschsetauthinfoコマンドで設定した任意のキーの値を設定したい場合は、以下のように値を“@任意のキー名@”の形式で指定することで、設定した値に置きかかわります。
			
			header: {
			ヘッダ名:@任意のキー名@
			}
			
			ヘッダ名および値は1024バイト以内で指定してください。
			
			省略した場合、リクエストヘッダを追加しません。

HTTPリクエストメッセージボディ	**キー名**	body	HTTPリクエストメッセージボディとして指定するファイルをフルパスで指定します。 ファイルパスの長さは256バイト以内で指定してください。ファイルパスには、半角英数字、“/”、“:”、“¥”、“.”、“_”(半角アンダーバー)、“-”(半角ハイフン)のみ指定可能です。 jobschsetauthinfoコマンドで設定した任意のキーの値をリクエストメッセージボディに設定する場合、“@任意のキー名@”の形式で指定することで、リクエスト時に設定した値に置き換わります。 省略した場合、リクエストメッセージボディは設定されません。
	型	string	
	デフォルト値	なし	
HTTPレスポンスヘッダ	**キー名**	resHeader	HTTPレスポンスのヘッダ情報を保存するファイルをフルパスで指定します。 ファイルパスの長さは256バイト以内で指定してください。ファイルパスには、半角英数字、“/”、“:”、“¥”、“.”、“_”(半角アンダーバー)、“-”(半角ハイフン)のみ指定可能です。 指定ファイルが存在していた場合は上書きします。 本パラメタの指定を変更せず一意のファイル名で出力したい場合は、ファイル名に<date>の文字列を含めることで、出力時に<date>がYYYYMMDDhhmmssの日時情報に置換されて出力されます。この場合、置換後のファイルパス長が256バイトを超
	型	string	
	デフォルト値	なし	

パラメタ	キー名・型・デフォルト値		説明
			えないようにしてください。超えた場合、結果は出力されません。 省略した場合、ヘッダ情報のファイル出力は行いません。
HTTPレスポンスボディ	**キー名**	resBody	HTTPレスポンスのボディ情報を保存するファイルをフルパスで指定します。 ファイルパスの長さは256バイト以内で指定してください。ファイルパスには、半角英数字、“/”、“:”、“¥”、“.”、“_”(半角アンダーバー)、“-”(半角ハイフン)のみ指定可能です。 指定ファイルが存在していた場合は上書きします。 本パラメタの指定を変更せず一意のファイル名で出力したい場合は、ファイル名に<date>の文字列を含めることで、出力時に<date>がYYYYMMDDhhmmssの日時情報に置換されて出力されます。この場合、置換後のファイルパス長が256バイトを超えないようにしてください。超えた場合、結果は出力されません。 省略した場合、ボディ情報のファイル出力は行いません。
	型	string	
	デフォルト値	なし	
接続タイムアウト時間	**キー名**	timeout	HTTPリクエストの接続タイムアウト時間を 1 ～1800(秒)で指定します。 省略した場合デフォルトの10(秒)が設定されます。
	型	number	
	デフォルト値	10	

レスポンスボディの値から監視、強制終了リクエストに引継ぐキー	**キー名**	takeoverKeys	monitoringオブジェクトで指定する監視リクエストや、terminationオブジェクトで指定する強制終了リクエストのURL、リクエストメッセージボディ、およびクエリパラメタに本HTTPリクエストの結果を引継ぎたい場合、引き継ぐレスポンス結果のキーを指定することでそのキーの値を引き継ぐことができます。
			監視リクエストの場合、監視の終了判定キーおよび正常終了判定キーにも引き継ぐことができます。
			引継ぎ先のURL、リクエストメッセージボディ、およびクエリパラメタには、本パラメタで指定した置き換え変数名を“<置き換え変数名>”の形式で記載しておくことで、抽出したキーの値に変換されます。
			本パラメタは“置き換え変数名:抽出するキー”の形式で指定します。以下のように置き換え変数名と抽出するキーは、takeoverKeysオブジェクト内で複数指定可能です。
			`takeoverKeys: {`
			`  置き換え変数名1:抽出するキー1(※),`

	型	object	
	デフォルト値	なし	

パラメタ	キー名・型・デフォルト値		説明
			置き換え変数名2:抽出するキー2, … } 置き換え変数名は、64バイト以内の半角英数字で指定してください。 (※)抽出するキーの指定方法の詳細は、“抽出するキーの指定方法”を参照してください。

レスポンスヘッダの値から監視、強制終了リクエストに引継ぐキー	**キー名**	takeoverKeysFromHeader	monitoringオブジェクトで指定する監視リクエストや、terminationオブジェクトで指定する強制終了リクエストのURL、リクエストメッセージボディ、およびクエリパラメタに本HTTPリクエストの結果を引継ぎたい場合、引き継ぐレスポンス結果のヘッダ名を指定することで、そのヘッダ名の値を引き継ぐことができます。なお、同じ名前のレスポンスヘッダが複数ある場合は、最初に取得されるヘッダ名の値を引き継ぎます。
			
			監視リクエストの場合、監視の終了判定キーおよび正常終了判定キーにも引き継ぐことができます。
			
			引継ぎ先のURL、リクエストメッセージボディ、およびクエリパラメタには、本パラメタで指定した置き換え変数名を“<置き換え変数名>”の形式で記載しておくことで、抽出したヘッダ名の値に変換されます
			
			本パラメタは“置き換え変数名:抽出するヘッダ名”の形式で指定します。以下のように置き換え変数名と抽出するキーは、takeoverKeysfromHeaderオブジェクト内で複数指定可能です。
			
			takeoverKeysFromHeader: {
			置き換え変数名1:抽出するヘッダ名1,
			置き換え変数名2:抽出するヘッダ名2,
			…
			}
			
			置き換え変数名は、64バイト以内の半角英数字で指定してください。また、抽出するヘッダ名は、string型で1024バイト以内で指定してください。

	型	object	
	デフォルト値	なし	
ジョブネット変数に設定するキー	**キー名**	jobnetValKeys	本HTTPレスポンスの結果をジョブネット変数に設定して後続ジョブと連携したい場合、本パラメタでレスポンスのキーを指定することでそのキーの値をジョブネット変数に設定することができます。 本パラメタで指定したキーは“ジョブネット変数名=キーの値”の形式で標準出力に出力されます。 リスト形式で複数のキーを指定することが可能です。
	型	object	
	デフォルト値	なし	

パラメタ	キー名・型・デフォルト値		説明

			本パラメタは“ジョブネット変数名:抽出するキー”の形式で指定します。以下のようにジョブネット変数名と抽出するキーはjobnetValKeysオブジェクト内で複数指定可能です。
			
			jobnetValKeys: {
			ジョブネット変数名1:抽出するキー1(※),
			ジョブネット変数名2:抽出するキー2,
			…
			}
			
			(※)抽出するキーの指定方法の詳細は、“抽出するキーの指定方法”を参照してください。
			
			ジョブの設定でジョブネット変数を利用する設定になっていない場合、本パラメタを指定していても、ジョブネット変数には設定されません。ジョブネット変数を利用する手順およびジョブネット変数で定義可能な文字列、制限値については“Systemwalker Operation Manager 運用ガイド”の“ジョブネット変数をジョブ間で利用する”を参照してください。
			
			ジョブネット変数名はREST実行ジョブの出力結果で出力される他のキー名と重複しないようにしてください。重複した場合、パラメタエラーとなります。

TCP Keep-Aliveの有効化	**キー名**	tcpKeepAlive	REST実行ジョブで実行されるすべてのHTTP通信でTCP Keep-Aliveを有効にします。 tcpKeepAliveオブジェクト内には以下のキー名のみ指定可能です。 ・ idelTime ・ probeInterval ・ probeCount TCP Keep-Aliveの各パラメタの詳細は、“request.tcpKeepAliveオブジェクトのパラメタ”を参照してください。 本パラメタを省略した場合、TCP Keep-Aliveは無効になります。
	型	object	
	デフォルト値	なし	

(*)：必須のパラメタ

request.tcpKeepAliveオブジェクトのパラメタ

TCP Keep-Aliveを有効化することで、REST実行ジョブがHTTPリクエストを送信してから連携先のクラウドサービスからHTTPレスポンスを受信するまでの間、REST実行ジョブからクラウドサービスに対して、定期的にTCP Keep-Aliveパケットを送信させることができます。AWS Network FirewallやAzure Firewallなどのサービスを利用してTCP/IP通信のアイドル時間のタイムアウトを設けている場合、タイムアウト時間内にTCP Keep-Aliveパケットを送信することで、TCP/IP通信を継続させることが可能です。上記のタイムアウト時間を

超える時間のジョブを実行する場合、この設定が無効であるとタイムアウトによる通信遮断でジョブが異常終了するため、この設定を有効化することを推奨します。

requestオブジェクトでtcpKeepAliveを指定した場合、tcpKeepAliveオブジェクト内で下記TCP Keep-Aliveのパラメタが指定可能となります。

パラメタ	キー名・型・デフォルト値		説明
TCP Keep-Aliveパケットを送信するまでの時間	**キー名**	idleTime	クラウドサービスに対してTCP/IP通信のパケットを送信してから、TCP Keep-Aliveパケットを送信するまでの時間を、30～7200(秒)で設定します。 この設定により、AWS Network FirewallやAzure Firewallなどのサービスによるアイドル時間のタイムアウト時間を、idleTimeの時間分延長させることができ、タイムアウトによる通信遮断を回避することが可能です。 省略した場合、デフォルトの60(秒)が設定されます。
	型	number	
	デフォルト値	60	
TCP Keep-Aliveパケットを再送するまでの時間	**キー名**	probeInterval	REST実行ジョブが送信したTCP Keep-Aliveパケットに対して、連携先のクラウドサービスから応答がなかった場合に、再度リトライ通信としてTCP Keep-Aliveパケットを再送するまでの時間を、10～75(秒)で設定します。 省略した場合、デフォルトの10(秒)が設定されます。
	型	number	
	デフォルト値	10	
TCP Keep-Aliveパケットの最大送信回数	**キー名**	probeCount	リトライ通信として送信するTCP Keep-Aliveパケットの最大送信回数を、1～9(回)で設定します。 TCP Keep-Aliveパケットの最大送信回数を超えて応答がなかった場合、ジョブは異常終了します。 省略した場合、デフォルトの9(回)が設定されます。
	型	number	
	デフォルト値	9	

monitoringオブジェクトのパラメタ

requestオブジェクトで指定した接続URLのAPIの処理の完了を待ち合わせる場合に、monitoringオブジェクトで実行APIの処理結果取得用APIのパラメタを指定します。

本オブジェクトで指定した接続先URLに指定のポーリング間隔、ポーリング回数でHTTPリクエストを送信します。

指定した終了判定キーの値がレスポンス結果の値と一致するか、指定したポーリング回数分のリクエストを送信するとREST実行ジョブは終了します。

monitoringオブジェクトで処理の完了を待ち合わせる場合、正常終了判定キーで指定した値とレスポンス結果の値が一致したときにREST実行ジョブは正常終了となります。それ以外の場合、REST実行ジョブは異常終了となります。

送信されるすべてのリクエストに対してパラメタは共通です。

レスポンス結果は最後のリクエスト分のみ出力されます。

monitoringオブジェクトは省略可能です。省略した場合、APIの完了の待ち合わせは行いません。

パラメタ	キー名・型・デフォルト値		説明
接続URL (*)	**キー名**	url	接続先のURLを以下の形式で指定します。 ・ httpsを使用する場合
	型	string	

パラメタ	キー名・型・デフォルト値		説明
	デフォルト値	なし	https://ホスト[:ポート番号]/リソースパス ・ httpを使用する場合 http://ホスト[:ポート番号]/リソースパス URLは2083バイト以内で指定します。URL内で利用可能な文字については接続先のクラウドサービスの規約に従ってください。 requestオブジェクトで“レスポンスボディの値から監視、強制終了リクエストに引継ぐキー”または“レスポンスヘッダの値から監視、強制終了リクエストに引継ぐキー”のパラメタで指定したキーの値を本URLに設定する場合、“<置き換え変数名>”の形式で記載することで、リクエスト時に置き換わります。
HTTPメソッド(*)	**キー名**	method	HTTPメソッドを設定します。 GET/PUT/POST/DELETEのいずれかを指定します。
	型	string	
	デフォルト値	なし	

クエリパラメタ	キー名	query	接続先URLにクエリパラメタを指定する場合、“パラメタ名:値”の形式で指定します。以下のようにパラメタ名と値はqueryオブジェクト内で複数指定可能です。値は string型で指定してください。
			`query: {`
			`  パラメタ名1:値1,`
			`  パラメタ名2:値2,`
			`  …`
			`}`
			クエリパラメタの値にjobschsetauthinfoコマンドで設定した任意のキーの値を設定したい場合は、以下のように値を“@任意のキー名@”の形式で指定することで、設定した値に置きかわります。
			`query: {`
			`  パラメタ名:@任意のキー名@`
			`}`
			requestオブジェクトで“レスポンスボディの値から監視、強制終了リクエストに引継ぐキー”または“レスポンスヘッダの値から監視、強制終了リクエストに引継ぐキー”のパラメタで指定したキーの値を本クエリパラメタに設定する場合、“<置き換え変数名>”の形式で記載することで、リクエスト時に置き換わります。
			パラメタ名および値は1024バイト以内で指定してください。
			省略した場合、URLにクエリパラメタを追加しません。

	型	object	
	デフォルト値	なし	
HTTPリクエストヘッダ	**キー名**	header	追加のHTTPリクエストヘッダを指定する場合、“ヘッダ名:値”の形式で指定します。以下のようにヘッダ名と値はheaderオブジェクト内で複
	型	object	

パラメタ	キー名・型・デフォルト値		説明

	デフォルト値	なし	数指定可能です。値は string型で指定してください。
			
			header: {
			　ヘッダ名1:値1,
			　ヘッダ名2:値2,
			　…
			}
			
			リクエストヘッダの値にjobschsetauthinfoコマンドで設定した任意のキーの値を設定したい場合は、以下のように値を“@任意のキー名@”の形式で指定することで、設定した値に置きかわります。
			
			header: {
			　ヘッダ名:@任意のキー名@
			}
			
			ヘッダ名および値は1024バイト以内で指定してください。
			
			省略した場合、リクエストヘッダを追加しません。

HTTPリクエストメッセージボディ	**キー名**	body	HTTPリクエストメッセージボディとして指定するファイルをフルパスで指定します。 ファイルパスの長さは256バイト以内で指定してください。ファイルパスには、半角英数字、“/”、“:”、“¥”、“.”、“_”(半角アンダーバー)、“-”(半角ハイフン)のみ指定可能です。 requestオブジェクトで“レスポンスボディの値から監視、強制終了リクエストに引継ぐキー”または“レスポンスヘッダの値から監視、強制終了リクエストに引継ぐキー”のパラメタで指定したキーの値をリクエストメッセージボディに設定する場合、“<置き換え変数名>”の形式で記載することで、リクエスト時に置き換わります。 jobschsetauthinfoコマンドで設定した任意のキーの値をリクエストメッセージボディに設定する場合、“@任意のキー名@”の形式で指定することで、リクエスト時に設定した値に置き換わります。 省略した場合、リクエストメッセージボディは設定されません。
	型	string	
	デフォルト値	なし	
HTTPレスポンスヘッダ	**キー名**	resHeader	HTTPレスポンスのヘッダ情報を保存するファイルをフルパスで指定します。 ファイルパスの長さは256バイト以内で指定してください。ファイルパスには、半角英数字、“/”、“:”、“¥”、“.”、“_”(半角アンダーバー)、“-”(半角ハイフン)のみ指定可能です。 指定ファイルが存在していた場合は上書きします。 本パラメタの指定を変更せず一意のファイル名で出力したい場合は、ファイル名に<date>の文
	型	string	
	デフォルト値	なし	

パラメタ	キー名・型・デフォルト値		説明
			字列を含めることで、出力時に<date>がYYYYMMDDhhmmssの日時情報に置換されて出力されます。この場合、置換後のファイルパス長が256バイトを超えないようにしてください。超えた場合、結果は出力されません。 監視の最後のHTTPリクエストに対するHTTPレスポンス結果のみ出力します。 省略した場合、ヘッダ情報のファイル出力は行いません。
HTTPレスポンスボディ	**キー名**	resBody	HTTPレスポンスのボディ情報を保存するファイルをフルパスで指定します。 ファイルパスの長さは256バイト以内で指定してください。ファイルパスには、半角英数字、“/”、“:”、“¥”、“.”、“_”(半角アンダーバー)、“-”(半角ハイフン)のみ指定可能です。 指定ファイルが存在していた場合は上書きします。 本パラメタの指定を変更せず一意のファイル名で出力したい場合は、ファイル名に<date>の文字列を含めることで、出力時に<date>がYYYYMMDDhhmmssの日時情報に置換されて出力されます。この場合、置換後のファイルパス長が256バイトを超えないようにしてください。超えた場合、結果は出力されません。 監視の最後のHTTPリクエストに対するHTTPレスポンス結果のみ出力します。 省略した場合、ボディ情報のファイル出力は行いません。
	型	string	
	デフォルト値	なし	
監視のポーリング間隔	**キー名**	pollingInterval	リクエストのポーリング間隔を 1 ～600(秒)で指定します。 省略した場合、デフォルトの60(秒)が設定されます。
	型	number	
	デフォルト値	60	
監視のポーリング回数	**キー名**	pollingCount	リクエストのポーリング回数を指定します。 0が指定された場合、無制限に待ち合わせます。 省略した場合、デフォルトの0が設定されます。
	型	number	
	デフォルト値	0	
初回の監視リクエスト実行までの間隔	**キー名**	firstMonitorInterval	requestオブジェクトで指定したリクエスト実行から初回の監視リクエストを実行するまでの間隔を1～600(秒)で指定します。 省略した場合、デフォルトの10(秒)が設定されます。
	型	number	
	デフォルト値	10	
監視の終了判定キー (*)	**キー名**	pollingEndKey	実行APIが完了したことを判定するためのレスポンス結果のキーを指定します。 キーの指定方法の詳細は、“抽出するキーの指定方法”を参照してください。 requestオブジェクトの“監視、強制終了リクエストに引継ぐキー”パラメタで指定したキー
	型	string	
	デフォルト値	なし	

パラメタ	キー名・型・デフォルト値		説明
			の値を監視の終了判定キーに設定する場合、“<置き換え変数名>”の形式で記載することで、リクエスト時に置き換わります。
監視の終了判定の値 (*)	**キー名**	pollingEndVal	実行APIが完了したことを判定するためのレスポンス結果のキーの値を指定します。 終了判定するための値をリスト形式で指定します。 [”値1”,”値2”,”値3”, …] 値は1024バイト以内で指定してください。
	型	array	
	デフォルト値	なし	
正常終了判定キー (*)	**キー名**	checkResultKey	実行APIが終了した後に、APIが正常終了したかどうかを判定するためのレスポンス結果のキーを指定します。 ジョブの待ち合わせが完了した時に正常終了判定キーで指定した値と合致しなければ、REST実行ジョブは異常終了となります。 キーの指定方法の詳細は、“抽出するキーの指定方法”を参照してください。 requestオブジェクトの“監視、強制終了リクエストに引継ぐキー”パラメタで指定したキーの値を監視の終了判定キーに設定する場合、“<置き換え変数名>”の形式で記載することで、リクエスト時に置き換わります。
	型	string	
	デフォルト値	なし	
正常終了判定の値	**キー名**	checkResultVal	実行APIが終了した後に、APIが正常終了したかどうかを判定するためのレスポンス結果のキーの値を指定します。 正常終了判定するための値をリスト形式で指定します。 [”値1”,”値2”,”値3”, …] 値は1024バイト以内で指定してください。
	型	array	
	デフォルト値	なし	

ジョブネット変数に設定するキー	**キー名**	jobnetValKeys	本HTTPレスポンスの結果をジョブネット変数に設定して後続ジョブと連携したい場合に、本パラメタでレスポンスのキーを指定することで、そのキーの値をジョブネット変数に設定できます。
			本パラメタで指定したキーは“ジョブネット変数名=キー”の値の形式で標準出力に出力されます。
			本パラメタはジョブネット変数名:抽出するキーの形式で指定します。以下のようにジョブネット変数名と抽出するキーはjobnetValKeysオブジェクト内で複数指定可能です。
			`jobnetValKeys: {`
			`  ジョブネット変数名1:抽出するキー1(※),`
			`  ジョブネット変数名2:抽出するキー2,`
			`  …`
			`}`
			(※)抽出するキーの指定方法の詳細は、“抽出するキーの指定方法”を参照してください。

	型	object	
	デフォルト値	なし	

パラメタ	キー名・型・デフォルト値		説明
			ジョブの設定でジョブネット変数を利用する設定になっていない場合、本パラメタを指定していても、ジョブネット変数には設定されません。ジョブネット変数を利用する手順およびジョブネット変数で定義可能な文字列、制限値については“Systemwalker Operation Manager 運用ガイド”の“ジョブネット変数をジョブ間で利用する”を参照してください。 ジョブネット変数名はREST実行ジョブの出力結果で出力される他のキー名と重複しないようにしてください。重複した場合、パラメタエラーとなります。

(*)：必須のパラメタ

terminationオブジェクトのパラメタ

REST実行ジョブの強制終了時に、強制終了リクエストをrequestオブジェクトで指定した接続URLのAPIの処理に送りたい場合は、terminationオブジェクトで以下のパラメタを指定します。

terminationオブジェクトは省略可能です。省略した場合、APIの強制終了リクエストは送りません。

パラメタ	キー名・型・デフォルト値		説明
接続URL (*)	**キー名**	url	接続先のURLを以下の形式で指定します。 ・ httpsを使用する場合 https://ホスト[:ポート番号]/リソースパス ・ httpを使用する場合 http://ホスト[:ポート番号]/リソースパス URLは2083バイト以内で指定します。URL内で利用可能な文字については接続先のクラウドサービスの規約に従ってください。 requestオブジェクトで“レスポンスボディの値から監視、強制終了リクエストに引継ぐキー”または“レスポンスヘッダの値から監視、強制終了リクエストに引継ぐキー”のパラメタで指定したキーの値を本URLに設定する場合、“<置き換え変数名>”の形式で記載することで、リクエスト時に置き換わります。
	型	string	
	デフォルト値	なし	
HTTPメソッド (*)	**キー名**	method	HTTPメソッドを設定します。 GET/PUT/POST/DELETEのいずれかを指定します。
	型	string	
	デフォルト値	なし	

クエリパラメタ	キー名	query	接続先URLにクエリパラメタを指定する場合、“パラメタ名:値”の形式で指定します。以下のようにパラメタ名と値はqueryオブジェクト内で複数指定可能です。値は string型で指定してください。
			`query: {`
			`  パラメタ名1:値1,`
			`  パラメタ名2:値2,`
			`  …`
			`}`

	型	object	
	デフォルト値	なし	

パラメタ	キー名・型・デフォルト値		説明

			クエリパラメタの値にjobschsetauthinfoコマンドで設定した任意のキーの値を設定したい場合は、以下のように値を“@任意のキー名@”の形式で指定することで、設定した値に置きかわります。
			
			query: {
			  パラメタ名:@任意のキー名@
			}
			
			requestオブジェクトで“レスポンスボディの値から監視、強制終了リクエストに引継ぐキー”または“レスポンスヘッダの値から監視、強制終了リクエストに引継ぐキー”のパラメタで指定したキーの値を本クエリパラメタに設定する場合、“<置き換え変数名>”の形式で記載することで、リクエスト時に置き換わります。
			
			パラメタ名および値は1024バイト以内で指定してください。
			
			省略した場合、URLにクエリパラメタを追加しません。

HTTPリクエストヘッダ	**キー名**	header	追加のHTTPリクエストヘッダを指定する場合、“ヘッダ名:値”の形式で指定します。以下のようにヘッダ名と値はheaderオブジェクト内で複数指定可能です。値は string型で指定してください。
			
			header: {
			  ヘッダ名1:値1,
			  ヘッダ名2:値2,
			  …
			}
			
			リクエストヘッダの値にjobschsetauthinfoコマンドで設定した任意のキーの値を設定したい場合は、以下のように値を“@任意のキー名@”の形式で指定することで、設定した値に置きかわります。
			
			header: {
			  ヘッダ名:@任意のキー名@
			}
			
			ヘッダ名および値は1024バイト以内で指定してください。
			
			省略した場合、リクエストヘッダを追加しません。

	型	object	
	デフォルト値	なし	
HTTPリクエストメッセージボディ	**キー名**	body	HTTPリクエストメッセージボディとして指定するファイルをフルパスで指定します。 ファイルパスの長さは256バイト以内で指定してください。ファイルパスには、半角英数字、“/”、“:”、“¥”、“.”、“_”(半角アンダーバー)、“-”(半角ハイフン)のみ指定可能です。 requestオブジェクトで“レスポンスボディの値から監視、強制終了リクエストに引継ぐキー”または“レスポンスヘッダの値から監視、
	型	string	
	デフォルト値	なし	

パラメタ	キー名・型・デフォルト値		説明
			強制終了リクエストに引継ぐキー”のパラメタで指定したキーの値をリクエストメッセージボディに設定する場合、“<置き換え変数名>”の形式で記載することで、リクエスト時に置き換わります。 jobschsetauthinfoコマンドで設定した任意のキーの値をリクエストメッセージボディに設定する場合、“@任意のキー名@”の形式で指定することで、リクエスト時に設定した値に置き換わります。 省略した場合、リクエストメッセージボディは設定されません。
HTTPレスポンスヘッダ	**キー名**	resHeader	HTTPレスポンスのヘッダ情報を保存するファイルをフルパスで指定します。 ファイルパスの長さは256バイト以内で指定してください。ファイルパスには、半角英数字、“/”、“:”、“¥”、“.”、“_”(半角アンダーバー)、“-”(半角ハイフン)のみ指定可能です。 指定ファイルが存在していた場合は上書きします。 本パラメタの指定を変更せず一意のファイル名で出力したい場合は、ファイル名に<date>の文字列を含めることで、出力時に<date>がYYYYMMDDhhmmssの日時情報に置換されて出力されます。この場合、置換後のファイルパス長が256バイトを超えないようにしてください。超えた場合、結果は出力されません。 省略した場合、ヘッダ情報のファイル出力は行いません。
	型	string	
	デフォルト値	なし	
HTTPレスポンスボディ	**キー名**	resBody	HTTPレスポンスのボディ情報を保存するファイルをフルパスで指定します。 ファイルパスの長さは256バイト以内で指定してください。ファイルパスには、半角英数字、“/”、“:”、“¥”、“.”、“_”(半角アンダーバー)、“-”(半角ハイフン)のみ指定可能です。 指定ファイルが存在していた場合は上書きします。 本パラメタの指定を変更せず一意のファイル名で出力したい場合は、ファイル名に<date>の文字列を含めることで、出力時に<date>がYYYYMMDDhhmmssの日時情報に置換されて出力されます。この場合、置換後のファイルパス長が256バイトを超えないようにしてください。超えた場合、結果は出力されません。 省略した場合、ボディ情報のファイル出力は行いません。
	型	string	
	デフォルト値	なし	

(*)：必須のパラメタ

抽出するキーの指定方法

takeoverKeys、jobnetValKeys、pollingEndKey、checkResultKeyで、レスポンス結果から抽出するキーを指定する方法について以下に示します。

なお、抽出するキーは、string型で1024バイト以内で指定してください。

- 第1階層のキーの場合

  キー名をそのまま指定します。

  キー名に“.”が含まれている場合は指定できません。

- 第N階層のキーの場合

  階層ごとに“.”を入れて指定します。

  キー名に“.”が含まれている場合は指定できません。

第1階層のキー名.第2階層のキー名.…. 第N階層のキー名

- リスト内のキーの場合

  抽出するキーを含むオブジェクトを以下のいずれかの方法で指定します。

  ー 他のキーの値を指定

[?"他のキー名"==他のキー名の値].抽出するキー名

  - 他のキー名、他のキー名の値、抽出するキー名に、“.”または“=”が含まれている場合は指定できません。
  - 他のキー名の値は、string型とnumber型が指定可能です。string型を指定する場合は、他のキー名の値をダブルクォーテーションで囲んでください。
  - 指定した他のキー名がリスト内に複数ある場合、リストの先頭から最初に条件に一致したものを抽出します。

  ー インデックスを指定

[インデックス番号].抽出するキー名

  - リストの先頭のオブジェクトのインデックス番号は0になります。
  - キー名に“.”が含まれている場合は指定できません。

なお、レスポンス結果がXML形式の場合は、JSON形式に読み替えて指定してください。XML形式からJSON形式への読み替えは、以降に記載の「レスポンス結果の例（XML形式の場合）」を参考にしてください。

レスポンス結果の例（JSON形式の場合）

```
{
    ”id”: ”id001”,
    ”info”: {
        ”testKey”: ”testValue”
     },
    ”properties”: [
            {
                ”name”: ”aaa”,
                ”rc”: 2
            },
            {
                ”name”: ”bbb”,
                ”rc”: 0
            }
```

```
    ]
}
```

- 抽出するキーが"id"の場合： "id"
- 抽出するキーが"info"配下の"testKey"の場合: "info.testKey"
- 抽出するキーが"properties"配下の"name"キーの値が"bbb"(string型)のオブジェクトの"rc"の場合:
  - "properties.[?¥"name¥"==¥"bbb¥"].rc"
  - "properties.[1].rc"
- 抽出するキーが"properties"配下の"rc"キーの値が"0"(number型)のオブジェクトの"name"の場合:
  - "properties.[?¥"rc¥"==0].name"
  - "properties.[1].name"

レスポンス結果の例（XML形式の場合）

```
<?xml version="1.0" encoding="UTF-8" ?>
 <root>
     <id>id002</id>
     <info>
         <testTag>test1</testTag>
     </info>
     <properties>
         <name>ccc</name>
         <rc>2</rc>
     </properties>
     <properties>
         <name>ddd</name>
         <rc>1</rc>
     </properties>
</root>
```

抽出するキーの指定方法は、XML形式のレスポンス結果を以下のJSON形式に読み替えた上で、レスポンス結果がJSON形式の場合の指定方法と同様に指定してください。

```
{
  "root": {
      "id": "id002",
      "info": {
          "testTag": "test1"
       },
      "properties": [
              {
                  "name": "ccc",
                  "rc": 2
              },
              {
                  "name": "ddd",
                  "rc": 1
              }
      ]
  }
}
```

ジョブ定義変数の利用について

下記のパラメタの値には、ジョブ定義変数を利用することができます。パラメタの値(全体または一部)を、“@ジョブ定義変数名@”の形式で指定することで、登録済のジョブ定義変数で置き換えることができます。

- 接続URLの値に含まれる先頭の"https://"以外の文字
- クエリパラメタに指定したパラメタの値に含まれる文字
- HTTPリクエストヘッダに指定したヘッダの値に含まれる文字
- HTTPリクエストメッセージボディの値に含まれる文字

なお、値に“@任意のキー名@”の指定が可能なパラメタの場合、「任意のキー名」と「ジョブ定義変数名」が同一であった場合は、任意のキー名が優先され置き換えられます。任意のキー名で置き換えられた値に“@ジョブ定義変数名@”の指定がある場合、その部分は登録済のジョブ定義変数で置き換えられます。

また、ジョブ定義変数の置き換え後の値に“<置き換え変数名>”の指定がある場合は、その部分は抽出したキーの値で置き換えられます。

ジョブ定義変数の詳細については、“Systemwalker Operation Manager 運用ガイド”の“ジョブ定義変数を利用してジョブ登録情報を一括置換する場合”を参照してください。

ジョブ定義変数の指定例

ジョブ定義変数“JOBDEF”で置き換える場合、以下のように指定します。

- 接続URL

```
”url”: ”https://@JOBDEF@.azurewebsites.net/api/HttpTrigger1”
```

- クエリパラメタ

```
”query”: {
      ”パラメタ名”: ”param@JOBDEF@”
}
```

- HTTPリクエストヘッダ

```
”header”: {
     ”ヘッダ名”: ”@JOBDEF@”
}
```

- HTTPリクエストメッセージボディ

```
{
     ”パラメタ1”: ”@JOBDEF@”
}
```

変数の置き換えができない指定方法について

パラメタの値に変数を利用する場合、指定方法によっては置き換えができません。置き換わる前の文字列がパラメタに残ることで、ジョブのエラーにつながる可能性があり注意が必要です。

変数の置き換えができない指定方法について以下に示します。

- ジョブ定義変数が未定義の場合
- 任意のキーが未定義の場合
- 置き換え変数が未定義の場合
- 変数の置き換え後の値に、@ジョブ定義変数名@、@任意のキー名@、および、<置き換え変数名>がある場合
  変数の置き換えは、REST実行ジョブの実行時に以下の順に実施されます。
  変数の値に、すでに置き換え済みとなる変数の値が含まれていても、その変数は置き換わりません。
  1. “@任意のキー名@”
  2. “@ジョブ定義変数名@”

3. takeoverKeysで指定された“<置き換え変数名>”
4. takeoverKeysFromHeaderで指定された“<置き換え変数名>”

注意事項

- monitoringオブジェクトとterminationオブジェクトの以下のパラメタは、requestオブジェクトと共通です。
  - 認証情報名
  - プロキシサーバの認証情報名
  - 接続タイムアウト時間
- requestオブジェクトの以下のパラメタで“<置き換え変数名>”を指定する場合、重複して指定することはできません。重複して指定された場合、REST実行ジョブは異常終了となります。
  - レスポンスボディの値から監視、強制終了リクエストに引継ぐキー
  - レスポンスヘッダの値から監視、強制終了リクエストに引継ぐキー
- 接続情報ファイルの記載内容に誤りがある場合、REST実行ジョブは異常終了となります。

使用例

連携可能なクラウドサービスと対象となる操作は、以下のとおりです。

サービス名	操作
AWS Batch	Batch処理を実行します。
AWS Lambda	Lambda関数を実行します。
AWS Step Functions	ワークフローを実行します。
Amazon Simple Storage Service (Amazon S3)	ファイルの待ち合わせを行います。
AWS Backup	リソースのバックアップを実行します。
Amazon Relational Database Service (Amazon RDS)	データベースインスタンスの起動/停止を行います。
Amazon Elastic Compute Cloud (Amazon EC2)	仮想マシンの起動/停止を行います。
Amazon Elastic Kubernetes Service (Amazon EKS)	ワーカーノードの起動/停止を行います。
Azure Batch	Batch処理を実行します。
Azure Functions	Functions関数を実行します。
Azure Logic Apps	ワークフローを実行します。
Azure Blob Storage	ファイルの待ち合わせを行います。
Azure Backup	リソースのバックアップを実行します。
Azure SQL Managed Instance	データベースインスタンスの起動/停止を行います。
Azure Virtual Machines	仮想マシンの起動/停止を行います。
Azure Kubernetes Service (AKS)	クラスタの起動/停止を行います。
Oracle Cloud Infrastructure Functions (OCI Functions)	Functions関数を実行します。
Oracle Cloud Infrastructure Compute (OCI Compute)	仮想マシンの起動/停止を行います。

サービス名	操作
Oracle Cloud Infrastructure Kubernetes Engine (OKE)	ノードの起動/停止を行います。
Oracle Autonomous Database	データベースサービスの起動/停止を行います。
Oracle Cloud Infrastructure Object Storage (OCI Object Storage)	ファイルの待ち合わせを行います。

以降の小項では、各サービスと連携する場合の接続情報ファイルの例を掲載しています。

リソース情報や認証情報、ファイルパスなどは具体的な値を記載しているため(例. “test1.txt” など)、実際のリソース情報や認証情報、ファイルパスなどに合わせて変更してください。

20.9.1.1 AWS Batchサービスと連携する例

AWS Batchサービス(API Version 2016-08-10)のジョブを実行、監視、強制終了する場合の例を示します。

```
{
  ”request”: {
    ”url”: ”https://batch.ap-northeast-1.amazonaws.com/v1/submitjob”,
    ”method”: ”POST”,
    ”authName”: ”Aws_Batch”,
    ”proxyAuthName”: ”Proxy1”,
    ”body”: ”/paramfiles/aws_batch_main_body.json”,
    ”resHeader”: ”/response/aws_batch_main_header_<date>.json”,
    ”resBody”: ”/response/aws_batch_main_body_<date>.txt”,
    ”takeoverKeys”: {
      ”JOBID”: ”jobId”
    },
  ”timeout”: 60
  },
  ”monitoring”: {
    ”url”: ”https://batch.ap-northeast-1.amazonaws.com/v1/describejobs”,
    ”method”: ”POST”,
    ”body”: ”/paramfiles/aws_batch_moni_body.json”,
    ”resHeader”: ”/response/aws_batch_moni_header_<date>.json”,
    ”resBody”: ”/response/aws_batch_moni_body_<date>.txt”,
    ”pollingInterval”: 60,
    ”pollingCount”: 5,
    ”pollingEndKey”: ”jobs.[?¥”jobId¥”==¥”<JOBID>¥”].status”,
    ”pollingEndVal”: [”SUCCEEDED”, ”FAILED”],
    ”checkResultKey”: ”jobs.[?¥”jobId¥”==¥”<JOBID>¥”].status”,
    ”checkResultVal”: [”SUCCEEDED”]
  },
  ”termination”: {
    ”url”: ”https://batch.ap-northeast-1.amazonaws.com/v1/terminatejob”,
    ”method”: ”POST”,
    ”body”: ”/paramfiles/aws_batch_term_body.json”,
    ”resHeader”: ”/response/aws_batch_term_header_<date>.json”,
    ”resBody”: ”/response/aws_batch_term_body_<date>.txt”
  }
}
```

上記のrequestオブジェクトで指定するリクエストメッセージボディ(aws_batch_main_body.json)の例

```
{
  ”jobName”: ”testjob1”,
  ”jobQueue”: ”testjob-queue”,
  ”jobDefinition”: ”testjob-def1”
}
```

上記のmonitoringオブジェクトで指定するリクエストメッセージボディ(aws_batch_moni_body.json)の例

```
{
  "jobs": [ "<JOBID>" ]
}
```

上記のterminationオブジェクトで指定するリクエストメッセージボディ(aws_batch_term_body.json)の例

```
{
  "reason": "Terminating job.",
  "jobId": "<JOBID>"
}
```

20.9.1.2 AWS Lambdaサービスと連携する例

AWS Lambdaサービス(API Version 2015-03-31)の関数を実行する場合の例を示します。

```
{
  "request": {
    "url": "https://lambda.ap-northeast-1.amazonaws.com/2015-03-31/functions/testfunc/invocations",
    "method": "POST",
    "authName": "Aws_Lambda",
    "proxyAuthName": "Proxy1",
    "resHeader": "/response/aws_lambda_main_header_<date>.json",
    "resBody": "/response/aws_lambda_main_body_<date>.json",
    "timeout": 60,
    "tcpKeepAlive": {
      "idleTime": 60,
      "probeInterval": 10,
      "probeCount": 9
    }
  }
}
```

注意

例に従って接続情報を記載してREST実行ジョブを実行した場合、Lambda関数でエラーが発生してもREST実行ジョブは正常終了します。

Lambda関数の結果が正常終了/異常終了であるかの判断を行いたい場合は、Lambda関数と接続情報、ジョブネットに変更を行ってください。以下にそれぞれの設定の例を示します。

- Lambda関数の復帰値にエラーを判別するための情報を付与します。

  以下は、Lambda関数の復帰値にある“statusCode”でエラーを判別する場合の例です。

```
def lambda_handler(event, context):
(途中省略)
    return {
      'statusCode': 400,
      'body': 'job failed'
    }
```

- 接続情報ファイルに、requestオブジェクトのパラメタであるジョブネット変数に設定するキーを定義し、HTTPレスポンスの結果内にある“statusCode”の値をジョブネット変数に設定します。

  ジョブネット変数に設定するキー“jobnetValKeys”を追記した接続情報ファイルの例は、以下のとおりです。

```
{
  "request": {
    "url": "https://lambda.ap-northeast-1.amazonaws.com/2015-03-31/functions/testfunc/invocations",
    "method": "POST",
    "authName": "Aws_Lambda",
```

```
    "proxyAuthName": "Proxy1",
    "resHeader": "/response/aws_lambda_main_header_<date>.json",
    "resBody": "/response/aws_lambda_main_body_<date>.json",
    "timeout": 60,
    "jobnetValKeys": {
        "StatusCode": "statusCode"
    },
    "tcpKeepAlive": {
      "idleTime": 60,
      "probeInterval": 10,
      "probeCount": 9
    }
  }
}
```

- REST実行ジョブの後続にジョブネット変数判定ジョブを配置し、ジョブネット変数の値(StatusCode)から正常・異常を判断します。

20.9.1.3 AWS Step Functionsサービスと連携する例

AWS API Gateway経由でAWS Step Functionsサービス(API Version 2016-11-23)のジョブを実行、監視、強制終了する場合の例を示します。

AWS Step FunctionsのAPIのエンドポイントは外部に公開されていないため、AWS API Gatewayを経由する必要があります。AWS API Gatewayの設定方法は、AWSの公式ドキュメントを参照してください。

```
{
  "request": {
    "url": "https://a1b2c3d4e5.execute-api.ap-northeast-1.amazonaws.com/step-functions-stage/startexecution",
    "method": "POST",
    "authName": "Aws_Step_Functions",
    "proxyAuthName": "Proxy1",
    "takeoverKeys": {
      "executionId": "executionArn"
    },
    "body": "/paramfiles/aws_stepFunctions_main_body.json",
    "resHeader": "/response/aws_stepFunctions_main_header_<date>.json",
    "resBody": "/response/aws_stepFunctions_main_body_<date>.json",
    "timeout": 60
  },
  "monitoring": {
    "url": "https://a1b2c3d4e5.execute-api.ap-northeast-1.amazonaws.com/step-functions-stage/
describeexecution",
    "method": "POST",
    "body": "/paramfiles/aws_stepFunctions_moni_body.json",
    "resHeader": "/response/aws_stepFunctions_moni_header_<date>.json",
    "resBody": "/response/aws_stepFunctions_moni_body_<date>.json",
    "pollingInterval": 60,
    "pollingCount": 5,
    "pollingEndKey": "status",
    "pollingEndVal": ["SUCCEEDED","FAILED","TIMED_OUT","ABORTED"],
    "checkResultKey": "status",
    "checkResultVal": ["SUCCEEDED"]
  },
  "termination": {
    "url": "https://a1b2c3d4e5.execute-api.ap-northeast-1.amazonaws.com/step-functions-stage/stopexecution",
    "method": "POST",
    "body": "/paramfiles/aws_stepFunctions_term_body.json",
    "resHeader": "/response/aws_stepFunctions_term_header_<date>.json",
    "resBody": "/response/aws_stepFunctions_term_body_<date>.json"
   }
}
```

上記のrequestオブジェクトで指定するリクエストメッセージボディ(aws_stepFunctions_main_body.json)の例

```
{
    "input": "{}",
    "name": "Sleep60",
    "stateMachineArn": "arn:aws:states:ap-northeast-1:123456789012:stateMachine:sleep60"
}
```

上記のmonitoringオブジェクトで指定するリクエストメッセージボディ(aws_stepFunctions_moni_body.json)の例

```
{
    "executionArn": "<executionId>"
}
```

上記のterminationオブジェクトで指定するリクエストメッセージボディ(aws_stepFunctions_term_body.json)の例

```
{
    "executionArn": "<executionId>"
}
```

20.9.1.4 Amazon Simple Storage Service (Amazon S3)サービスと連携する例

Amazon Simple Storage Service (Amazon S3)サービスのファイル待ち合わせをするLambda関数(API Version 2015-03-31)を実行、監視する場合の例を示します。

Amazon S3サービスを対象としたファイル待ち合わせでは、作成、削除、更新、サイズ更新を待ち合わせることができます。待ち合わせ方法は、monitoringオブジェクト内で指定します。

Amazon S3サービスを対象とした上記のファイル待ち合わせをする場合は、Lambda関数を介して行います。

注意

AWS Lambdaを介して連携するサービスの使用例を利用する場合

本使用例を利用する際は、セキュリティを考慮し、以下を実施してください。

- 関数をデプロイする際は、適切な認証の設定、およびアクセス制限をしてください。
- 関数で指定する引数のチェックは必要に応じて追加してください。

ファイルが存在しない場合、作成されるまで待つ、または、ファイルが削除されるまで待つ場合の例

```
{
  "request": {
    "url": "https://3abcdef7argst12345xs66kja0nmvss.lambda-url.ap-northeast-1.on.aws/",
    "method": "POST",
    "authName": "Aws_S3_Lambda",
    "proxyAuthName": "Proxy1",
    "body": "/paramfiles/amazon_s3_main_body.json",
    "resHeader": "/response/amazon_s3_main_header_<date>.json",
    "resBody": "/response/amazon_s3_main_body_<date>.json",
    "timeout": 60,
    "tcpKeepAlive": {
       "idleTime": 60,
       "probeInterval": 10,
       "probeCount": 9
    }
  },
```

```
  "monitoring": {
    "url": "https://3abcdef7argst12345xs66kja0nmvss.lambda-url.ap-northeast-1.on.aws/",
    "method": "POST",
    "body": "/paramfiles/amazon_s3_moni_body.json",
    "resHeader": "/response/amazon_s3_moni_header_<date>.json",
    "resBody": "/response/amazon_s3_moni_body_<date>.json",
    "firstMonitorInterval": 3,
    "pollingInterval": 60,
    "pollingCount": 5,
    "pollingEndKey": "result",
    "pollingEndVal": ["SUCCEEDED"],
    "checkResultKey": "result",
    "checkResultVal": ["SUCCEEDED"]
  }
}
```

上記のファイルが存在しない場合、作成されるまで待つ場合のrequestオブジェクトで指定するリクエストメッセージボディ(amazon_s3_main_body.json)の例

```
{
    "bucketName": "test-bucket1",
    "fileName": "test1.txt",
    "option": "wait"
}
```

上記のファイルが存在しない場合、作成されるまで待つ場合のmonitoringオブジェクトで指定するリクエストメッセージボディ(amazon_s3_moni_body.json)の例

```
{
    "bucketName": "test-bucket1",
    "fileName": "test1.txt",
    "option": "wait"
}
```

上記のファイルが削除されるまで待つ場合のrequestオブジェクトで指定するリクエストメッセージボディ(amazon_s3_main_body.json)の例

```
{
    "bucketName": "test-bucket1",
    "fileName": "test1.txt",
    "option": "delete"
}
```

上記のファイルが削除されるまで待つ場合のmonitoringオブジェクトで指定するリクエストメッセージボディ(amazon_s3_moni_body.json)の例

```
{
    "bucketName": "test-bucket1",
    "fileName": "test1.txt",
    "option": "delete"
}
```

ファイルが更新されるまで待つ場合の例

```
{
  "request": {
    "url": "https://3abcdef7argst12345xs66kja0nmvss.lambda-url.ap-northeast-1.on.aws/",
    "method": "POST",
    "authName": "Aws_S3_Lambda",
    "proxyAuthName": "Proxy1",
    "takeoverKeys": {
      "firstState": "status",
      "firstUpdateDate": "updateDate"
```

```
    },
    "body": "/paramfiles/amazon_s3_main_body.json",
    "resHeader": "/response/amazon_s3_main_header_<date>.json",
    "resBody": "/response/amazon_s3_main_body_<date>.json",
    "timeout": 60,
    "tcpKeepAlive": {
      "idleTime": 60,
       "probeInterval": 10,
       "probeCount": 9
    }
  },
  "monitoring": {
    "url": "https://3abcdef7argst12345xs66kja0nmvss.lambda-url.ap-northeast-1.on.aws/",
    "method": "POST",
    "body": "/paramfiles/amazon_s3_moni_body.json",
    "resHeader": "/response/amazon_s3_moni_header_<date>.json",
    "resBody": "/response/amazon_s3_moni_body_<date>.json",
    "firstMonitorInterval": 3,
    "pollingInterval": 60,
    "pollingCount": 5,
    "pollingEndKey": "result",
    "pollingEndVal": ["SUCCEEDED"],
    "checkResultKey": "result",
    "checkResultVal": ["SUCCEEDED"]
  }
}
```

上記のファイルが更新されるまで待つ場合のrequestオブジェクトで指定するリクエストメッセージボディ(amazon_s3_main_body.json)の例

```
{
    "bucketName": "test-bucket1",
    "fileName": "test1.txt",
    "option": "update",
    "activity": "request"
}
```

上記のファイルが更新されるまで待つ場合のmonitoringオブジェクトで指定するリクエストメッセージボディ(amazon_s3_moni_body.json)の例

```
{
    "bucketName": "test-bucket1",
    "fileName": "test1.txt",
    "reqStatus": "<firstState>",
    "reqUpdateDate": "<firstUpdateDate>",
    "option": "update",
    "activity": "monitoring"
}
```

呼び出すLambda関数(API Version 2015-03-31)の例

使用言語はpython v3.12です。

```
import json
import base64
import boto3
import botocore.exceptions

class ParameterError(Exception):
    pass

def lambda_handler(event, context):
    try:
```

```
        # リクエストメッセージボディから各パラメタを取得
        arg = base64.b64decode(event["body"]).decode()
        arg = json.loads(arg)
        bucket_name = arg.get("bucketName", None)
        file_name = arg.get("fileName", None)
        option = arg.get("option", None)

        # パラメタのチェック
        if bucket_name is None or file_name is None or option is None:
            raise ParameterError("The request body does not contain the required parameters.")

        # S3のクライアントオブジェクトを作成
        s3_client = boto3.client('s3')

        # バケットの存在確認
        try:
            s3_client.head_bucket(Bucket=bucket_name)
        except botocore.exceptions.ClientError as e:
            if e.response['Error']['Code'] == '404':
                raise ParameterError("The specified bucket does not exist.") from e

            raise

        res = {}

        # ファイル作成待ち合わせの場合
        if option == "wait":
            result, status = chk_object_exists(s3_client, bucket_name, file_name)

        # ファイル削除待ち合わせの場合
        elif option == "delete":
            result, status = chk_object_delete(s3_client, bucket_name, file_name)

        # ファイル更新待ち合わせの場合
        elif option == "update":
            activity = arg.get("activity", None)
            if activity is None:
                raise ParameterError("The request body does not contain the required parameters.")
            req_status = arg.get("reqStatus", None)
            req_update_date = arg.get("reqUpdateDate", None)

            if activity == "request":
                pass
            elif activity == "monitoring":
                if req_status is None or req_update_date is None:
                    raise ParameterError("The request body does not contain the required parameters.")
            else:
                raise ParameterError("Invalid parameter for activity.")

            result, status, update_date = chk_object_update(s3_client, bucket_name, file_name, req_status,
req_update_date)
            res['updateDate'] = update_date

        else:
            raise ParameterError("Invalid parameter for option.")

        res['result'] = result
        res['status'] = status

        return {
            'statusCode': 200,
            'body': json.dumps(res, indent=4),
            'headers': {
```

```
                'Content-Type': 'application/json'
            }
        }

    # パラメタエラー発生時の処理
    except ParameterError as e:
        result = {"error": str(e)}
        return {
            'statusCode': 400,
            'body': json.dumps(result, indent=4),
            'headers': {
                'Content-Type': 'application/json'
            }
        }

    # その他のエラー発生時の処理
    except Exception as e:
        result = {"error": str(e)}
        return {
            'statusCode': 500,
            'body': json.dumps(result, indent=4),
            'headers': {
                'Content-Type': 'application/json'
            }
        }


# ファイル作成待ち合わせ用関数
def chk_object_exists(s3_client, bucket_name, file_name):
    try:
        s3_client.head_object(Bucket=bucket_name, Key=file_name)
        result = 'SUCCEEDED'
        status = 'EXIST'
    except botocore.exceptions.ClientError as e:
        if e.response["Error"]["Code"] == "404":
            result = 'WAITING'
            status = 'NOT_EXIST'
        else:
            raise
    return result, status


# ファイル削除待ち合わせ用関数
def chk_object_delete(s3_client, bucket_name, file_name):
    try:
        s3_client.head_object(Bucket=bucket_name, Key=file_name)
        result = 'WAITING'
        status = 'EXIST'
    except botocore.exceptions.ClientError as e:
        if e.response['Error']['Code'] == '404':
            result = 'SUCCEEDED'
            status = 'NOT_EXIST'
        else:
            raise
    return result, status


# ファイル更新待ち合わせ用関数
def chk_object_update(s3_client, bucket_name, file_name, req_status, req_update_date):
    try:
        response = s3_client.head_object(Bucket=bucket_name, Key=file_name)
        current_update_date = response['LastModified'].strftime('%Y/%m/%d %H:%M:%S')
```

```
        if req_status == "WAITING_FOR_UPDATE":
            if current_update_date != req_update_date:
                result = 'SUCCEEDED'
                status = 'UPDATED'
            else:
                result = 'WAITING'
                status = 'WAITING_FOR_UPDATE'
            return result, status, current_update_date

        if req_status == "WAITING_FOR_CREATE":
            result = 'SUCCEEDED'
            status = 'FILE_CREATED'
            return result, status, current_update_date

        result = 'WAITING'
        status = 'WAITING_FOR_UPDATE'
        return result, status, current_update_date

    except botocore.exceptions.ClientError as e:
        if e.response['Error']['Code'] == '404':
            result = 'WAITING'
            status = 'WAITING_FOR_CREATE'
            update_date = ''
            return result, status, update_date

        raise
```

20.9.1.5 AWS Backupサービスと連携する例

AWS Backupサービス(API Version 2023-12-06)のジョブ(Amazon S3のオブジェクトのバックアップ)を実行、監視、強制終了する場合の例を示します。

```
{
  "request": {
    "url": "https://backup.ap-northeast-1.amazonaws.com/backup-jobs/",
    "method": "PUT",
    "authName": "Aws_backup",
    "proxyAuthName": "Proxy1",
    "body": "/paramfiles/aws_backup_main_body.json",
    "resHeader": "/response/aws_backup_main_header_<date>.json",
    "resBody": "/response/aws_backup_main_body_<date>.json",
    "timeout": 60,
    "takeoverKeys": {
      "BACKUPJOBID": "BackupJobId"
    }
  },
  "monitoring": {
    "url": "https://backup.ap-northeast-1.amazonaws.com/backup-jobs/<BACKUPJOBID>",
    "method": "GET",
    "resHeader": "/response/aws_backup_moni_header_<date>.json",
    "resBody": "/response/aws_backup_moni_body_<date>.json",
    "pollingInterval": 60,
    "pollingCount": 5,
    "pollingEndKey": "State",
    "pollingEndVal": ["COMPLETED","FAILED","ABORTED", "EXPIRED"],
    "checkResultKey": "State",
    "checkResultVal": ["COMPLETED"]
  },
  "termination": {
    "url": "https://backup.ap-northeast-1.amazonaws.com/backup-jobs/<BACKUPJOBID>",
    "method": "POST",
    "resHeader": "/response/aws_backup_term_header_<date>.json",
    "resBody": "/response/aws_backup_term_body_<date>.json"
```

```
  }
}
```

上記のrequestオブジェクトで指定するリクエストメッセージボディ(aws_backup_main_body.json)の例

```
{
  "BackupVaultName": "test",
  "ResourceArn": "arn:aws:s3:::test-backup",
  "IamRoleArn": "arn:aws:iam::123456789012:role/test_backup",
  "Lifecycle": {
        "DeleteAfterDays": 1
    }
}
```

20.9.1.6 Amazon Relational Database Service (Amazon RDS)サービスと連携する例

Amazon Relational Database Service (Amazon RDS)サービス(API Version 2014-10-31)のジョブ(RDSインスタンスの起動)を実行、監視する場合の例

```
{
  "request": {
    "url": "https://rds.ap-northeast-1.amazonaws.com/",
    "method": "POST",
    "authName": "Aws_rds",
    "proxyAuthName": "Proxy1",
    "resHeader": "/response/amazon_rds_main_header_<date>.json",
    "resBody": "/response/amazon_rds_main_body_<date>.json",
    "query": {
      "DBInstanceIdentifier": "rs-rds-database-1",
      "Action": "StartDBInstance",
      "Version": "2014-10-31"
    },
    "timeout": 60
  },
  "monitoring": {
    "url": "https://rds.ap-northeast-1.amazonaws.com/",
    "method": "GET",
    "resHeader": "/response/amazon_rds_moni_header_<date>.json",
    "resBody": "/response/amazon_rds_moni_body_<date>.json",
    "query": {
      "DBInstanceIdentifier": "rs-rds-database-1",
      "Action": "DescribeDBInstances",
      "Version": "2014-10-31"
    },
    "pollingInterval": 60,
    "pollingCount": 5,
    "pollingEndKey":
"DescribeDBInstancesResponse.DescribeDBInstancesResult.DBInstances.DBInstance.DBInstanceStatus",
    "pollingEndVal": ["available","failed"],
    "checkResultKey":
"DescribeDBInstancesResponse.DescribeDBInstancesResult.DBInstances.DBInstance.DBInstanceStatus",
    "checkResultVal": ["available"]
  }
}
```

Amazon Relational Database Service (Amazon RDS)サービス(API Version 2014-10-31)のジョブ(RDSインスタンスの停止)を実行、監視する場合の例

```
{
  "request": {
    "url": "https://rds.ap-northeast-1.amazonaws.com/",
    "method": "POST",
```

```
    "authName": "Aws_rds",
    "proxyAuthName": "Proxy1",
    "resHeader": "/response/amazon_rds_main_header_<date>.json",
    "resBody": "/response/amazon_rds_main_body_<date>.json",
    "query": {
      "DBInstanceIdentifier": "rs-rds-database-1",
      "Action": "StopDBInstance",
      "Version": "2014-10-31"
    },
    "timeout": 60
  },
  "monitoring": {
    "url": "https://rds.ap-northeast-1.amazonaws.com/",
    "method": "GET",
    "resHeader": "/response/amazon_rds_moni_header_<date>.json",
    "resBody": "/response/amazon_rds_moni_body_<date>.json",
    "query": {
      "DBInstanceIdentifier": "rs-rds-database-1",
      "Action": "DescribeDBInstances",
      "Version": "2014-10-31"
    },
    "pollingInterval": 60,
    "pollingCount": 5,
    "pollingEndKey":
"DescribeDBInstancesResponse.DescribeDBInstancesResult.DBInstances.DBInstance.DBInstanceStatus",
    "pollingEndVal": ["stopped","failed"],
    "checkResultKey":
"DescribeDBInstancesResponse.DescribeDBInstancesResult.DBInstances.DBInstance.DBInstanceStatus",
    "checkResultVal": ["stopped"]
  }
}
```

20.9.1.7 Amazon Elastic Compute Cloud (Amazon EC2)サービスと連携する例

Amazon Elastic Compute Cloud (Amazon EC2)サービス(API Version 2016-11-15)のジョブ(インスタンス起動)を実行、監視する場合の例を示します。

```
{
  "request": {
    "url": "https://ec2.ap-northeast-1.amazonaws.com/",
    "method": "POST",
    "authName": "Aws_ec2",
    "proxyAuthName": "Proxy1",
    "resHeader": "/response/amazon_ec2_main_header_<date>.json",
    "resBody": "/response/amazon_ec2_main_body_<date>.json",
    "query": {
      "Action": "StartInstances",
      "InstanceId.1": "i-1234567890abcdef0",
      "Version": "2016-11-15"
    },
    "timeout": 60
  },
  "monitoring": {
    "url": "https://ec2.ap-northeast-1.amazonaws.com/",
    "method": "GET",
    "resHeader": "/response/amazon_ec2_moni_header_<date>.json",
    "resBody": "/response/amazon_ec2_moni_body_<date>.json",
    "query": {
      "Action": "DescribeInstances",
      "InstanceId.1": "i-1234567890abcdef0",
      "Version": "2016-11-15"
```

```
    },
    "pollingEndKey": "DescribeInstancesResponse.reservationSet.item.instancesSet.item.instanceState.name",
    "pollingEndVal": ["running", "terminated"],
    "checkResultKey": "DescribeInstancesResponse.reservationSet.item.instancesSet.item.instanceState.name",
    "checkResultVal": ["running"]
  }
}
```

Amazon Elastic Compute Cloud (Amazon EC2)サービス(API Version 2016-11-15)のジョブ(インスタンス停止)を実行、監視する場合の例

```
{
  "request": {
    "url": "https://ec2.ap-northeast-1.amazonaws.com/",
    "method": "POST",
    "authName": "Aws_ec2",
    "proxyAuthName": "Proxy1",
    "resHeader": "/response/amazon_ec2_main_header_<date>.json",
    "resBody": "/response/amazon_ec2_main_body_<date>.json",
    "query": {
      "Action": "StopInstances",
      "InstanceId.1": "i-1234567890abcdef0",
      "Version": "2016-11-15"
    },
    "timeout": 60
  },
  "monitoring": {
    "url": "https://ec2.ap-northeast-1.amazonaws.com/",
    "method": "GET",
    "resHeader": "/response/amazon_ec2_moni_header_<date>.json",
    "resBody": "/response/amazon_ec2_moni_body_<date>.json",
    "query": {
      "Action": "DescribeInstances",
      "InstanceId.1": "i-1234567890abcdef0",
      "Version": "2016-11-15"
    },
    "pollingEndKey": "DescribeInstancesResponse.reservationSet.item.instancesSet.item.instanceState.name",
    "pollingEndVal": ["stopped", "terminated"],
    "checkResultKey": "DescribeInstancesResponse.reservationSet.item.instancesSet.item.instanceState.name",
    "checkResultVal": ["stopped"]
  }
}
```

20.9.1.8 Amazon Elastic Kubernetes Service (Amazon EKS)サービスと連携する例

Amazon Elastic Kubernetes Service (Amazon EKS)サービス(API Version 2011-01-01)のジョブ(ワーカーノード起動)を実行、監視する場合の例を示します。

```
{
  "request": {
    "url": "https://autoscaling.ap-northeast-1.amazonaws.com/",
    "method": "POST",
    "authName": "Aws_eks",
    "proxyAuthName": "Proxy1",
    "resHeader": "/response/amazon_eks_main_header_<date>.json",
    "resBody": "/response/amazon_eks_main_body_<date>.json",
    "query": {
      "Action": "UpdateAutoScalingGroup",
      "AutoScalingGroupName": "AutoScalingGrp1",
      "MinSize": "10",
      "MaxSize": "20",
```

```
      "DesiredCapacity": "20",
      "Version": "2011-01-01"
    },
    "timeout": 60
  },
  "monitoring": {
    "url": "https://autoscaling.ap-northeast-1.amazonaws.com/",
    "method": "GET",
    "resHeader": "/response/amazon_eks_moni_header_<date>.json",
    "resBody": "/response/amazon_eks_moni_body_<date>.json",
    "query": {
      "Action": "DescribeScalingActivities",
      "AutoScalingGroupName": "AutoScalingGrp1",
      "MaxRecords": "1",
      "Version": "2011-01-01"
    },
    "pollingInterval": 60,
    "pollingCount": 5,
    "pollingEndKey":
"DescribeScalingActivitiesResponse.DescribeScalingActivitiesResult.Activities.member.StatusCode",
    "pollingEndVal": ["Successful", "Failed", "Cancelled"],
    "checkResultKey":
"DescribeScalingActivitiesResponse.DescribeScalingActivitiesResult.Activities.member.StatusCode",
    "checkResultVal": ["Successful"]
  }
}
```

Amazon Elastic Kubernetes Service (Amazon EKS)サービス(API Version 2011-01-01)のジョブ(ワーカーノード停止)を実行、監視、強制終了する場合の例

```
{
  "request": {
    "url": "https://autoscaling.ap-northeast-1.amazonaws.com/",
    "method": "POST",
    "authName": "Aws_eks",
    "proxyAuthName": "Proxy1",
    "resHeader": "/response/amazon_eks_main_header_<date>.json",
    "resBody": "/response/amazon_eks_main_body_<date>.json",
    "query": {
      "Action": "UpdateAutoScalingGroup",
      "AutoScalingGroupName": "AutoScalingGrp1",
      "MinSize": "0",
      "MaxSize": "0",
      "DesiredCapacity": "0",
      "Version": "2011-01-01"
    },
    "timeout": 60
  },
  "monitoring": {
    "url": "https://autoscaling.ap-northeast-1.amazonaws.com/",
    "method": "GET",
    "resHeader": "/response/amazon_eks_moni_header_<date>.json",
    "resBody": "/response/amazon_eks_moni_body_<date>.json",
    "query": {
      "Action": "DescribeScalingActivities",
      "AutoScalingGroupName": "AutoScalingGrp1",
      "MaxRecords": "1",
      "Version": "2011-01-01"
    },
    "pollingInterval": 60,
    "pollingCount": 5,
    "pollingEndKey":
"DescribeScalingActivitiesResponse.DescribeScalingActivitiesResult.Activities.member.StatusCode",
```

```
    "pollingEndVal": ["Successful", "Failed", "Cancelled"],
    "checkResultKey":
"DescribeScalingActivitiesResponse.DescribeScalingActivitiesResult.Activities.member.StatusCode",
    "checkResultVal": ["Successful"]
  }
}
```

20.9.1.9 Azure Batchサービスと連携する例

Azure Batchサービス(API Version 2023-05-01.17.0)のジョブを実行、監視、強制終了する場合の例を示します。

```
{
  "request": {
    "url": "https://testbatch112233.japaneast.batch.azure.com/jobs/testJob/tasks",
    "method": "POST",
    "authName": "Azure-Batch",
    "proxyAuthName": "Proxy1",
    "query": {
      "api-version": "2023-05-01.17.0"
    },
    "header": {
      "Content-Type": "application/json;odata=minimalmetadata"
    },
    "body": "/paramfiles/azure_batch_main_body.json",
    "resHeader": "/response/azure_batch_main_header_<date>.json",
    "resBody": "/response/azure_batch_main_body_<date>.json",
    "timeout": 60
  },
  "monitoring": {
    "url": "https://testbatch112233.japaneast.batch.azure.com/jobs/testJob/tasks/testTask",
    "method": "GET",
    "query": {
      "api-version": "2023-05-01.17.0"
    },
    "resHeader": "/response/azure_batch_moni_header_<date>.json",
    "resBody": "/response/azure_batch_moni_body_<date>.json",
    "pollingInterval": 60,
    "pollingCount": 5,
    "pollingEndKey": "state",
    "pollingEndVal": ["completed"],
    "checkResultKey": "executionInfo.result",
    "checkResultVal": ["success"]
  },
  "termination": {
    "url": "https://testbatch112233.japaneast.batch.azure.com/jobs/testJob/tasks/testTask/terminate",
    "method": "POST",
    "query": {
      "api-version": "2023-05-01.17.0"
    },
    "resHeader": "/response/azure_batch_term_header_<date>.json",
    "resBody": "/response/azure_batch_term_body_<date>.json"
  }
}
```

上記のrequestオブジェクトで指定するリクエストメッセージボディ(azure_batch_main_body.json)の例

```
{
  "id": "testTask",
  "commandLine": "cmd /c ¥"set AZ_BATCH & timeout /t 90 > NUL¥""
}
```

20.9.1.10 Azure Functionsサービスと連携する例

Azure Functionsサービス(ランタイムバージョン4)の関数を実行する場合の例を示します。

```
{
  "request": {
    "url": "https://testfunc123456.azurewebsites.net/api/HttpTrigger1",
    "method": "GET",
    "authName":  "Azure-Functions",
    "proxyAuthName": "Proxy1",
    "query": {
      "name": "Function"
    },
    "resHeader": "/response/azure_func_main_header_<date>.json",
    "resBody": "/response/azure_func_main_body_<date>.json",
    "timeout": 60,
    "tcpKeepAlive": {
      "idleTime": 60,
       "probeInterval": 10,
       "probeCount": 9
    }
  }
}
```

20.9.1.11 Azure Logic Appsサービスと連携する例

Azure Logic Appsサービス(API Version 2016-06-01)のジョブを実行、監視、強制終了する場合の例を示します。

```
{
  "request": {
    "url": "https://prod-27.japaneast.logic.azure.com:443/workflows/abcd1234-5678-90ef-ghij-klmnopqrstuv/
triggers/manual/paths/invoke",
    "method": "POST",
    "authName": "Azure-Logic-Apps",
    "proxyAuthName": "Proxy1",
    "query": {
      "api-version": "2016-06-01"
    },
    "resHeader": "/response/azure_logicApps_main_header_<date>.json",
    "resBody": "/response/azure_logicApps_main_body_<date>.json",
    "takeoverKeys": {
      "RunId": "name"
    },
    "timeout": 60
  },
  "monitoring": {
    "url": "https://management.azure.com/subscriptions/12345678-1234-1234-1234-123456789abc/resourceGroups/
testLogicApps/providers/Microsoft.Logic/workflows/testlogicapps/runs/<RunId>",
    "method": "GET",
    "query": {
      "api-version": "2016-06-01"
    },
    "resHeader": "/response/azure_logicApps_moni_header_<date>.json",
    "resBody": "/response/azure_logicApps_moni_body_<date>.json",
    "pollingInterval": 60,
    "pollingCount": 5,
    "pollingEndKey": "properties.status",
    "pollingEndVal": ["Succeeded","Failed","Aborted","Cancelled","Timed out"],
    "checkResultKey": "properties.status",
    "checkResultVal": ["Succeeded"]
  },
  "termination": {
     "url": "https://management.azure.com/subscriptions/12345678-1234-1234-1234-123456789abc/resourceGroups/
```

```
testLogicApps/providers/Microsoft.Logic/workflows/testlogicapps/runs/<RunId>/cancel",
     "method": "POST",
     "query": {
       "api-version": "2016-06-01"
     },
     "resHeader": "/response/azure_logicApps_term_header_<date>.json",
     "resBody": "/response/azure_logicApps_term_body_<date>.json"
   }
}
```

20.9.1.12 Azure Blob Storageサービスと連携する例

Azure Blob Storageサービスのファイル待ち合わせをするAzure Functions関数(ランタイムバージョン4)を実行、監視する場合の例を示します。

Azure Blob Storageサービスを対象としたファイル待ち合わせでは、作成、削除、更新、サイズ更新を待ち合わせることができます。待ち合わせ方法は、monitoringオブジェクト内で指定します。

Azure Blob Storageサービスを対象とした上記のファイル待ち合わせをする場合はFunctions関数を介して行います。

注意

Azure Functionsを介して連携するサービスの使用例を利用する場合

本使用例を利用する際は、セキュリティを考慮し、以下を実施してください。

- 関数をデプロイする際は、適切な認証の設定、およびアクセス制限をしてください。
- 関数で指定する引数のチェックは必要に応じて追加してください。

ファイルが存在しない場合、作成されるまで待つ、または、ファイルが削除されるまで待つ場合の例

```
{
  "request": {
    "url": "https://test-blob-function.azurewebsites.net/api/chkfile_blob",
    "method": "POST",
    "authName":  "Azure-Blob-Functions",
    "proxyAuthName": "Proxy1",
    "body": "/paramfiles/azure_blobStorage_main_body.json",
    "resHeader": "/response/azure_blobStorage_main_header_<date>.json",
    "resBody": "/response/azure_blobStorage_main_body_<date>.json",
    "timeout": 60,
    "tcpKeepAlive": {
      "idleTime": 60,
      "probeInterval": 10,
      "probeCount": 9
    }
},
"monitoring": {
    "url": "https://test-blob-function.azurewebsites.net/api/chkfile_blob",
    "method": "POST",
    "body": "/paramfiles/azure_blobStorage_moni_body.json",
    "resHeader": "/response/azure_blobStorage_moni_header_<date>.json",
    "resBody": "/response/azure_blobStorage_moni_body_<date>.json",
    "pollingInterval": 60,
    "pollingCount": 5,
    "pollingEndKey": "result",
    "pollingEndVal": ["SUCCEEDED"],
    "checkResultKey": "result",
    "checkResultVal": ["SUCCEEDED"]
```

```
  }
}
```

上記のファイルが存在しない場合、作成されるまで待つ場合のrequestオブジェクトで指定するリクエストメッセージボディ(amazon_blobStorage_main_body.json)の例

```
{
    "storageAccountName": "test-storage-account1",
    "containerName": "test-container1",
    "fileName": "test1.txt",
    "option": "wait"
}
```

上記のファイルが存在しない場合、作成されるまで待つ場合のmonitoringオブジェクトで指定するリクエストメッセージボディ(amazon_blobStorage_moni_body.json)の例

```
{
    "storageAccountName": "test-storage-account1",
    "containerName": "test-container1",
    "fileName": "test1.txt",
    "option": "wait"
}
```

上記のファイルが削除されるまで待つ場合のrequestオブジェクトで指定するリクエストメッセージボディ(amazon_blobStorage_main_body.json)の例

```
{
    "storageAccountName": "test-storage-account1",
    "containerName": "test-container1",
    "fileName": "test1.txt",
    "option": "delete"
}
```

上記のファイルが削除されるまで待つ場合のmonitoringオブジェクトで指定するリクエストメッセージボディ(amazon_blobStorage_moni_body.json)の例

```
{
    "storageAccountName": "test-storage-account1",
    "containerName": "test-container1",
    "fileName": "test1.txt",
    "option": "delete"
}
```

ファイルが更新されるまで待つ場合の例

```
{
  "request": {
    "url": " https://test-blob-function.azurewebsites.net/api/chkfile_blob",
    "method": "POST",
    "authName":  "Azure-Blob-Functions",
    "proxyAuthName": "Proxy1",
    "takeoverKeys": {
      "firstState": "status",
      "firstUpdateDate": "updateDate"
    },
    "body": "/paramfiles/azure_blobStorage_main_body.json",
    "resHeader": "/response/azure_blobStorage_main_header_<date>.json",
    "resBody": "/response/azure_blobStorage_main_body_<date>.json",
    "timeout": 60,
    "tcpKeepAlive": {
      "idleTime": 60,
      "probeInterval": 10,
      "probeCount": 9
```

```
    }
},
"monitoring": {
    "url": "https://test-blob-function.azurewebsites.net/api/chkfile_blob",
    "method": "POST",
    "body": "/paramfiles/azure_blobStorage_moni_body.json",
    "resHeader": "/response/azure_blobStorage_moni_header_<date>.json",
    "resBody": "/response/azure_blobStorage_moni_body_<date>.json",
    "pollingInterval": 60,
    "pollingCount": 5,
    "pollingEndKey": "result",
    "pollingEndVal": ["SUCCEEDED"],
    "checkResultKey": "result",
    "checkResultVal": ["SUCCEEDED"]
    }
}
```

上記のファイルが更新されるまで待つ場合のrequestオブジェクトで指定するリクエストメッセージボディ(amazon_blobStorage_main_body.json)の例

```
{
    "storageAccountName": "test-storage-account1",
    "containerName": "test-container1",
    "fileName": "test1.txt",
    "option": "update",
    "activity": "request"
}
```

上記のファイルが更新されるまで待つ場合のmonitoringオブジェクトで指定するリクエストメッセージボディ(amazon_blobStorage_moni_body.json)の例

```
{
    "storageAccountName": "test-storage-account1",
    "containerName": "test-container1",
    "fileName": "test1.txt",
    "option": "update",
    "reqStatus": "<firstState>",
    "reqUpdateDate": "<firstUpdateDate>",
    "activity": "monitoring"
}
```

呼び出すFunctions関数(ランタイムバージョン4)の例

使用言語はpython v3.11です。

```
import json
import azure.functions as func
from azure.identity import ManagedIdentityCredential
from azure.storage.blob import BlobServiceClient

class ParameterError(Exception):
    pass

app = func.FunctionApp(http_auth_level=func.AuthLevel.ANONYMOUS)

@app.route(route="chkfile_blob")
def chkfile_blob(req: func.HttpRequest) -> func.HttpResponse:
    try:
        # リクエストメッセージボディから各パラメタを取得
        req_body = req.get_json()
        account_name = req_body.get("storageAccountName", None)
        container_name = req_body.get("containerName", None)
        file_name = req_body.get("fileName", None)
```

```
        option = req_body.get("option", None)

        # パラメタのチェック
        if account_name is None or container_name is None or file_name is None or option is None:
            raise ParameterError("The request body does not contain the required parameters.")

        # マネージドIDを使用して資格情報を取得
        credential = ManagedIdentityCredential()

        # Blob Storageのクライアントオブジェクトを作成
        blob_service_client = BlobServiceClient(
            account_url=f"https://{account_name}.blob.core.windows.net/",
            credential=credential
        )

        # コンテナの存在確認
        container_client = blob_service_client.get_container_client(container_name)
        if not container_client.exists():
            raise ParameterError("The specified container does not exist.")

        res = {}

        # ファイル作成待ち合わせの場合
        if option == "wait":
            result, status = chk_object_exists(blob_service_client, container_name, file_name)

        # ファイル削除待ち合わせの場合
        elif option == "delete":
            result, status = chk_object_delete(blob_service_client, container_name, file_name)

        # ファイル更新待ち合わせの場合
        elif option == "update":
            activity = req_body.get("activity", None)
            if activity is None:
                raise ParameterError("The request body does not contain the required parameters.")
            req_status = req_body.get("reqStatus", None)
            req_update_date = req_body.get("reqUpdateDate", None)

            if activity == "request":
                pass
            elif activity == "monitoring":
                if req_status is None or req_update_date is None:
                    raise ParameterError("The request body does not contain the required parameters.")
            else:
                raise ParameterError("Invalid parameter for activity.")

            result, status, update_date = chk_object_update(blob_service_client, container_name, file_name,
req_status, req_update_date)
            res['updateDate'] = update_date

        else:
            raise ParameterError("Invalid parameter for option.")

        res['result'] = result
        res['status'] = status

        return func.HttpResponse(
            json.dumps(res, indent=4),
            status_code=200,
            headers={"Content-Type": "application/json"}
        )

    # パラメタエラー発生時の処理
```

```
    except ParameterError as e:
        result = {"error": str(e)}
        return func.HttpResponse(
            json.dumps(result, indent=4),
            status_code=400,
            headers={"Content-Type": "application/json"}
        )

    # その他のエラー発生時の処理
    except Exception as e:
        result = {"error": str(e)}
        return func.HttpResponse(
            json.dumps(result, indent=4),
            status_code=500,
            headers={"Content-Type": "application/json"}
        )


# ファイル作成待ち合わせ用関数
def chk_object_exists(blob_service_client, container_name, file_name):
    blob_client = blob_service_client.get_blob_client(container=container_name, blob=file_name)
    exists = blob_client.exists()

    if exists:
        result = 'SUCCEEDED'
        status = 'EXIST'
    else:
        result = 'WAITING'
        status = 'NOT_EXIST'

    return result, status


# ファイル削除待ち合わせ用関数
def chk_object_delete(blob_service_client, container_name, file_name):
    blob_client = blob_service_client.get_blob_client(container=container_name, blob=file_name)
    exists = blob_client.exists()

    if exists:
        result = 'WAITING'
        status = 'EXIST'
    else:
        result = 'SUCCEEDED'
        status = 'NOT_EXIST'

    return result, status


# ファイル更新待ち合わせ用関数
def chk_object_update(blob_service_client, container_name, file_name, req_status, req_update_date):
    blob_client = blob_service_client.get_blob_client(container=container_name, blob=file_name)
    exists = blob_client.exists()

    if exists:
        properties = blob_client.get_blob_properties()
        current_update_date = properties.last_modified.strftime('%Y/%m/%d %H:%M:%S')

        if req_status == "WAITING_FOR_UPDATE":
            if current_update_date != req_update_date:
                result = 'SUCCEEDED'
                status = 'UPDATED'
            else:
                result = 'WAITING'
```

```
                status = 'WAITING_FOR_UPDATE'
            return result, status, current_update_date

        if req_status == "WAITING_FOR_CREATE":
            result = 'SUCCEEDED'
            status = 'FILE_CREATED'
            return result, status, current_update_date

        result = 'WAITING'
        status = 'WAITING_FOR_UPDATE'
        return result, status, current_update_date

    result = 'WAITING'
    status = 'WAITING_FOR_CREATE'
    update_date = ''
    return result, status, update_date
```

20.9.1.13 Azure Backupサービスと連携する例

Azure Backupサービスのジョブ(Azure Blob Storageのオブジェクトのバックアップ)を実行、監視、強制終了する場合の例を示します。

Azure Backupサービスのジョブの監視は、Functions関数を介して行います。

注意

Azure Functionsを介して連携するサービスの使用例を利用する場合

本使用例を利用する際は、セキュリティを考慮し、以下を実施してください。

- 関数をデプロイする際は、適切な認証の設定、およびアクセス制限をしてください。
- 関数で指定する引数のチェックは必要に応じて追加してください。

```
{
  "request": {
    "url": "https://test-backup-function.azurewebsites.net/api/backup",
    "method": "POST",
    "authName": "Azure-Backup-Functions",
    "proxyAuthName": "Proxy1",
    "takeoverKeys": {
        "BACKUPJOBID": "backupJobId"
    },
    "body": "/paramfiles/azure_batch_main_body.json",
    "resHeader": "/response/azure_backup_main_header_<date>.json",
    "resBody": "/response/azure_backup_main_body_<date>.json",
    "timeout": 60,
    "tcpKeepAlive": {
      "idleTime": 60,
       "probeInterval": 10,
       "probeCount": 9
    }
  },
  "monitoring": {
    "url": "https://test-backup-function.azurewebsites.net/api/backup",
    "method": "POST",
    "body": "/paramfiles/azure_batch_moni_body.json",
    "resHeader": "/response/azure_backup_moni_header_<date>.json",
    "resBody": "/response/azure_backup_moni_body_<date>.json",
    "pollingInterval": 60,
    "pollingCount": 5,
    "pollingEndKey": "properties.status",
```

```
    "pollingEndVal": ["Completed","Failed","CompletedWithWarnings","Cancelled"],
    "checkResultKey": "properties.status",
    "checkResultVal": ["Completed"]
  },
  "termination": {
    "url": "https://test-backup-function.azurewebsites.net/api/backup",
    "method": "POST",
    "body": "/paramfiles/azure_batch_term_body.json",
    "resHeader": "/response/azure_backup_term_header_<date>.json",
    "resBody": "/response/azure_backup_term_body_<date>.json"
  }
}
```

上記のrequestオブジェクトで指定するリクエストメッセージボディ(azure_backup_main_body.json)の例

```
{
    "activity": "request",
    "subscriptionId": "12345678-1234-1234-1234-123456789abc",
    "resourceGroupName": "test-grp1",
    "containerName": "test-container1",
    "vmName": "test-vm1"
}
```

上記のmonitoringオブジェクトで指定するリクエストメッセージボディ(azure_backup_moni_body.json)の例

```
{
    "activity": "monitoring",
    "subscriptionId": "12345678-1234-1234-1234-123456789abc",
    "resourceGroupName": "test-grp1",
    "containerName": "test-container1",
    "backupJobId": "<BACKUPJOBID>"
}
```

上記のterminationオブジェクトで指定するリクエストメッセージボディ(azure_backup_term_body.json)の例

```
{
    "activity": "termination",
    "subscriptionId": "12345678-1234-1234-1234-123456789abc",
    "resourceGroupName": "test-grp1",
    "containerName": "test-container1",
    "backupJobId": "<BACKUPJOBID>"
}
```

呼び出すFunctions関数(ランタイムバージョン4)の例

使用言語はpython v3.11です。

```
import json
import time
import re
import azure.functions as func
import requests
from azure.identity import ManagedIdentityCredential

class ParameterError(Exception):
    pass

app = func.FunctionApp(http_auth_level=func.AuthLevel.ANONYMOUS)

@app.route(route="backup", auth_level=func.AuthLevel.ANONYMOUS)
def backup(req: func.HttpRequest) -> func.HttpResponse:
    try:
        # リクエストメッセージボディから各パラメタを取得
        req_body = req.get_json()
```

```
        activity = req_body.get('activity', None)
        subscription_id = req_body.get('subscriptionId', None)
        resource_group = req_body.get('resourceGroupName', None)
        vault_name = req_body.get('containerName', None)

        # 正規表現パターン
        valid_param_pattern = re.compile(r'^[a-zA-Z0-9-_]+$')

        # パラメタのチェック
        if activity is None or subscription_id is None or resource_group is None or vault_name is None:
            raise ParameterError("The request body does not contain the required parameters.")

       if not (valid_param_pattern.match(subscription_id) and valid_param_pattern.match(resource_group) and
valid_param_pattern.match(vault_name)):
            raise ParameterError("The parameter format is incorrect.")

        # Azure認証情報を取得
        credential = ManagedIdentityCredential()
        token = credential.get_token('https://management.azure.com/.default')
        headers = {
            'Authorization': 'Bearer ' + token.token,
            'Content-Type': 'application/json'
        }

        # バックアップの実行処理
        if activity == 'request':
            # パラメタのチェック
            vm_name = req_body.get('vmName', None)
            if vm_name is None:
                raise ParameterError("The request body does not contain the required parameters.")
            if not valid_param_pattern.match(vm_name):
                raise ParameterError("The parameter format is incorrect.")

            # バックアップ開始
            url = f"https://management.azure.com/Subscriptions/{subscription_id}/resourceGroups/
{resource_group}/providers/Microsoft.RecoveryServices/vaults/{vault_name}/backupFabrics/Azure/
protectionContainers/iaasvmcontainer;iaasvmcontainerv2;{resource_group};{vm_name}/protectedItems/
VM;iaasvmcontainerv2;{resource_group};{vm_name}/backup?api-version=2024-04-01"
            req_body = {
                "properties": {
                    "objectType": "IaasVMBackupRequest"
                }
            }
            response = requests.post(url, headers=headers, json=req_body, timeout=60)
            response.raise_for_status()

            # バックアップジョブIDの取得
            url_id = response.headers['Azure-AsyncOperation']
            # ループ回数は必要に応じて適宜変更可能
            for _ in range(30):
                response = requests.get(url_id, headers=headers, timeout=60)
                response.raise_for_status()
                response_json = response.json()
                if 'properties' in response_json and 'jobId' in response_json['properties']:
                    break
                time.sleep(1)
            else:
                raise Exception("A timeout occurred while waiting for a response.")
            return func.HttpResponse(
                json.dumps({"backupJobId": response_json['properties']['jobId']}, indent=4),
                status_code=200,
                headers={"Content-Type": "application/json"}
            )
```

```
        # バックアップジョブの状態取得処理
        if activity == 'monitoring':
            # パラメタのチェック
            job_id = req_body.get('backupJobId', None)
            if job_id is None:
                raise ParameterError("The request body does not contain the required parameters.")
            if not valid_param_pattern.match(job_id):
                raise ParameterError("The parameter format is incorrect.")

            # バックアップジョブの状態取得
            url = f"https://management.azure.com/Subscriptions/{subscription_id}/resourceGroups/
{resource_group}/providers/Microsoft.RecoveryServices/vaults/{vault_name}/backupJobs/{job_id}?api-
version=2024-04-01"
            response = requests.get(url, headers=headers, timeout=60)
            response.raise_for_status()
            return func.HttpResponse(
                json.dumps(response.json(), indent=4),
                status_code=200,
                headers={"Content-Type": "application/json"}
            )

        # バックアップジョブ停止処理
        if activity == 'termination':
            # パラメタのチェック
            job_id = req_body.get('backupJobId', None)
            if job_id is None:
                raise ParameterError("The request body does not contain the required parameters.")
            if not valid_param_pattern.match(job_id):
                raise ParameterError("The parameter format is incorrect.")

            # バックアップの中止
            url = f"https://management.azure.com/Subscriptions/{subscription_id}/resourceGroups/
{resource_group}/providers/Microsoft.RecoveryServices/vaults/{vault_name}/backupJobs/{job_id}/cancel?api-
version=2024-04-01"
            response = requests.post(url, headers=headers, timeout=60)
            response.raise_for_status()
            return func.HttpResponse(
                status_code=200
            )

        raise ParameterError("Invalid parameter for activity.")


    # パラメタエラー発生時の処理
    except ParameterError as e:
        result = {"error": str(e)}
        return func.HttpResponse(
            json.dumps(result, indent=4),
            status_code=400,
            headers={"Content-Type": "application/json"}
        )

    # その他のエラー発生時の処理
    except Exception as e:
        result = {"error": str(e)}
        return func.HttpResponse(
            json.dumps(result, indent=4),
            status_code=500,
            headers={"Content-Type": "application/json"}
        )
```

20.9.1.14 Azure SQL Managed Instanceサービスと連携する例

Azure SQL Managed Instanceサービス(API Version 2021-11-01)のジョブ(インスタンスを起動)を実行、監視する場合の例を示します。

```
{
  "request": {
    "url": "https://management.azure.com/subscriptions/12345678-1234-1234-1234-123456789abc/resourceGroups/
test_group/providers/Microsoft.Sql/managedInstances/rs-sqlmi/start",
    "method": "POST",
    "authName": "Azure-SQL-Mi",
    "proxyAuthName": "Proxy1",
    "query": {
      "api-version": "2021-11-01"
    },
    "resHeader": "/response/azure_sql_mi_main_header_<date>.json",
    "resBody": "/response/azure_sql_mi_main_body_<date>.json",
    "timeout": 60
  },
  "monitoring": {
    "url": "https://management.azure.com/subscriptions/12345678-1234-1234-1234-123456789abc/resourceGroups/
test_group/providers/Microsoft.Sql/managedInstances/rs-sqlmi",
    "query": {
      "api-version": "2021-11-01"
    },
    "method": "GET",
    "resHeader": "/response/azure_sql_mi_moni_header_<date>.json",
    "resBody": "/response/azure_sql_mi_moni_body_<date>.json",
    "pollingInterval": 60,
    "pollingCount": 5,
    "pollingEndKey": "properties.provisioningState",
    "pollingEndVal": ["Succeeded","Failed","Cancelled","Deleted","TimedOut"],
    "checkResultKey": "properties.provisioningState",
    "checkResultVal": ["Succeeded"]
  }
}
```

Azure SQL Managed Instanceサービス(API Version 2021-11-01)のジョブ(インスタンスを停止)を実行、監視する場合の例を示します。

```
{
  "request": {
    "url": "https://management.azure.com/subscriptions/12345678-1234-1234-1234-123456789abc/resourceGroups/
test_group/providers/Microsoft.Sql/managedInstances/rs-sqlmi/stop",
    "method": "POST",
    "authName": "Azure-SQL-Mi",
    "proxyAuthName": "Proxy1",
    "query": {
      "api-version": "2021-11-01"
    },
    "resHeader": "/response/azure_sql_mi_main_header_<date>.json",
    "resBody": "/response/azure_ sql_mi_main_body_<date>.json",
    "timeout": 60
  },
  "monitoring": {
    "url": "https://management.azure.com/subscriptions/12345678-1234-1234-1234-123456789abc/resourceGroups/
test_group/providers/Microsoft.Sql/managedInstances/rs-sqlmi",
    "query": {
      "api-version": "2021-11-01"
    },
    "method": "GET",
    "resHeader": "/response/azure_sql_mi_moni_header_<date>.json",
    "resBody": "/response/azure_sql_mi_moni_body_<date>.json",
```

```
    "pollingInterval": 60,
    "pollingCount": 5,
    "pollingEndKey": "properties.provisioningState",
    "pollingEndVal": ["Succeeded","Failed","Cancelled","Deleted","TimedOut"],
    "checkResultKey": "properties.provisioningState",
    "checkResultVal": ["Succeeded"]
  }
}
```

20.9.1.15 Azure Virtual Machinesサービスと連携する例

Azure Virtual Machinesサービス(API Version 2024-03-01)のジョブ(インスタンス起動)を実行、監視する場合の例を示します。

```
{
  "request": {
    "url": "https://management.azure.com/subscriptions/12345678-1234-1234-1234-123456789abc/resourceGroups/
test -group/providers/Microsoft.Compute/virtualMachines/vm-test/start",
    "method": "POST",
    "authName": "Azure-VM",
    "proxyAuthName": "Proxy1",
    "query": {
      "api-version": "2024-03-01"
    },
    "resHeader": "/response/azure_vm_main_header_<date>.json",
    "resBody": "/response/azure_vm_main_body_<date>.json",
    "timeout": 60
  },
  "monitoring": {
    "url": "https://management.azure.com/subscriptions/12345678-1234-1234-1234-123456789abc/resourceGroups/
test- group/providers/Microsoft.Compute/virtualMachines/vm-test/instanceView",
    "query": {
      "api-version": "2024-03-01"
    },
    "method": "GET",
    "resHeader": "/response/azure_vm_moni_header_<date>.json",
    "resBody": "/response/azure_vm_moni_body_<date>.json",
    "pollingInterval": 60,
    "pollingCount": 5,
    "pollingEndKey": "statuses.[0].displayStatus",
    "pollingEndVal": ["Provisioning succeeded", "Provisioning failed"],
    "checkResultKey": "statuses.[0].displayStatus",
    "checkResultVal": ["Provisioning succeeded"]
  }
}
```

Azure Virtual Machinesサービス(API Version 2024-03-01)のジョブ(インスタンス停止)を実行、監視する場合の例を示します。

```
{
  "request": {
    "url": "https://management.azure.com/subscriptions/12345678-1234-1234-1234-123456789abc/resourceGroups/
test-group/providers/Microsoft.Compute/virtualMachines/vm-test/deallocate",
    "method": "POST",
    "authName": "Azure-VM",
    "proxyAuthName": "Proxy1",
    "query": {
      "api-version": "2024-03-01"
    },
    "resHeader": "/response/azure_vm_main_header_<date>.json",
    "resBody": "/response/azure_vm_main_body_<date>.json",
    "timeout": 60
  },
```

```
  "monitoring": {
    "url": "https://management.azure.com/subscriptions/12345678-1234-1234-1234-123456789abc/resourceGroups/
test- group/providers/Microsoft.Compute/virtualMachines/vm-test/instanceView",
    "query": {
      "api-version": "2024-03-01"
    },
    "method": "GET",
    "resHeader": "/response/azure_vm_moni_header_<date>.json",
    "resBody": "/response/azure_vm_moni_body_<date>.json",
    "pollingInterval": 60,
    "pollingCount": 5,
    "pollingEndKey": "statuses.[0].displayStatus",
    "pollingEndVal": ["Provisioning succeeded", "Provisioning failed"],
    "checkResultKey": "statuses.[0].displayStatus",
    "checkResultVal": ["Provisioning succeeded"]
  }
}
```

20.9.1.16 Azure Kubernetes Service (AKS)サービスと連携する例

Azure Kubernetes Service (AKS)サービス(API Version 2024-02-01)のジョブ(クラスタの起動)を実行、監視、強制終了する場合の例を示します。

```
{
  "request": {
    "url": "https://management.azure.com/subscriptions/12345678-1234-1234-1234-123456789abc/resourcegroups/
test-group/providers/Microsoft.ContainerService/managedClusters/testcluster/start",
    "method": "POST",
    "authName": "Azure-AKS",
    "proxyAuthName": "Proxy1",
    "query": {
      "api-version": "2024-02-01"
    },
    "resHeader": "/response/azure_aks_main_header_<date>.json",
    "resBody": "/response/azure_aks_main_body_<date>.json",
    "timeout": 60
  },
  "monitoring": {
    "url": "https://management.azure.com/subscriptions/12345678-1234-1234-1234-123456789abc/resourcegroups/
test-group/providers/Microsoft.ContainerService/managedClusters/testcluster",
    "query": {
      "api-version": "2024-02-01"
    },
    "method": "GET",
    "resHeader": "/response/azure_aks_moni_header_<date>.json",
    "resBody": "/response/azure_aks_moni_body_<date>.json",
    "pollingInterval": 60,
    "pollingCount": 5,
    "pollingEndKey": "properties.provisioningState",
    "pollingEndVal": ["Succeeded","Failed","Canceled"],
    "checkResultKey": "properties.provisioningState",
    "checkResultVal": ["Succeeded"]
  },
  "termination": {
    "url": "https://management.azure.com/subscriptions/12345678-1234-1234-1234-123456789abc/resourcegroups/
test-group/providers/Microsoft.ContainerService/managedClusters/testcluster/abort",
    "method": "POST",
    "query": {
      "api-version": "2024-02-01"
    },
    "resHeader": "/response/azure_aks_term_header_<date>.json",
    "resBody": "/response/azure_aks_term_body_<date>.json"
```

```
  }
}
```

Azure Kubernetes Service (AKS)サービス(API Version 2024-02-01)のジョブ(クラスタの停止)を実行、監視、強制終了する場合の例を示します。

```
{
  "request": {
    "url": "https://management.azure.com/subscriptions/12345678-1234-1234-1234-123456789abc/resourcegroups/test-group/providers/Microsoft.ContainerService/managedClusters/testcluster/stop",
    "method": "POST",
    "authName": "Azure-AKS",
    "proxyAuthName": "Proxy1",
    "query": {
      "api-version": "2024-02-01"
    },
    "resHeader": "/response/azure_aks_main_header_<date>.json",
    "resBody": "/response/azure_aks_main_body_<date>.json",
    "timeout": 60
  },
  "monitoring": {
    "url": "https://management.azure.com/subscriptions/12345678-1234-1234-1234-123456789abc/resourcegroups/test-group/providers/Microsoft.ContainerService/managedClusters/testcluster",
    "query": {
      "api-version": "2024-02-01"
    },
    "method": "GET",
    "resHeader": "/response/azure_aks_moni_header_<date>.json",
    "resBody": "/response/azure_aks_moni_body_<date>.json",
    "pollingInterval": 60,
    "pollingCount": 5,
    "pollingEndKey": "properties.provisioningState",
    "pollingEndVal": ["Succeeded","Failed","Canceled"],
    "checkResultKey": "properties.provisioningState",
    "checkResultVal": ["Succeeded"]
  },
  "termination": {
    "url": "https://management.azure.com/subscriptions/12345678-1234-1234-1234-123456789abc/resourcegroups/test-group/providers/Microsoft.ContainerService/managedClusters/testcluster/abort",
    "method": "POST",
    "query": {
      "api-version": "2024-02-01"
    },
    "resHeader": "/response/azure_aks_term_header_<date>.json",
    "resBody": "/response/azure_aks_term_body_<date>.json"
  }
}
```

20.9.1.17 Oracle Cloud Infrastructure Functions(OCI Functions)サービスと連携する例

Oracle Cloud Infrastructure Functions(OCI Functions)サービス(API Version 2018-12-01)のジョブを実行する場合の例を示します。

なお、"fn-intent"リクエストヘッダに "httprequest"を指定することで、REST実行ジョブは"Fn-Http-Status"レスポンスヘッダの値をHTTPレスポンスのステータスコードとして取得して、正常終了または異常終了の判定をします。"Fn-Http-Status"レスポンスヘッダの値はOCI Functionにデプロイしたファンクションで指定したHTTPレスポンスのステータスコードとなります。OCI Fucntionsと連携する場合以外では、本指定は不要です。

```
{
  "request": {
    "url":"https://3eko57tueak.functions.ap-tokyo-1.oci.oraclecloud.com/20181201/functions/
```

```
ocid1.fnfunc.oc40.ap-tokyo-1.ssssssssfeb7adlaoftudar9aktifisu34pnvx6ci67q7d6tan3g2sa1f38q/actions/invoke",
    "method": "POST",
    "authName":  "OCI-Functions",
    "proxyAuthName":"Proxy1",
    "header":{
        "fn-intent" : "httprequest"
    },
    "resHeader": "/response/oci_functions_main_header_<date>.json",
    "resBody": "/response/oci_functions_main_body_<date>.json",
    "timeout": 60
  }
}
```

20.9.1.18 Oracle Cloud Infrastructure Compute(OCI Compute)サービスと連携する例

Oracle Cloud Infrastructure Compute(OCI Compute)サービス(API Version 2016-09-18)のジョブ(インスタンス起動)を実行、監視する場合の例を示します。

```
{
  "request": {
    "url":"https://iaas.ap-tokyo-1.oraclecloud.com/20160918/instances/ocid1.instance.oc40.ap-
tokyo-1.aseagerty4fo345acp4j62cb4lk7kutebdunxvi3aqkkdae43rn6dmqlhi79q",
    "method": "POST",
    "authName":  "OCI-Compute",
    "proxyAuthName":"Proxy1",
    "query": {
        "action":"START"
    },
    "resHeader": "/response/oci_vm_start_main_header_<date>.json",
    "resBody": "/response/oci_vm_start_main_body_<date>.json",
    "timeout": 60
  },
  "monitoring": {
    "url":"https://iaas.ap-tokyo-1.oraclecloud.com/20160918/instances/ocid1.instance.oc40.ap-
tokyo-1.aseagerty4fo345acp4j62cb4lk7kutebdunxvi3aqkkdae43rn6dmqlhi79q",
    "method": "GET",
    "resHeader": "/response/oci_vm_start_moni_header_<date>.json",
    "resBody": "/response/oci_vm_start_moni_body_<date>.json",
    "pollingInterval": 60,
    "pollingCount": 5,
    "pollingEndKey": "lifecycleState",
    "pollingEndVal": ["RUNNING", "STOPPED","TERMINATED"],
    "checkResultKey": "lifecycleState",
    "checkResultVal": ["RUNNING"]
  }
}
```

Oracle Cloud Infrastructure Compute(OCI Compute)サービス(API Version 2016-09-18)のジョブ(インスタンス停止)を実行、監視する場合の例を示します。

```
{
  "request": {
    "url":"https://iaas.ap-tokyo-1.oraclecloud.com/20160918/instances/ocid1.instance.oc40.ap-
tokyo-1.aseagerty4fo345acp4j62cb4lk7kutebdunxvi3aqkkdae43rn6dmqlhi79q",
    "method": "POST",
    "authName":  " OCI-Compute",
    "proxyAuthName":"Proxy1",
    "query": {
        "action":"STOP"
    },
    "resHeader": "/response/oci_vm_stop_main_header_<date>.json",
```

```
    "resBody": "/response/oci_vm_stop_main_body_<date>.json",
    "timeout": 60
  },
  "monitoring": {
    "url":"https://iaas.ap-tokyo-1.oraclecloud.com/20160918/instances/ocid1.instance.oc40.ap-
tokyo-1.aseagerty4fo345acp4j62cb4lk7kutebdunxvi3aqkkdae43rn6dmqlhi79q",
    "method": "GET",
    "resHeader": "/response/oci_vm_stop_moni_header_<date>.json",
    "resBody": "/response/oci_vm_stop_moni_body_<date>.json",
    "pollingInterval": 60,
    "pollingCount": 5,
    "pollingEndKey": "lifecycleState",
    "pollingEndVal": ["STOPPED", "RUNNING","TERMINATED"],
    "checkResultKey": "lifecycleState",
    "checkResultVal": ["STOPPED"]
  }
}
```

20.9.1.19 Oracle Cloud Infrastructure Kubernetes Engine(OKE)サービスと連携する例

Oracle Cloud Infrastructure Kubernetes Engine(OKE)サービス(API Version 2018-02-22)のジョブ(管理対象ノード起動)を実行、監視する場合の例を示します。

```
{
  "request": {
    "url":"https://containerengine.ap-tokyo-1.oci.oraclecloud.com/20180222/nodePools/ocid1.nodepool.oc40.ap-
tokyo-1.cccccccccerga2karalchrht722wu236snmv8idfrde24jw5ofnyv6akriwuc",
    "method": "PUT",
    "authName":  "OCI-OKE",
    "proxyAuthName":"Proxy1",
    "takeoverKeysFromHeader":{
        "workRequestOCID":"opc-work-request-id"
    },
    "body":"/paramfiles/oci_oke_start_main_body.json",
    "resHeader": "/response/oci_oke_start_main_header_<date>.json",
    "resBody": "/response/oci_oke_start_main_body_<date>.json",
    "timeout": 60
  },
  "monitoring": {
    "url":"https://containerengine.ap-tokyo-1.oci.oraclecloud.com/20180222/workRequests/<workRequestOCID>",
    "method": "GET",
    "resHeader": "/response/oci_oke_start_moni_header_<date>.json",
    "resBody": "/response/oci_oke_start_moni_body_<date>.json",
    "pollingInterval": 60,
    "pollingCount": 5,
    "pollingEndKey": "status",
    "pollingEndVal": ["SUCCEEDED", "FAILED"],
    "checkResultKey": "status",
    "checkResultVal": ["SUCCEEDED"]
  }
}
```

上記のrequestオブジェクトで指定するリクエストメッセージボディ(oci_oke_start_main_body.json)の例

```
{
  "nodeConfigDetails":{
    "size":3
  }
}
```

Oracle Cloud Infrastructure Kubernetes Engine(OKE)サービス(API Version 2018-02-22)のジョブ(管理対象ノード停止)を実行、監視する場合の例を示します。

```
{
  "request": {
    "url":"https://containerengine.ap-tokyo-1.oci.oraclecloud.com/20180222/nodePools/ocid1.nodepool.oc40.ap-tokyo-1.cccccccccerga2karalchrht722wu236snmv8idfrde24jw5ofnyv6akriwuc",
    "method": "PUT",
    "authName":  "OCI-OKE",
    "proxyAuthName":"Proxy1",
    "takeoverKeysFromHeader":{
        "workRequestOCID":"opc-work-request-id"
    },
    "body":"/paramfiles/oci_oke_stop_main_body.json",
    "resHeader": "/response/oci_oke_stop_main_header_<date>.json",
    "resBody": "/response/oci_oke_stop_main_body_<date>.json",
    "timeout": 60
  },
  "monitoring": {
    "url":"https://containerengine.ap-tokyo-1.oci.oraclecloud.com/20180222/workRequests/<workRequestOCID>",
    "method": "GET",
    "resHeader": "/response/oci_oke_stop_moni_header_<date>.json",
    "resBody": "/response/oci_oke_stop_moni_body_<date>.json",
    "pollingInterval": 60,
    "pollingCount": 5,
    "pollingEndKey": "status",
    "pollingEndVal": ["SUCCEEDED", "FAILED"],
    "checkResultKey": "status",
    "checkResultVal": ["SUCCEEDED"]
  }
}
```

上記のrequestオブジェクトで指定するリクエストメッセージボディ(oci_oke_stop_main_body.json)の例

```
{
  "nodeConfigDetails":{
    "size":0
  }
}
```

Oracle Cloud Infrastructure Kubernetes Engine(OKE)サービス(API Version 2018-02-22)のジョブ(仮想ノード起動)を実行、監視する場合の例を示します。

```
{
  "request": {
    "url":"https://containerengine.ap-tokyo-1.oci.oraclecloud.com/20180222/virtualNodePools/ocid1.virtualnodepool.oc40.ap-tokyo-1.amaaaaaa4fo345aaticncpxwxizm5k3zgze467j3jmu35fgzyf7kfxf53qca",
    "method": "PUT",
    "authName":  "OCI-OKE",
    "proxyAuthName":"Proxy1",
    "takeoverKeysFromHeader":{
        "workRequestOCID":"opc-work-request-id"
    },
    "body":"/paramfiles/oci_oke_start_main_body.json",
    "resHeader": "/response/oci_oke_start_main_header_<date>.json",
    "resBody": "/response/oci_oke_start_main_body_<date>.json",
    "timeout": 60
  },
  "monitoring": {
    "url":"https://containerengine.ap-tokyo-1.oci.oraclecloud.com/20180222/workRequests/<workRequestOCID>",
    "method": "GET",
    "resHeader": "/response/oci_oke_start_moni_header_<date>.json",
    "resBody": "/response/oci_oke_start_moni_body_<date>.json",
```

```
    "pollingInterval": 60,
    "pollingCount": 5,
    "pollingEndKey": "status",
    "pollingEndVal": ["SUCCEEDED", "FAILED"],
    "checkResultKey": "status",
    "checkResultVal": ["SUCCEEDED"]
  }
}
```

上記のrequestオブジェクトで指定するリクエストメッセージボディ(oci_oke_start_main_body.json)の例

```
{
    "size":3
}
```

Oracle Cloud Infrastructure Kubernetes Engine(OKE)サービス(API Version 2018-02-22)のジョブ(仮想ノード停止)を実行、監視する場合の例を示します。

```
{
  "request": {
    "url":"https://containerengine.ap-tokyo-1.oci.oraclecloud.com/20180222/virtualNodePools/
ocid1.virtualnodepool.oc40.ap-tokyo-1.amaaaaaa4fo345aaticncpxwxizm5k3zgze467j3jmu35fgzyf7kfxf53qca",
    "method": "PUT",
    "authName":  "OCI-OKE",
    "proxyAuthName":"Proxy1",
    "takeoverKeysFromHeader":{
        "workRequestOCID":"opc-work-request-id"
    },
    "body":"/paramfiles/oci_oke_stop_main_body.json",
    "resHeader": "/response/oci_oke_stop_main_header_<date>.json",
    "resBody": "/response/oci_oke_stop_main_body_<date>.json",
    "timeout": 60
  },
  "monitoring": {
    "url":"https://containerengine.ap-tokyo-1.oci.oraclecloud.com/20180222/workRequests/<workRequestOCID>",
    "method": "GET",
    "resHeader": "/response/oci_oke_stop_moni_header_<date>.json",
    "resBody": "/response/oci_oke_stop_moni_body_<date>.json",
    "pollingInterval": 60,
    "pollingCount": 5,
    "pollingEndKey": "status",
    "pollingEndVal": ["SUCCEEDED", "FAILED"],
    "checkResultKey": "status",
    "checkResultVal": ["SUCCEEDED"]
  }
}
```

上記のrequestオブジェクトで指定するリクエストメッセージボディ(oci_oke_stop_main_body.json)の例

```
{
    "size":0
}
```

20.9.1.20 Oracle Autonomous Databaseサービスと連携する例

Oracle Autonomous Databaseサービス(API Version 2016-09-18)のジョブ(データベースサービス起動)を実行、監視する場合の例を示します。

```
{
  "request": {
    "url":"https://database.ap-tokyo-1.oraclecloud.com/20160918/autonomousDatabases/
ocid1.autonomousdatabase.oc40.ap-tokyo-1.akdicmel6fo289qqolg7yyz2sk9q0ox5dkjio3zsxx3eup6nljc2jx9l78sa/
actions/start",
```

```
    "method": "POST",
    "authName":  "OCI-AutonomousDB",
    "proxyAuthName":"Proxy1",
    "takeoverKeysFromHeader":{
        "workRequestOCID":"opc-work-request-id"
    },
    "resHeader": "/response/oci_autonomousDB_start_main_header_<date>.json",
    "resBody": "/response/oci_autonomousDB_start_main_body_<date>.json",
    "timeout": 60
  },
  "monitoring": {
    "url":"https://iaas.ap-tokyo-1.oraclecloud.com/20160918/workRequests/<workRequestOCID>",
    "method": "GET",
    "resHeader": "/response/oci_autonomousDB_start_moni_header_<date>.json",
    "resBody": "/response/oci_autonomousDB_start_moni_body_<date>.json",
    "pollingInterval": 60,
    "pollingCount": 5,
    "pollingEndKey": "status",
    "pollingEndVal": ["SUCCEEDED", "FAILED"],
    "checkResultKey": "status",
    "checkResultVal": ["SUCCEEDED"]
  }
}
```

Oracle Autonomous Databaseサービス(API Version 2016-09-18)のジョブ(データベースサービス停止)を実行、監視する場合の例を示します。

```
{
  "request": {
    "url":"https://database.ap-tokyo-1.oraclecloud.com/20160918/autonomousDatabases/
ocid1.autonomousdatabase.oc40.ap-tokyo-1.akdicmel6fo289qqolg7yyz2sk9q0ox5dkjio3zsxx3eup6nljc2jx9l78sa/
actions/stop",
    "method": "POST",
    "authName":  "OCI-AutonomousDB",
    "proxyAuthName":"Proxy1",
    "takeoverKeysFromHeader":{
        "workRequestOCID":"opc-work-request-id"
    },
    "resHeader": "/response/oci_autonomousDB_stop_main_header_<date>.json",
    "resBody": "/response/oci_autonomousDB_stop_main_body_<date>.json",
    "timeout": 60
  },
  "monitoring": {
    "url":"https://iaas.ap-tokyo-1.oraclecloud.com/20160918/workRequests/<workRequestOCID>",
    "method": "GET",
    "resHeader": "/response/oci_autonomousDB_stop_moni_header_<date>.json",
    "resBody": "/response/oci_autonomousDB_stop_moni_body_<date>.json",
    "pollingInterval": 60,
    "pollingCount": 5,
    "pollingEndKey": "status",
    "pollingEndVal": ["SUCCEEDED", "FAILED"],
    "checkResultKey": "status",
    "checkResultVal": ["SUCCEEDED"]
  }
}
```

20.9.1.21 Oracle Cloud Infrastructure Object Storage(OCI Object Storage)サービスと連携する例

Oracle Cloud Infrastructure Object Storage(OCI Object Storage)サービスのファイル待ち合わせをするFunctions関数(API Version 2018-12-01)を実行、監視する場合の例を示します。

OCI Object Storageサービスを対象としたファイル待ち合わせでは、作成、削除、更新、サイズ更新を待ち合わせることができます。待ち合わせ方法は、monitoringオブジェクト内で指定します。

OCI Object Storageサービスを対象とした上記のファイル待ち合わせをする場合は、Functions関数を介して行います。

注意

OCI Functionsを介して連携するサービスの使用例を利用する場合

本使用例を利用する際は、セキュリティを考慮し、以下を実施してください。

- 関数をデプロイする際は、適切な認証の設定、およびアクセス制限をしてください。
- 関数で指定する引数のチェックは必要に応じて追加してください。

ファイルが存在しない場合、作成されるまで待つ、または、ファイルが削除されるまで待つ場合の例

```
{
  "request": {
    "url":"https://8ike20qoeuv.functions.ap-tokyo-1.oci.oraclecloud.com/20181201/functions/
ocid1.fnfunc.oc40.ap-tokyo-1.sssssssfeb7adlaoftudar9aktifisu34pnvx6ci67q7d6tan3g2sa1f38q/actions/invoke",
    "method": "POST",
    "authName": "OCI_ObjectStorage",
    "proxyAuthName": "Proxy1",
    "header":{
        "fn-intent" : "httprequest"
    },
    "body": "/paramfiles/oci_objectStorage_main_body.json",
    "resHeader": "/response/oci_objectStorage_main_header_<date>.json",
    "resBody": "/response/oci_objectStorage_main_body_<date>.json",
    "timeout": 60
  },
  "monitoring": {
    "url":"https://8ike20qoeuv.functions.ap-tokyo-1.oci.oraclecloud.com/20181201/functions/
ocid1.fnfunc.oc40.ap-tokyo-1.sssssssfeb7adlaoftudar9aktifisu34pnvx6ci67q7d6tan3g2sa1f38q/actions/invoke",
    "method": "POST",
    "header":{
        "fn-intent" : "httprequest"
    },
    "body": "/paramfiles/oci_objectStorage_moni_body.json",
    "resHeader": "/response/oci_objectStorage_moni_header_<date>.json",
    "resBody": "/response/oci_objectStorage_moni_body_<date>.json",
    "firstMonitorInterval": 3,
    "pollingInterval": 60,
    "pollingCount": 5,
    "pollingEndKey": "result",
    "pollingEndVal": ["SUCCEEDED"],
    "checkResultKey": "result",
    "checkResultVal": ["SUCCEEDED"]
  }
}
```

上記のファイルが存在しない場合、作成されるまで待つ場合の requestオブジェクトで指定するリクエストメッセージボディ(oci_objectStorage_main_body.json)の例

```
{
    "namespaceName": "testnamespace",
    "bucketName": "test-bucket1",
    "fileName": "test1.txt",
    "option": "wait"
}
```

上記のファイルが存在しない場合、作成されるまで待つ場合の monitoringオブジェクトで指定するリクエストメッセージボディ(oci_objectStorage_moni_body.json)の例

```
{
    "namespaceName": "testnamespace",
    "bucketName": "test-bucket1",
    "fileName": "test1.txt",
    "option": "wait"
}
```

上記のファイルが削除されるまで待つ場合の requestオブジェクトで指定するリクエストメッセージボディ(oci_objectStorage_main_body.json)の例

```
{
    "namespaceName": "testnamespace",
    "bucketName": "test-bucket1",
    "fileName": "test1.txt",
    "option": "delete"
}
```

上記のファイルが削除されるまで待つ場合の monitoringオブジェクトで指定するリクエストメッセージボディ(oci_objectStorage_moni_body.json)の例

```
{
    "namespaceName": "testnamespace",
    "bucketName": "test-bucket1",
    "fileName": "test1.txt",
    "option": "delete"
}
```

ファイルが更新されるまで待つ場合の例

```
{
  "request": {
    "url":"https://8ike20qoeuv.functions.ap-tokyo-1.oci.oraclecloud.com/20181201/functions/
ocid1.fnfunc.oc40.ap-tokyo-1.sssssssfeb7adlaoftudar9aktifisu34pnvx6ci67q7d6tan3g2sa1f38q/actions/invoke",
    "method": "POST",
    "authName": "OCI_ObjectStorage",
    "proxyAuthName": "Proxy1",
    "header":{
        "fn-intent" : "httprequest"
    },
    "takeoverKeys": {
      "firstState": "status",
      "firstUpdateDate": "updateDate"
    },
    "body": "/paramfiles/oci_objectStorage_main_body.json",
    "resHeader": "/response/oci_objectStorage_main_header_<date>.json",
    "resBody": "/response/oci_objectStorage_main_body_<date>.json",
    "timeout": 60
  },
  "monitoring": {
    "url":"https://8ike20qoeuv.functions.ap-tokyo-1.oci.oraclecloud.com/20181201/functions/
ocid1.fnfunc.oc40.ap-tokyo-1.sssssssfeb7adlaoftudar9aktifisu34pnvx6ci67q7d6tan3g2sa1f38q/actions/invoke",
    "method": "POST",
    "header":{
        "fn-intent" : "httprequest"
    },
    "body": "/paramfiles/oci_objectStorage_moni_body.json",
    "resHeader": "/response/oci_objectStorage_moni_header_<date>.json",
    "resBody": "/response/oci_objectStorage_moni_body_<date>.json",
    "firstMonitorInterval": 3,
    "pollingInterval": 60,
    "pollingCount": 5,
```

```
    "pollingEndKey": "result",
    "pollingEndVal": ["SUCCEEDED"],
    "checkResultKey": "result",
    "checkResultVal": ["SUCCEEDED"]
  }
}
```

上記のファイルが更新されるまで待つ場合の requestオブジェクトで指定するリクエストメッセージボディ(oci_objectStorage_main_body.json)の例

```
{
    "namespaceName": "testnamespace",
    "bucketName": "test-bucket1",
    "fileName": "test1.txt",
    "option": "update",
    "activity": "request"
}
```

上記のファイルが更新されるまで待つ場合の monitoringオブジェクトで指定するリクエストメッセージボディ(oci_objectStorage_moni_body.json)の例

```
{
    "namespaceName": "testnamespace",
    "bucketName": "test-bucket1",
    "fileName": "test1.txt",
    "reqStatus": "<firstState>",
    "reqUpdateDate": "<firstUpdateDate>",
    "option": "update",
    "activity": "monitoring"
}
```

上記のファイルが削除されるまで待つ場合の requestオブジェクトで指定するリクエストメッセージボディ

呼び出すFunctions関数(API Version 2018-12-01)の例

使用言語はPython v3.11です。

```
import io
import oci
import json

from fdk import response

class ParameterError(Exception):
    pass

def handler(ctx, data: io.BytesIO = None):
    try:
        # リクエストボディから各パラメータを取得
        body = json.loads(data.getvalue().decode('utf-8'))
        namespace_name = body.get("namespaceName", None)
        bucket_name = body.get("bucketName", None)
        file_name = body.get("fileName", None)
        option = body.get("option", None)

        # パラメータのチェック
        if (namespace_name is None or bucket_name is None or file_name is None
or option is None):
            raise ParameterError("The request body does not contain the required parameters.")

        # 資格情報を取得
        signer = oci.auth.signers.get_resource_principals_signer()

        # Object Storageのクライアントオブジェクトを作成
```

```
        # プライベート・エンドポイント経由でアクセスしない場合
        object_storage_client = oci.object_storage.ObjectStorageClient({}, signer=signer)
        # プライベート・エンドポイント経由でアクセスする場合
        # private_endpoint = "##プライベート・エンドポイント##"
        # object_storage_client = oci.object_storage.ObjectStorageClient({}, signer=signer,
        #     service_endpoint=private_endpoint)

        # バケットの存在確認
        try:
            object_storage_client.head_bucket(namespace_name, bucket_name)
        except oci.exceptions.ServiceError as e:
            if e.status == 404:
                raise ParameterError("The specified bucket does not exist.") from e
            raise

        res = {}

        # ファイル作成待ち合わせの場合
        if option == "wait":
            result, status = chk_object_exists(object_storage_client, namespace_name,
bucket_name, file_name)

        # ファイル削除待ち合わせの場合
        elif option == "delete":
            result, status = chk_object_delete(object_storage_client, namespace_name,
bucket_name, file_name)

        # ファイル更新待ち合わせの場合
        elif option == "update":
            activity = body.get("activity", None)

            if activity is None:
                raise ParameterError("The request body does not contain the required parameters.")
            req_status = body.get("reqStatus", None)
            req_update_date = body.get("reqUpdateDate", None)

            if activity == "request":
                pass
            elif activity == "monitoring":
                if req_status is None or req_update_date is None:
                    raise ParameterError("The request body does not contain the required parameters")
            else:
                raise ParameterError("Invalid parameter for activity.")
            result, status, update_date = chk_object_update(object_storage_client, namespace_name,
bucket_name, file_name, req_status, req_update_date)
            res['updateDate'] = update_date
        else:
            raise ParameterError("Invalid parameter for option.")

        res['result'] = result
        res['status'] = status

        return response.Response(
            ctx,
            status_code=200,
            response_data=json.dumps(res, indent=4),
            headers={"Content-Type": "application/json"}
        )

    # パラメータエラー発生時の処理
    except ParameterError as e:
        result = {"error": str(e)}
        return response.Response(
```

```
            ctx,
            status_code=400,
            response_data=json.dumps(result),
            headers={"Content-Type": "application/json"}
        )

    # その他のエラー発生時の処理
    except Exception as e:
        result = {"error": str(e)}
        return response.Response(
            ctx,
            status_code=500,
            response_data=json.dumps(result, indent=4),
            headers={"Content-Type": "application/json"}
        )

# ファイル作成待ち合わせ用関数
def chk_object_exists(client, namespace_name, bucket_name, file_name):
    try:
        client.head_object(namespace_name, bucket_name, file_name)
        result = 'SUCCEEDED'
        status = 'EXIST'
    except oci.exceptions.ServiceError as e:
        if e.status == 404:
            result = 'WAITING'
            status = 'NOT_EXIST'
        else:
            raise
    return result, status

# ファイル削除待ち合わせ用関数
def chk_object_delete(client, namespace_name, bucket_name, file_name):
    try:
        client.head_object(namespace_name, bucket_name, file_name)
        result = 'WAITING'
        status = 'EXIST'
    except oci.exceptions.ServiceError as e:
        if e.status == 404:
            result = 'SUCCEEDED'
            status = 'NOT_EXIST'
        else:
            raise
    return result, status

# ファイル更新待ち合わせ用関数
def chk_object_update(client, namespace_name, bucket_name, file_name,
req_status, req_update_date):
    try:
        response = client.head_object(namespace_name, bucket_name, file_name)
        current_update_date = response.headers['last-modified']

        if req_status == "WAITING_FOR_UPDATE":
            if current_update_date != req_update_date:
                result = 'SUCCEEDED'
                status = 'UPDATED'
            else:
                result = 'WAITING'
                status = 'WAITING_FOR_UPDATE'
            return result, status, current_update_date

        if req_status == "WAITING_FOR_CREATE":
            result = 'SUCCEEDED'
            status = 'FILE_CREATED'
```

```
            return result, status, current_update_date

        result = 'WAITING'
        status = 'WAITING_FOR_UPDATE'
        return result, status, current_update_date

    except oci.exceptions.ServiceError as e:
        if e.status == 404:
            result = 'WAITING'
            status = 'WAITING_FOR_CREATE'
            update_date = ''
            return result, status, update_date
        else:
            raise
```

20.10　PATH環境変数定義ファイル

ファイル名

mjspath.txt

使用用途

指定されたジョブファイルを検索するために必要なPATH環境変数を定義するファイルです。

この定義ファイルで定義したPATH環境変数はジョブ実行時にも引き継がれます。

格納場所

Windows	Systemwalkerインストールディレクトリ¥MpWalker.JM¥mpmjessv¥user 複数サブシステム運用を行っている場合 Systemwalkerインストールディレクトリ¥MpWalker.JM¥mpmjessv¥mjes*n*¥user (*n*：サブシステム番号)
Solaris Linux	/etc/mjes 複数サブシステム運用を行っている場合 /etc/mjes/mjes*n* (*n*：サブシステム番号)

ファイル形式【Windows】

PATH=<パス1>;<パス2>;<パス3>・・・

- 1カラム目から“PATH=”で始まる行が複数あった場合はすべてを連結します。形式が不正な行は無視されます。
- コメントを記述する場合は1カラム目に“#”を使用します。“#”以降、その1行がコメントとして扱われます。
- 1行は4096バイト以内で記述してください。4096バイトを超えた場合、1行全体が無視されます。
- “PATH=”に続くパスの先頭および最後尾に半角スペース、タブがあった場合は無視されます。
- 複数行に分けて記述することもできます。

```
PATH=<パス1>;<パス2><改行>
PATH=<パス3>;<パス4>;<パス5><改行>
```

上記の場合、上から順に以下のように解釈されます。

```
PATH=<パス1>;<パス2>;<パス3>;<パス4>;<パス5>
```

パラメタ【Windows】

<パス>

ジョブファイルが存在するパスを記述します。複数指定する場合は“;”(セミコロン)で区切ります。指定した書式の有効性はOSの仕様に準じます。

ファイル形式【Solaris】【Linux】

```
PATH=<パス1>:<パス2>:<パス3>・・・
```

- 1カラム目から“PATH=”で始まる行が複数あった場合はすべてを連結します。形式が不正な行は無視されます。
- コメントを記述する場合は1カラム目に“#”を使用します。“#”以降、その1行がコメントとして扱われます。
- 1行は4096バイト以内で記述してください。4096バイトを超えた場合、1行全体が無視されます。
- “PATH=”に続くパスの先頭および最後尾に半角スペース、タブがあった場合は無視されます。
- 複数行に分けて記述することもできます。

```
PATH=<パス1>:<パス2><改行>
PATH=<パス3>:<パス4>:<パス5><改行>
```

上記の場合、上から順に以下のように解釈されます。

```
PATH=<パス1>:<パス2>:<パス3>:<パス4>:<パス5>
```

パラメタ【Solaris】【Linux】

<パス>

ジョブファイルが存在するパスを記述します。複数指定する場合は“:”(コロン)で区切ります。

指定した書式の有効性はOSの仕様に準じます。

文字コード

定義ファイルの文字コードは、サーバと同じ文字コードにします。
なお、Windows版の場合は、SJISコードで格納してください。

注意事項

- 旧バージョン互換(旧・標準)(NORMAL)属性【UNIX版】はPATH環境変数定義ファイルの対象外です。
- PATH環境変数の値の長さが、最終的に4096バイトを超えた場合、4096バイトを超えた分は切り捨てられます。切り捨てた旨のエラーメッセージは出力されません。
- 定義ミスがあった場合でも、エラーメッセージは出力されません。定義ファイルの指定値をPATH環境変数に追加して設定します。その後の動作は、OSの仕様に準じます。
- 定義は即時有効になります。運用中の変更は注意してください。
- ネットワークジョブは、実行サーバ側のPATH環境変数定義ファイルが有効となります。

使用例

【Windows】

```
# Schedule Server
```

```
PATH=c:¥job1¥jobfile;c:¥job2¥jobfile
PATH=c:¥job3¥jobfile
```

上記のように定義した場合、以下のPATH環境変数でジョブファイルが検索されます。

```
c:¥job1¥jobfile;c:¥job2¥jobfile;c:¥job3¥jobfile
```

【Solaris】【Linux】

```
# Schedule Server
PATH=/job1/jobfile:/job2/jobfile
PATH=/job3/jobfile
```

上記のように定義した場合、以下のPATH環境変数でジョブファイルが検索されます。

```
/job1/jobfile:/job2/jobfile:/job3/jobfile
```

第21章　イベントログ【Windows版】

本章では、Systemwalker Operation Managerが提供する各機能が出力するイベントログについて説明します。

21.1　イベントログの内容

本節では、イベントログに出力されるコンポーネントごとのソース名およびイベントの内容について説明します。
イベントログに出力されたメッセージの意味については、“Systemwalker Operation Manager メッセージ集”を参照してください。

ソース名

ソース名は、コンポーネントごとに異なり、利用者はソース名によって、どのコンポーネントが出力したイベントなのか識別することができます。

イベントの内容

各コンポーネントが出力するイベントの内容について、コンポーネント名とそれぞれのソース名、分類およびイベントの内容を対応させて、以下の表に示します。

コンポーネント名	ソース名	分類	説明
カレンダ	MpJmCal	クライアント	クライアントからの要求処理中に出力するメッセージです。
		サービス	サービスの処理中に出力するメッセージです。
		API	APIの処理中に出力するメッセージです。
電源制御／サービス・アプリケーション起動	MpAosfP	電源制御	電源制御サービスが電源制御処理をするときに、出力するメッセージです。
		サービス制御	サービス・アプリケーション起動サービスが出力する、サービス、アプリケーションの制御などについてのメッセージです。
		MS-Mail送信サーバ	自動アクションのMS-Mailを送信する場合に必要となる、MS-Mail送信サーバが出力するメッセージです。
ジョブスケジューラ	MpJobsch (注1)	サービス	ジョブスケジューラサービスが出力する情報やエラーなどのメッセージです。
		ジョブネット	ジョブネットの実行状況です。(注2)
		コマンド	jobschmsgeventコマンドが出力するイベントです。
ジョブ実行制御	MpMjes (注1)	サービス	ジョブ実行制御サービスが出力する情報やエラーなどのメッセージです。
		初期化文解析	初期化ファイル内に記述されている初期化文の解析によって検出された、初期化文の誤りやエラーのメッセージです。
		ジョブ	ジョブの警告メッセージです。
イベント監視	MpAosfB	—	イベント監視およびアクション管理が出力するメッセージです。

コンポーネント名	ソース名	分類	説明
／アクション管理			
業務連携	mpnjsosv	—	業務連携サービスが出力する情報やエラーなどのメッセージです。
Systemwalker Operation Manager 基盤	MpJmSrv	サービス	Systemwalker Operation Manager 基盤サービスが出力する情報やエラーなどのメッセージです。
ACLマネージャ	ACL Manager	サービス	ACLマネージャサービスが出力する情報やエラーなどのメッセージです。
監査ログ	MpAuditLogO	—	監査ログが出力する情報やエラーなどのメッセージです。
プロセス監視	MpPmonO	サービス	プロセス監視が出力する情報やエラーなどのメッセージです。
運用環境保守	Mpcmtoolo	—	運用環境保守ウィザード、バックアップ/リストアコマンドなどが出力する情報やエラーなどのメッセージです。
マスタスケジュール管理機能	MpStem (注1)	コマンド	マスタスケジュール管理機能が出力する情報やエラーなどのメッセージです。【EE/GEE】
Webコンソール/Web API	AHS F3FMpcmi-STFJMJAVAEE F3FMglassfish5-STFJMJAVAEE	—	Webコンソール/Web APIが出力する情報やエラーなどのメッセージです。

注1)

複数サブシステム運用を行っている場合、各ソース名にサブシステム番号(1～9)が付加されます。なお、サブシステム番号が0の場合、番号は付加されません。例：MpJobsch2, MpMjes8 など

注2)

実行状況のイベントを出力するかどうかは、[イベント出力の詳細設定]ウィンドウで設定できます。

第22章　ジョブの終了コード

本章では、Systemwalker Operation Managerが出力するジョブの終了コードについて説明します。

22.1　ジョブスケジューラの終了コード

ジョブの終了コードのうち、201～256までの値はジョブスケジューラで設定されています。ジョブスケジューラが返すジョブの終了コードとその意味を以下に示します。

ポイント

201～256以外の終了コード、および201～256内でも未使用の終了コードはOperation Managerが設定した値ではなく、ジョブ自身、すなわちジョブのコマンド欄に登録したコマンド、バッチファイル、シェルスクリプトなどが出力した値です。コマンド、バッチファイル、シェルスクリプトなど、ジョブ自身を調査してください。

また、ジョブとして登録したコマンド、バッチファイル、シェルスクリプトなどの終了コードが201～256とならないように考慮してください。ジョブ自身の復帰値として201～256が使用された場合、終了コードがジョブ自身で設定された値か、ジョブスケジューラが設定した値かの区別がつきにくくなります。なお、ジョブ自身の復帰値が255以上の場合は255として扱われます。

ジョブスケジューラが設定するジョブの終了コードとその意味

【Windows版】

終了コード	意 味
201～231	ジョブが異常終了しました。
232～238	未使用(ジョブスケジューラで予約)。
239	ジョブを実行中にジョブスケジューラサービスが停止しました。(注1)
240～241	未使用(ジョブスケジューラで予約)。
242	ジョブネットの複写起動に失敗しました。
243～244	(ジョブ実行制御に依頼した)ジョブの実行に失敗しました。
245	可変パラメタ、ジョブ定義変数、またはジョブネット変数の値に使用不可文字が含まれています。 または、ジョブの標準出力ファイル名/標準エラー出力ファイル名がフルパスで指定されていません。
246	可変パラメタ、ジョブ定義変数、またはジョブネット変数を置き換えた結果、文字列長の上限を超えました。 または、ジョブネット変数が、保持できる上限数640個を超えました。
247	ジョブ定義変数情報またはジョブネット変数の読み込みまたは書き込みに失敗しました。
248	OR条件の待ち合わせで、複数の先行ジョブのうち他のジョブが正常終了して後続ジョブが起動されたため、条件に該当せずにジョブが終了しました。
249	OR条件の待ち合わせで、複数の先行ジョブのうち他のジョブが異常終了し、条件判定ができなくなったためジョブが終了しました。
250	指定されたジョブが存在しません。
251～254	未使用(ジョブスケジューラで予約)。
255	ジョブの終了コードが 256以上です。(注3)
256	ジョブが強制終了されました。

【UNIX版】

終了コード	意 味
201～231	[ジョブ実行制御属性の場合] ジョブが異常終了しました。 [旧バージョン互換(旧・標準)属性の場合] ジョブがシグナルで終了しました。終了コードは、“200+シグナル番号”です。ただし、終了コード201、213、218、224、226および227はジョブスケジューラに無視されます。
232～237	未使用（ジョブスケジューラで予約）。
238	ジョブネット管理者の権限設定(initgroupsシステムコール)が失敗しました。
239	ジョブを実行中にジョブスケジューラデーモンが停止しました。(注1)
240～241	未使用（ジョブスケジューラで予約）。
242	ジョブネットの複写起動に失敗しました。
243～244	ジョブの実行に失敗しました。
245	可変パラメタ、ジョブ定義変数、またはジョブネット変数の値に使用不可文字が含まれています。 または、ジョブの標準出力ファイル名/標準エラー出力ファイル名がフルパスで指定されていません。
246	可変パラメタ、ジョブ定義変数、またはジョブネット変数を置き換えた結果、文字列長の上限を超えました。 または、ジョブネット変数が、保持できる上限数640個を超えました。
247	ジョブ定義変数情報またはジョブネット変数の読み込みまたは書き込みに失敗しました。
248	OR条件の待ち合わせで、複数の先行ジョブのうち他のジョブが正常終了して後続ジョブが起動されたため、条件に該当せずにジョブが終了しました。
249	OR条件の待ち合わせで、複数の先行ジョブのうち他のジョブが異常終了し、条件判定ができなくなったためジョブが終了しました。
250	ジョブネット管理者の情報取得(getpwnamシステムコール)が失敗しました。なお、旧バージョン互換(旧・標準)属性以外の時に、指定されたジョブが存在しなかった場合があります。(注2)
251	ジョブネット管理者の権限設定(setgidシステムコール)が失敗しました。
252	ジョブネット管理者の権限設定(setuidシステムコール)が失敗しました。
253	ジョブに指定されたディレクトリへの移動(chdirシステムコール)が失敗しました。
254	ジョブの起動(execvpシステムコール)が失敗しました。
255	ジョブの起動(forkシステムコール)が失敗しました。 旧バージョン互換(旧・標準)属性以外の場合、ジョブの終了コードが 256 以上です。(注3)
256	ジョブが強制終了されました。

注1)

ジョブの実行中にバックアップを行い、その情報をリストアした場合、実行中だったジョブは終了コード239の異常終了となります。

注2)

旧バージョン互換(旧・標準)属性以外の時にジョブが存在しなかった場合は、ジョブの前回履歴シートにエラーメッセージが出力されます。

注3)

実際の終了コードが 256以上の場合は、ジョブの履歴表示ウィンドウで以下のように表示されます。

```
・・・ended code=255.(実際の終了コード)
```

22.2 ジョブ実行制御の終了コード

ジョブ実行制御が出力するジョブの終了コードには、ジョブの種類によって、以下に示す値が設定されます。

ジョブの種類	ジョブの終了コード
バッチファイル【Windows版】/シェルスクリプト【UNIX版】	バッチファイル【Windows版】/シェルスクリプト【UNIX版】の復帰コード
プログラム	プログラムの復帰コード
業務手続き言語(JCL)	各ステップの最大値/最終ステップの終了コード

なお、ジョブの実行が中断された場合、ジョブの終了コードには、特別な値が設定されます。ジョブが中断した原因、および中断時に設定されるジョブの終了コードを以下に示します。

【Windows版/UNIX版】

終了コード	ジョブが中断した原因
0x40000008	・ JCLでのネットワークジョブ実行において、ネットワークジョブ制御コマンド(注1)の指定形式に誤りがありました。または、ネットワークジョブにおいて、ジョブ実行制御の内部処理でエラーが発生しました。 ・ 分散実行キューに投入されたジョブが、実行サーバ(スケジュールサーバとは異なるリモートのホスト)で起動された場合、以下のいずれかのエラーが発生しました。 － 入力ファイルが存在しない － 出力ファイルが存在しない ・ ネットワークジョブの投入先がWindows版サーバの場合、デスクトップヒープの枯渇が発生している可能性があります(注2)。
0x40000012	JCLでのネットワークジョブ実行において、ジョブステップに関する制御文(注3)の指定形式に誤りがありました。
0x40000016	・ ネットワークジョブにおいて、リモートサーバ上でのジョブ投入処理でエラーとなりました。 ・ 分散実行キューに依頼ホストを指定したジョブが投入されたため、実行依頼を拒否しました。 ・ スケジュールサーバ、または、実行サーバのディスク容量が不足しています。
0x40000020	ネットワークジョブの実行時に、スケジュールサーバから実行依頼サーバへの依頼において下位バージョンではサポートされていない機能を使用しています。そのため、拡張機能をサポートしていない下位バージョンのサーバに接続できないエラーが発生しました。
0x40000024	ネットワークジョブにおいて、リモートサーバ上でのジョブ投入処理でエラーとなりました。ジョブの実行ユーザは実行ユーザ制限リストによって制限されています。

終了コード	ジョブが中断した原因
0x40000132	・ ネットワークジョブにおいて、回線エラー、相手サーバダウンなど、ジョブの続行が不可能になりました。 ・ 分散実行キューに投入されたジョブが、実行サーバ(スケジュールサーバとは異なるリモートのホスト)で起動された場合、以下のいずれかのエラーが発生しました。 － 入出力ファイルの転送処理中にネットワーク系のエラーが発生 － 入出力ファイルの転送処理中にシステムエラーが発生 ・ スケジュールサーバ、または、実行サーバのディスク容量が不足しています。
0x40000222	画面上からの操作、qdelコマンド、ジョブ削除APIのいずれかにより、ジョブが強制終了されました。
0x40000322	打ち切り時間到来(ジョブ実行経過時間の到来)により、ジョブが強制終了されました。
0x40000806	JCLでのジョブ実行において、ジョブステップに指定された実行ファイルが存在しません。
0x40000913	【Windows版】 ・ ジョブ所有者の権限でジョブ実行する機能が有効な場合に、ジョブ実行時のユーザ認証処理でエラーとなりました(注4)。 ・ 分散実行ジョブ、ネットワークジョブにおいてリモートサーバ上での認証処理でエラーとなりました。 【UNIX版】 ・ 実行サーバがWindows版のネットワークジョブにおいて、実行サーバ側でジョブ所有者の権限でジョブを実行する機能が指定されているため、ジョブが異常終了しました。 ・ 分散実行ジョブ、ネットワークジョブにおいてリモートサーバ上での認証処理でエラーとなりました。
0x40000999	ジョブ実行制御の処理において、ジョブを続行することができないエラーが起きました。

注1)

以下のコマンドになります。

－ connect

－ send

－ execute

－ receive

－ disconnect

注2)

Windows Server 2003以降の場合は、OS起動後最初のデスクトップヒープ不足が発生した時点で、以下のイベントがイベントログに出力されます。

種類 : 警告
ソース : Win32k
イベント ID : 243
説明 : デスクトップ ヒープの割り当てに失敗しました。

注3)

以下の制御文になります。

- file
- exec

注4)

以下の原因が考えられます。

- ジョブを所有者の権限で実行するために必要な設定がされていない。

  必要な設定については、“Systemwalker Operation Manager 解説書”を参照してください。
- ドメインコントローラ(PDC、BDC)が停止または再起動された。
- Windowsインターネットネームサービス(WINS)が正常に動作していない。

【Windows版】

終了コード	ジョブが中断した原因
0xC0000005	ユーザアプリケーションが不当なメモリにアクセスしました。
0xC0000094	ユーザアプリケーションにおいて0除算が行われました。
0xC0000135	ユーザアプリケーションの実行に必要なDLLが見つかりません(必要なDLLのパスがPATH環境変数に存在しません)。
0xC0000142	ユーザアプリケーションの実行に必要なDLLの初期化処理に失敗しました。デスクトップヒープの枯渇が発生している可能性があります(注1)。

注1)

Windows Server 2003以降の場合は、OS起動後最初のデスクトップヒープ不足が発生した時点で、以下のイベントがイベントログに出力されます。

```
種類 : 警告
ソース : Win32k
イベント ID : 243
説明 : デスクトップ ヒープの割り当てに失敗しました。
```

オンライン業務の実行が中断された場合は、以下に示すジョブの終了コードが設定されます。

終了コード	意 味
0	オンライン業務は正常に終了しました。
1	Interstageが導入されていないため、オンライン業務の起動に失敗しました。
2	オンライン業務起動環境の初期化処理中に、Interstageにおいてエラーが発生し、オンライン業務の起動に失敗しました。
3	オンライン業務の起動処理中に、Interstageにおいてエラーが発生し、オンライン業務の起動に失敗しました。
4	オンライン業務の監視処理中に、Interstageにおいてエラーが発生し、オンライン業務の監視を中断しました。
5	オンライン業務の停止処理中に、Interstageにおいてエラーが発生し、オンライン業務の停止に失敗しました。
6	オンライン業務は異常終了しました。
256	オンライン業務が起動される前に、強制停止されました。
0x40000132	・ ネットワークジョブにおいて、回線エラー、相手サーバダウンなど、ジョブの続行が不可能になりました。

終了コード	意 味
	・ スケジュールサーバ、または、実行サーバのディスク容量が不足しています。
0x40000322	実行経過時間制限値を超えたか、ジョブ実行中に打ち切り時刻を迎えたため、オンライン業務は強制停止されました。
0x40000999	システムエラーが発生し、オンライン業務の起動に失敗しました。

付録A　本書の表記、商標について

A.1　略称表記について

- 以下の製品すべてを示す場合は、“Windows Server 2025”と表記します。
  - Microsoft(R) Windows Server(R) 2025 Datacenter(x64)
  - Microsoft(R) Windows Server(R) 2025 Standard(x64)
- 以下の製品すべてを示す場合は、“Windows Server 2022”と表記します。
  - Microsoft(R) Windows Server(R) 2022 Datacenter(x64)
  - Microsoft(R) Windows Server(R) 2022 Standard(x64)
- 以下の製品すべてを示す場合は、“Windows Server 2019”と表記します。
  - Microsoft(R) Windows Server(R) 2019 Datacenter(x64)
  - Microsoft(R) Windows Server(R) 2019 Standard(x64)
- 以下の製品すべてを示す場合は、“Windows Server 2016”と表記します。
  - Microsoft(R) Windows Server(R) 2016 Datacenter(x64)
  - Microsoft(R) Windows Server(R) 2016 Standard(x64)
- 以下の製品すべてを示す場合は、“Server Core”と表記します。
  - Microsoft(R) Windows Server(R) 2025 Datacenter Server Core
  - Microsoft(R) Windows Server(R) 2025 Standard Server Core
  - Microsoft(R) Windows Server(R) 2022 Datacenter Server Core
  - Microsoft(R) Windows Server(R) 2022 Standard Server Core
  - Microsoft(R) Windows Server(R) 2019 Datacenter Server Core
  - Microsoft(R) Windows Server(R) 2019 Standard Server Core
  - Microsoft(R) Windows Server(R) 2016 Datacenter Server Core
  - Microsoft(R) Windows Server(R) 2016 Standard Server Core
- 以下の製品すべてを示す場合は、“Windows(R) 11”または“Windows 11”と表記します。
  - Windows(R) 11 Home(x64)
  - Windows(R) 11 Pro(x64)
  - Windows(R) 11 Enterprise(x64)
- Windows Internet Explorer(R)を示す場合は、“Internet Explorer”と表記します。
- 以下のOS上で動作するSystemwalker Operation Managerを“Windows版 Systemwalker Operation Manager”または“Windows版”と表記します。
  - Windows
  - Itanium以外の64ビットアーキテクチャに対応したWindows
- 32ビットアーキテクチャに対応したWindows上で動作するSystemwalker Operation Managerの固有記事を“Windows x86版”と表記します。
- Itaniumに対応したWindows上で動作するSystemwalker Operation Managerの固有記事を“Windows for Itanium版”と表記します。

- Itanium以外の64ビットアーキテクチャに対応したWindows上で動作するSystemwalker Operation Managerの固有記事を“Windows x64版”と表記します。
- Server Core、Windows Server 2025、Windows Server 2022、Windows Server 2019およびWindows Server 2016を、“Windowsサーバ”と表記している箇所があります。
- Oracle Solarisは、Solaris、Solaris Operating System、Solaris OSと記載することがあります。
- Solaris上で動作するSystemwalker Operation Managerを“Solaris版 Systemwalker Operation Manager”または“Solaris版”と表記します。
- 32ビットアーキテクチャに対応したSolaris上で動作するSystemwalker Operation Managerの固有記事を“Solaris 32bit版”と表記します。
- 64ビットアーキテクチャに対応したSolaris上で動作するSystemwalker Operation Managerの固有記事を“Solaris 64bit版”と表記します。
- UXP/DS上で動作するSystemwalker Operation Managerを“DS版 Systemwalker Operation Manager”または“DS版”と表記します。
- HP-UX上で動作するSystemwalker Operation Managerを“HP-UX版 Systemwalker Operation Manager”または“HP-UX版”と表記します。
- AIX上で動作するSystemwalker Operation Managerを“AIX版 Systemwalker Operation Manager”または“AIX版”と表記します。
- Itanium以外の64ビットアーキテクチャに対応したLinux上で動作するSystemwalker Operation Managerの記事を“Linux版”または“Linux x64版”と表記します。
- 32ビットアーキテクチャに対応したLinux上で動作するSystemwalker Operation Managerの固有記事を“Linux x86版”と表記します。
- Itaniumに対応したLinux上で動作するSystemwalker Operation Managerの固有記事を“Linux for Itanium版”と表記します。
- Solaris版、DS版、HP-UX版、AIX版、Linux版およびLinux for Itanium版のSystemwalker Operation Managerを包括して、“UNIX版 Systemwalker Operation Manager”または“UNIX版”と表記します。
- Solaris、HP-UX、AIX、Linux、およびUXP/DSを、“UNIXサーバ”と表記している箇所があります。
- Systemwalker Operation Manager Standard Editionを“SE版”と表記している箇所があります。
- Systemwalker Operation Manager Enterprise Editionを“EE版”と表記している箇所があります。
- Systemwalker Operation Manager Global Enterprise Editionを“GEE版”と表記している箇所があります。
- Standard EditionをSE、Enterprise EditionをEE、Global Enterprise EditionをGEEと表記している箇所があります。
- Windows Server フェールオーバー クラスターおよびWindows Server Failover Clusteringは、Microsoft(R) Fail Over Clusteringと記載することがあります。
- Arcserve(R) Backup for Windowsを、Arcserveと表記します。
- Windows NT(R)で標準に提供されているMicrosoft(R)-MailをMS-Mailと表記します。

A.2 商標について

Amazon Web Services、その他のAWS商標は、Amazon.com, Inc.またはその関連会社の商標です。

APC、PowerChuteは、シュナイダー・エレクトリック・アイティー・コーポレーションの登録商標です。

Arcserveのすべての製品名、サービス名、会社名およびロゴは、Arcserve (USA), LLC.またはその子会社の登録商標または商標です。

Google ChromeおよびChromiumは、Google LLCの登録商標です。

HP-UXは、米国およびその他の国におけるHewlett-Packard Companyの登録商標です。

IBM、IBMロゴ、AIX、AIX 5L、HACMP、Power、PowerHAは、世界の多くの国で登録された International Business Machines Corporationの商標です。

Intel、Itaniumは、アメリカ合衆国および / またはその他の国における Intel Corporation またはその子会社の商標です。

Linux(R)は米国およびその他の国におけるLinus Torvaldsの登録商標です。

MC/ServiceGuardは、米国Hewlett Packard Enterprise Development LPの米国およびその他の国における登録商標です。

Microsoft、Powershell、Excel、Windows、Windows Server、Azure、Edgeまたはその他のマイクロソフト製品の名称および製品名は、米国 Microsoft Corporation の、米国およびその他の国における登録商標または商標です。

Oracle(R)およびJavaは、オラクルおよびその関連会社の登録商標です。

Red Hatは、米国およびその他の国におけるRed Hat, Inc.およびその子会社の登録商標です。

SAP、SAP R/3、およびその他のSAPの製品やサービス、ならびにそれらの個々のロゴは、ドイツおよびその他の国におけるSAP SE(またはSAPの関連会社)の商標または登録商標です。

UNIXは、米国およびその他の国におけるオープン・グループの登録商標です。

VMwareおよびVMwareの製品名は、Broadcom Inc.の米国および各国での商標または登録商標です。

ショートメールは、株式会社NTTドコモの登録商標です。

そのほか、本マニュアルに記載されている会社名および製品名は、それぞれ各社の商標または登録商標です。

Microsoft Corporationのガイドラインに従って画面写真を使用しています。