

IBM Sterling Connect:Direct for UNIX

ユーザー・ガイド

Version 4.1



本書は、新しい版で明記されていない限り、IBM® Sterling Connect:Direct® for UNIX バージョン 4.1、また、それ以降のすべてのリリースおよびモディフィケーションに適用されます。

本書および本書で紹介する製品をご使用になる前に、124 ページの『特記事項』に記載されている情報をお読みください。

Licensed Materials - Property of IBM

IBM® Sterling Connect:Direct® for UNIX

© Copyright IBM Corp. 1999, 2011. All Rights Reserved.

US Government Users Restricted Rights - Use, duplication or disclosure restricted by GSA ADP

Schedule Contract with IBM Corp.

第 1 章 Sterling Connect:Direct for UNIX 7

サーバー・コンポーネント	7
Process マネージャー	7
コマンド・マネージャー	8
セッション・マネージャー	8
Sterling Connect:Direct for UNIX の概念	8
ユーザー認証	9
Process 再始動	9
アーカイブ統計ファイル	11
Process、シェル・スクリプト、および API プログラムのサンプル	11
Sterling Connect:Direct for UNIX ファイル	13
Sterling Connect:Direct for UNIX の構成ファイル	13
Sterling Connect:Direct for UNIX のディレクトリー構成	14
Sterling Connect:Direct for UNIX 文書	16
本書について	16
タスクの概要	16

第 2 章 Process の制御および監視 18

コマンド・ライン・インターフェース (CLI) の使用	18
CLI の起動	18
Job 制御の使用	21
CLI での history の使用	21
CLI の停止	22
Sterling Connect:Direct コマンドの使用におけるガイドライン	22
コマンド概要	22
パラメーターの省略形	23
プログラムおよびスクリプトの使用における制限	23
コマンド構文	24
汎用	24
リスト	24

Process の実行依頼.....	25
submit コマンドのパラメーター.....	25
submit コマンドのサンプル.....	31
毎週実行する Process の実行依頼.....	31
開始時刻が指定された Process の実行依頼.....	32
ファイル値が存在しない Process の実行依頼.....	32
Process の実行依頼およびトレースの有効化.....	32
Process パラメーターの変更.....	32
Process の削除.....	35
実行キューからの Process の削除.....	37
Sterling Connect:Direct の停止.....	39
TCQ 内の Process の表示.....	40
TCQ 上の Process ステータスの監視.....	43
Process の出力の決定.....	47
詳細な出力のサンプル.....	53
サマリー出力のサンプル.....	53
システム診断の実行.....	54

第 3 章 Process キューイング 58

伝送制御キュー (TCQ).....	58
Sterling Connect:Direct アクティビティのスケジューリング.....	59
TCQ を通した Process の進捗.....	60
実行キュー.....	60
待機キュー.....	61
タイマー・キュー.....	62
保留キュー.....	62

第 4 章 Sterling Connect:Direct for UNIX ユーティリティの使用 64

変換テーブルの作業.....	64
変換テーブルの作成および修正.....	65
例: 変換テーブルの作成.....	66
例: モデル変換テーブルの修正.....	67
ファイル転送操作時の変換の使用.....	67
診断.....	68
Sterling Connect:Direct メッセージへのアクセス.....	68
メッセージ・ファイルの内容.....	68
メッセージ・ファイル・レコード・フォーマットの理解.....	69
メッセージ・テキストの表示.....	69
スタンドアロン型バッチ圧縮ユーティリティを使用した、ファイルの事前圧縮/圧縮解除	
圧縮ユーティリティ.....	70
スタンドアロン型バッチ圧縮ユーティリティの使用における特別な考慮事項.....	71
スタンドアロン型バッチ圧縮ユーティリティの使用.....	71
例: テキスト・ファイルの事前圧縮.....	74
例: コード・ページ変換を使用したテキスト・ファイルの事前圧縮.....	74
例: バイナリー・ファイルの事前圧縮.....	75
例: テキスト・ファイルの圧縮解除.....	75
例: csdacomp コマンドのヘルプ.....	75

例: Copy ステップ時のリモート・ノードでのファイルの圧縮解除	76
例: 事前圧縮されたファイルの z/OS への送信および事前圧縮済みとしての保存	76
構成ファイルの検証	77
構成レポートの生成	78
基本インストールに関するレポート生成	78
Sterling Connect:Direct for SWIFTNet for UNIX に関するレポート生成	80
Sterling Connect:Direct for SWIFTNet for UNIX に関するレポート生成	81
第 5 章 カスタム・プログラムの作成	82
カスタム・プログラムのコンパイル	82
カスタム C プログラム	85
ndmapi_connect() または ndmapi_connect_c() を使用した、 Sterling Connect:Direct への接続の作成	86
ndmapi_disconnect() または ndmapi_disconnect_c() を使用した、接続の終了	87
ndmapi_recvresp() または ndmapi_recvresp_c() を使用した、応答の受信	87
ndmapi_sendcmd() または ndmapi_sendcmd_c() を使用した、 Sterling Connect:Direct へのコマンドの送信	91
カスタム C++ プログラムの作成	93
第 6 章 ユーザー出口の作成	99
ユーザー出口関数の理解	99
ユーザー出口関数	100
exit_child_init() または exit_child_init_c() を使用した通信の初期化	100
recv_exit_msg() または recv_exit_msg_c() を使用したメッセージの待機	101
send_exit_file() または send_exit_file_c() を使用したファイル記述子の引き渡し	102
send_exit_msg() または send_exit_msg_c() を使用した Sterling Connect:Direct へのメッセージ送信	103
ユーザー出口メッセージの理解	104
統計出口メッセージ	104
ファイル・オープン出口メッセージ	104
FILE_OPEN_OUTPUT_MSG	105
FILE_OPEN_OUTPUT_REPLY_MSG	105
FILE_OPEN_INPUT_MSG	105
FILE_OPEN_INPUT_REPLY_MSG	105
セキュリティー出口メッセージ	106
GENERATE_MSG	106
GENERATE_REPLY_MSG	106
VALIDATE_MSG	106
VALIDATE_REPLY_MSG	107
ユーザー出口停止メッセージ	107
copy 制御ブロック	107
出口ログ・ファイル	107
用語集	110

目次

索引 **116**

特記事項 **124**

商標.....126

Sterling Connect:Direct for UNIX

IBM® Sterling Connect:Direct® for UNIX は、複数の技術を関連付けて、ネットワーク化されたシステムおよびコンピューター間において、あらゆるタイプの情報を移動します。また、自動化、信頼性、効率的なリソース使用、アプリケーション統合、使いやすさといった特性を提供することにより、ハイパフォーマンスな転送を管理します。Sterling Connect:Direct for UNIX は、通信プロトコル、ハードウェア・プラットフォーム、およびオペレーティング・システムにおいて複数の選択肢を提供するとともに、メインフレーム・システム、ミッドレンジ・システム、デスクトップ・システム、および LAN ベースのワークステーション間での情報の移動を可能とする柔軟性も備えています。

Sterling Connect:Direct for UNIX は、クライアント/サーバー・アーキテクチャーに基づいています。Sterling Connect:Direct for UNIX のサーバー・コンポーネントは、ユーザー・インターフェース (API、CLI、IBM® Sterling Connect:Direct® Browser User Interface、IBM® Sterling Control Center) と通信することで、Sterling Connect:Direct for UNIX のステートメントおよびコマンドの実行依頼、実行、監視することができるようになります。

サーバー・コンポーネント

Sterling Connect:Direct for UNIX には、以下のサーバー・コンポーネントが搭載されています。

Process マネージャー

Process マネージャー (PMGR) は、Sterling Connect:Direct for UNIX サーバー環境を初期化するデーモンです。PMGR は、以下の機能を提供します。

- ◆ Sterling Connect:Direct for UNIX の初期化
- ◆ Sterling Connect:Direct for UNIX のクライアント API およびリモート・ノードからの接続要求の受け入れ
- ◆ API およびリモート・ノードと通信するための、コマンド・マネージャーおよびセッション・マネージャーの子 Process の作成
- ◆ 一元化された Sterling Connect:Direct for UNIX 機能が必要とされる場合の、コマンド・マネージャーおよびセッション・マネージャーからの要求の受け入れ
- ◆ Sterling Connect:Direct for UNIX の停止

注: PMGR に接続可能なすべてのコンピューターにおいて、エンド・ユーザー・アプリケーション (End User Application: EUA) を含めたあらゆるアプリケーションを実行することができます。

コマンド・マネージャー

正常に確立された API 接続のすべてに対して、コマンド・マネージャー (CMGR) が作成されます。PMGR が作成できるコマンド・マネージャーの数は、システムによって異なり、各 UNIX Process で使用可能なファイル記述子の数によって制限されます。UNIX オペレーティング・システムによってセットアップされるファイル記述子の数は、Sterling Connect:Direct for UNIX の運用に影響する可能性があります。許可される同時 Sterling Connect:Direct for UNIX セッションの数 (最大 999 個) を処理するのに十分な数のファイル記述子を定義する必要があります。

CMGR は、以下の機能を提供します。

- ◆ API によって送信されるコマンドの実行し、その結果を API に対して送信
- ◆ API と連動して、Sterling Connect:Direct for UNIX の認証手順を実行し、Sterling Connect:Direct for UNIX に対するアクセスを判断
- ◆ コマンド実行時の PMGR との通信

セッション・マネージャー

セッション・マネージャー (SMGR) は、リソースが利用可能であり、Process の実行準備ができていて、またはリモート・ノードがローカル・ノードとの接続を要求する場合に、PMGR によって作成および呼び出しされます。SMGR は、以下の機能を提供します。

- ◆ 必要な Sterling Connect:Direct for UNIX 作業の実行
- ◆ 1 次ノード (PNODE) として機能し、Process の実行を開始
- ◆ 2 次ノード (SNODE) として機能し、PNODE によって開始された Process に参加

ノードに実行依頼される Process を実行するために SMGR が作成されると、リモート・ノードへの接続が作成されます。ローカル Process を実行するために PMGR によって SMGR が開始されると、SMGR は全 Process が完了するまで、このセッション上で各 Process を実行します。

リモート・ノードが接続を開始したことによって SMGR が作成される場合、SMGR が接続を完了させます。リモート Process を実行するために、PMGR によって SMGR が開始される場合、SMGR は、リモート SMGR が保持する Process をすべて完了させるまで、リモート SMGR によって提供されるリモート Process ステップを実行します。

SMGR は、伝送制御キュー (TCQ) サービスおよびその他の一元化されたサービスの PMGR に依存します。TCQ の概要については、15 ページの『伝送制御キュー (TCQ)』を参照してください。

Sterling Connect:Direct for UNIX の概念

本セクションでは、Sterling Connect:Direct for UNIX システム・オペレーションの理解に役立つ、概念および定義についてご紹介します。Sterling Connect:Direct の一般的な概念については、「*IBM Sterling Connect:Direct for UNIX 製品概要 (IBM Sterling Connect:Direct for UNIX Product Overview)*」を参照してください。

ユーザー 認証

Sterling Connect:Direct for UNIX では、Sterling Connect:Direct for UNIX の特定のタスクの実行において、ローカルおよびリモートのユーザーを認証することができます。Sterling Connect:Direct for UNIX を使用するにあたって、各ユーザーは userfile.cfg と呼ばれるユーザー認証ファイル内に定義されたレコードを持つ必要があります。各ローカル・ユーザーは、ユーザー認証ファイル内にレコードを持つ必要があります。リモート・ユーザーは、プロキシー関係でローカル・ユーザー ID にマッピングされる可能性があります。

一般ユーザーが、Sterling Connect:Direct for UNIX を通して root アクセス権を得ることを防止する対策として、Sterling Connect:Direct for UNIX インストール時に、強固なアクセス制御 (SACL) ファイルと呼ばれる 2 つ目のアクセス・ファイルが作成されます。このファイルは sysacl.cfg と命名されます。sysacl.cfg ファイルで指定される **root:deny.access** パラメーターは、Sterling Connect:Direct for UNIX に対するアクセスを許可、拒否、または制限します。SACL ファイルが削除されている、または破損している場合には、全ユーザーに対する Sterling Connect:Direct for UNIX へのアクセスが拒否されます。

userfile.cfg および sysacl.cfg ファイルでのユーザー認証の指定に関する詳細については、「*Sterling Connect:Direct for UNIX 管理ガイド (Sterling Connect:Direct for UNIX Administration Guide)*」の『*アクセス情報ファイルの保守 (Maintaining Access Information Files)*』を参照してください。

Processの再始動

システム障害後の Process リカバリー用に複数の機能が提供されます。Process リカバリーは、システム障害発生後、できるだけ早急の実行を再開して、無駄なデータ送信を最小限に抑えることを目的としています。Process リカバリーを可能とするために、以下の Sterling Connect:Direct for UNIX 機能を利用することができます。

- ◆ **Process ステップ・リスタート:** Process の実行に伴い、ステップが TCQ に記録されます。何らかの理由によって Process が停止されると、その Process は TCQ 内で保留されます。実行を継続するために Process が解放されると、その Process は停止されたステップから自動的に開始します。
- ◆ **自動セッション再試行:** ネットワーク・マップ・ファイルのリモート・ノード情報レコード内に short-term および long-term の 2 つの接続再試行パラメーター・セットが定義されます。これらのパラメーター値を、リモート・ノード情報レコード内で指定しない場合は、ネットワーク・マップ・ファイルの local.node エントリーからデフォルト値が使用されます。short-term パラメーターは、即時の再試行を可能とします。long-term パラメーターは、すべての short-term の再試行が実行された後に使用されます。long-term の試行では、接続の問題を迅速に修正することができず、再試行を実行するまでにより長い時間を要することが想定されるため、接続再試行のオーバーヘッドが軽減されます。
- ◆ **チェックポイント・リスタート:** この機能は、copy ステートメントによって使用できます。

チェックポイント・リスタートは、**ckpt** パラメーターを通して、**copy** ステップ内で明示的に構成することができます。(次の URL から IBM Sterling Connect:Direct for UNIX Process の Web サイトを参照してください:

<http://www.sterlingcommerce.com/documentation/processes/processhome.html>)

チェックポイント・リスタートが **copy** ステップで構成されない場合は、**ckpt.interval** パラメー

ターを使用して、Initparms で構成することができます。(このパラメーターに関する詳細については、「Sterling Connect:Direct for UNIX 管理ガイド (Sterling Connect:Direct for UNIX Administration Guide)」の『初期化パラメーター・ファイルの保守 (Maintaining Configuration Files)』を参照してください。)

- ◆ Run Task リスタート: SNODE 上の run task ステップの実行時に Process が中断された場合、Sterling Connect:Direct for UNIX は SNODE の前の run task ステップと現行の run task ステップとの同期を図ります。同期は、以下のいずれかの方法で実行されます。
 - Process の再始動時に、SNODE がタスクを実行中の場合には、そのタスクの完了を待ってから、PNODE に対して task completion ステータスを返します。処理が継続されます。
 - Process が再始動される前に SNODE のタスクが完了している場合には、タスク結果を保存します。Process の再始動時に、SNODE は結果を報告し、処理が継続されます

同期に失敗した場合、Sterling Connect:Direct for UNIX は、run task ステップ内の restart パラメーターまたは初期化パラメーター・ファイルを読み込み、run task ステップを再度実行するかどうかを決定します。run task ステップの restart パラメーターは、初期化パラメーターの設定をオーバーライドします。

例えば、SNODE が Sterling Connect:Direct for UNIX のコールド・リスタートによって、run task ステップの結果を失った場合、Sterling Connect:Direct for UNIX は、restart パラメーター内で定義されている値を確認して、run task を再度実行するかどうかを決定します。

注: run task リスタートは、Sterling Connect:Direct for UNIX が接続ロード・バランサーの裏で稼働している場合には、異なった機能を行います。ロード・バランシング環境で Sterling Connect:Direct for UNIX を稼働する場合に知っておく必要のある考慮事項についての詳細な情報については、「IBM Sterling Connect:Direct for UNIX リリース・ノート (IBM Sterling Connect:Direct for UNIX Release Notes)」、「IBM Sterling Connect:Direct for UNIX 管理ガイド (IBM Sterling Connect:Direct for UNIX Administration Guide)」および「Sterling Connect:Direct for UNIX スタートアップ・ガイド (Sterling Connect:Direct for UNIX Getting Started Guide)」を参照してください。

- ◆ SNODE が Sterling Connect:Direct for UNIX ノードの場合の Process アクティビティーの中断: SNODE が Sterling Connect:Direct for UNIX ノードであり、PNODE が、Process アクティビティーの停止コマンドを発行したり、実行している Process を削除したりすることによってプロセス・アクティビティーを中断させた場合、または、リンクが失敗したり、転送時に I/O エラーが発生したりした場合には、その Process は、WS ステータスで待機キューに設定されます。

Process アクティビティーが継続されない場合、その Process を TCQ から手動で削除する必要があります。delete process コマンドにおけるコマンド構文およびパラメーター記述については、「Sterling Connect:Direct for UNIX ユーザー・ガイド (Sterling Connect:Direct for UNIX User Guide)」を参照してください。

注: SNODE から change process コマンドを発行して、Process アクティビティーを継続させることはできません。Process は、常にセッションの制御を行っている PNODE によってのみ再始動させることができます。

アーカイブ統計ファイル

Sterling Connect:Direct for UNIX は、統計ファイルをアーカイブしてパージするユーティリティを提供します。Sterling Connect:Direct for UNIX を構成する際に、初期化パラメーター・ファイルの統計レコードに **max.age** のパラメーターを設定することによって、統計ファイルをアーカイブするタイミングを特定します。**max.age** パラメーターは、統計ファイルをアーカイブの対象する際の要件となる、ファイルの古さを定義します。

毎日 1 回、statarch.sh と呼ばれるスクリプトが開始されます。このスクリプトは、**max.age** に一致する統計ファイルを特定した後、tar コマンドおよび compress コマンドを実行して、**max.age** パラメーターに一致する統計レコードのすべてをアーカイブした圧縮ファイルを作成します。統計ファイルは、アーカイブ後、パージされます。Linux コンピューター上でアーカイブされるファイルに関しては、アーカイブされる統計ファイルは gzip 形式で圧縮されるため、.gz の接尾辞が付加されます。その他すべての UNIX プラットフォーム上でアーカイブされるファイルは、compress 形式を使用して圧縮されていることを示す .Z の接尾辞が付加されます。

アーカイブ・ファイルは、統計ファイルおよび TCQ が格納されているディレクトリーに保存されます。statarch.sh のシェル・スクリプトは、ndm/bin ディレクトリーに置かれています。必要に応じて、スクリプトを修正し、使用する環境用にカスタマイズしてください。

アーカイブされた統計ファイルをリストアしたい場合には、**statrestore.sh** スクリプトを実行します。このスクリプトは、**uncompress** および **tar** コマンドを使用して、アーカイブ内のすべての統計ファイルをリストアします。この際、**statrestore** コマンドに対して 2 つの引数を入力する必要があります。1 つ目の引数は統計ファイルが格納されているディレクトリーのパスになります。2 つ目の引数は、リストアしたいだけの数のアーカイブ・ファイル名が続く、アーカイブ・ファイル名を特定します。以下は、**statrestore** コマンドのサンプルです。

```
qa160sol: ./statrestore.sh/export/home/users/cd4000/ndm/bin archive1
```

ファイルのリストア後、select statistics コマンドを使用することにより、統計レコードを確認することができます。

Process、シェル・スクリプト、および API プログラムのサンプル

Sterling Connect:Direct for UNIX は、*d_dir*/ndm/src 内で Process およびシェル・スクリプトのサンプルを提供します。*d_dir* は、Sterling Connect:Direct for UNIX ソフトウェアの宛先ディレクトリーを示しています。テキスト・エディターで同様のファイルを作成することができます。また、Process およびシェル・スクリプトのサンプルの作成手順に関しては、同じディレクトリー内の README ファイルに記載されています。

以下の表は、サンプル Process のファイル名を記載しています。必要に応じて Process を修正してください。

ファイル名	Process タイプ
cpunx.cd	copy
rtunx.cd	run task

rjunx.cd	run job
sbunx.cd	submit

以下の表は、サンプル・シェル・スクリプトのファイル名を記載しています。必要に応じて、シェル・スクリプトを修正してください。

ファイル名	シェル・スクリプト・タイプ
selstat.sh	統計の選択
send.sh	送信
recv.sh	受信
wildcard	複数ファイルの PDS への送信
statarch.sh	統計ファイルのアーカイブ
statrestore.sh	アーカイブされた統計ファイルのリストア
lcu.sh	ローカル接続ユーティリティー (LCU) ツールの起動
spadmin.sh	Secure+ 管理ツールの起動
spcli.sh	Secure+ CLI (SPCLI) の起動
spcust_sample1.sh	STS プロトコル用の Secure+ の構成
spcust_sample2.sh	STS プロトコル用の Secure+ の構成
spcust_sample3.sh	SSL または TLS プロトコルを使用するための Secure+ の構成

以下の表は、サンプル・プログラムの名前を記載しています。

プログラム名	説明
apicheck.c	このプログラムは、ファイルをリモート・システムにコピーする Process の実行依頼を行います。この例では、MAXDELAY が使用されます。これは、プログラムが、ファイルの転送が終わるまで実行を終了しないことを意味します。このモジュールのコンパイルには、標準の c コンパイラーが使用されます。
apicheck.C	「Sterling Connect:Direct for UNIX ユーザー・ガイド (Sterling Connect:Direct for UNIX User Guide)」に記載される C++ コンパイラーの 1 つを使用してコンパイルされることを除き、apicheck.c と同じ。
exit_skeleton.c	このプログラムは、Sterling Connect:Direct for UNIX と連携して機能する ユーザー出口プログラムのスケルトンです。これは、全 3 種のユーザー出口の使用について示します。
exit_skeleton.C	「Sterling Connect:Direct for UNIX ユーザー・ガイド (Sterling Connect:Direct for UNIX User Guide)」に記載される C++ コンパイラーの 1 つを使用してコンパイルされることを除き、exit_skeleton_c と同じ。
exit_sample.c	これは、セキュリティ出口が PassTicket 機能に近いコードを使用して示され

sdksample.C

このプログラムは、Sterling Connect:Direct for UNIX に対する SDK インターフェイスを使用して、さまざまなコマンドを実行します。

Sterling Connect:Direct for UNIX ファイル

このセクションでは、Sterling Connect:Direct for UNIX の構成ファイルを説明するとともに、ディレクトリー構成を例示し、インストールされる個別ファイルの一覧を記載します。

Sterling Connect:Direct for UNIX の構成ファイル

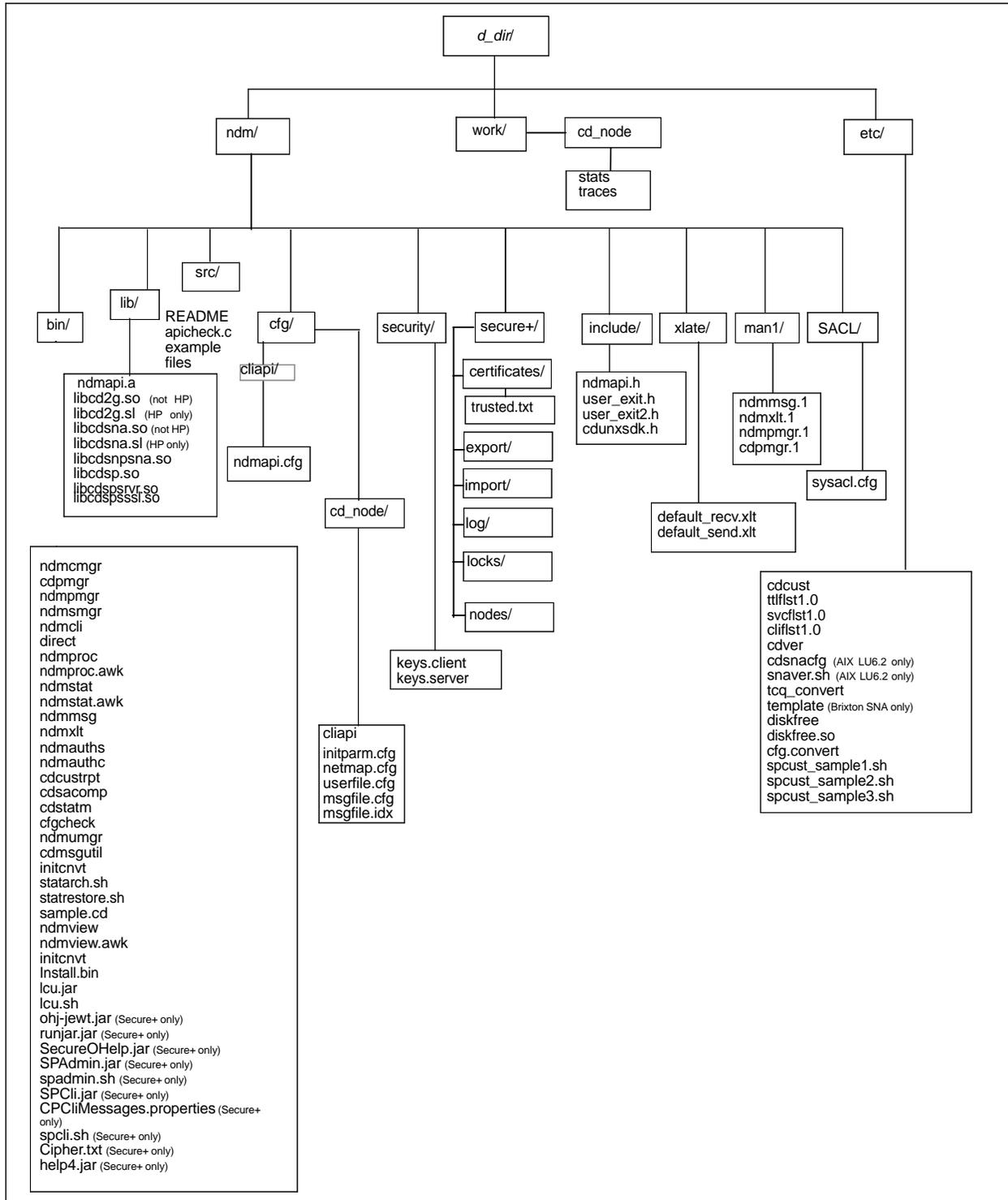
Sterling Connect:Direct for UNIX は、インストールおよびカスタマイズの実行時に、以下の構成ファイルを作成します。これらのファイルは、Sterling Connect:Direct for UNIX サーバーが正確に稼働するために必要となります。

構成ファイル	説明
初期化パラメーター・ファイル	起動時に使用する情報をサーバーに提供します。インストール時に、初期化パラメーター・ファイルに必要な設定を特定します。
ユーザー認証情報ファイル	ローカル・ユーザー情報およびリモート・ユーザー情報のレコード・タイプを格納しています。インストール時にこのファイルをカスタマイズして、リモート・ユーザー ID をローカル・ユーザー ID に対してマッピングし、ユーザー認証情報ファイルにリモート・ユーザー情報レコードを作成します。
強固なアクセス制御ファイル	Sterling Connect:Direct for UNIX のセキュリティを向上させて、root アクセス権における許可、拒否、または制限を行います。このファイルは、Sterling Connect:Direct for UNIX のインストール時に作成されます。このファイルが削除されている、または破損している場合には、全ユーザーに対する Sterling Connect:Direct for UNIX へのアクセスが拒否されます。
ネットワーク・マップ・ファイル	ローカル・ノードおよびネットワーク内のその他の Sterling Connect:Direct for UNIX について記載します。Sterling Connect:Direct for UNIX が通信する各ノードのリモート・ノード・レコードを定義することができます。
サーバー認証キー・ファイル	クライアントの API 接続要求の妥当性を検証します。妥当性が確認されたクライアントのみ、接続が許可されます。
クライアント構成ファイル	クライアントが Sterling Connect:Direct for UNIX への接続に使用するポートおよびホスト名を特定します。
クライアント認証キー・ファイル	Sterling Connect:Direct for UNIX クライアントが接続する先の Sterling Connect:Direct for UNIX サーバーを特定します。複数のサーバー・エントリーを保持することが可能です。

Sterling Connect:Direct for UNIX のディレクトリー構成

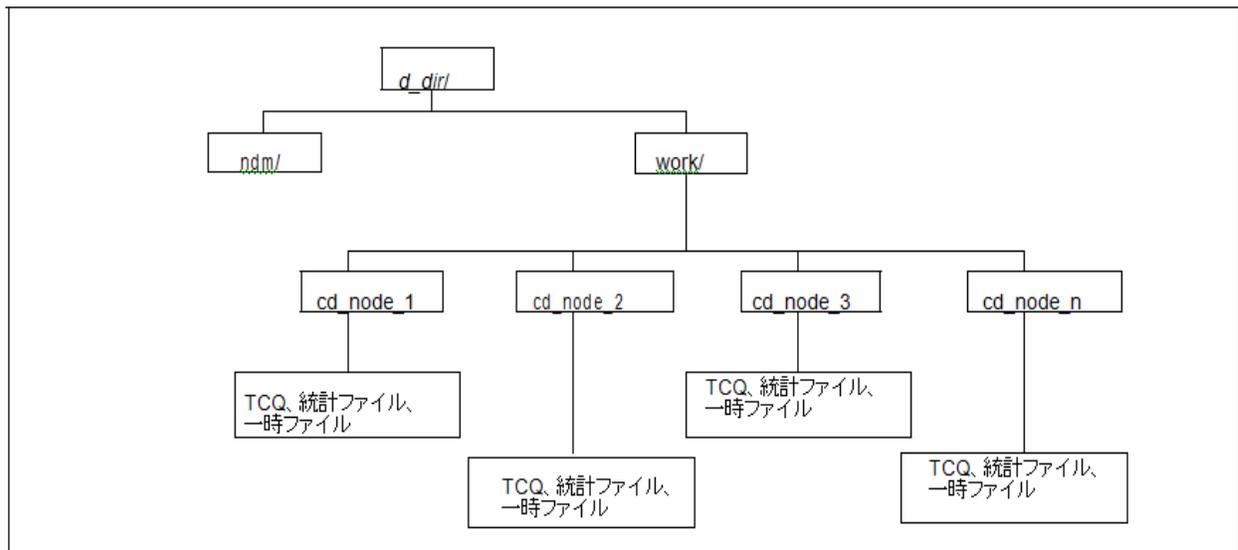
以下の図は、Sterling Connect:Direct for UNIX のディレクトリー構成を示しています。このディレクトリー・ツリーは、Sterling Connect:Direct for UNIX がインストールされる宛先ディレクトリーである `d_dir/` で始まっています。このディレクトリー構成は、同一のネットワーク上に存在、また、同一のコンピュータ上に存在する可能性のある複数のノードについて表示しています。このディレクトリー構成の体系化によって、`cdpmgr` や `ndmcmgr` といった Sterling Connect:Direct for UNIX のプログラムを共有することができるようになります。複数のノードが存在する場合には、各ノードに対し、構成ファイル用に独自の `d_dir/ndm/cfg/cd_node/` ディレクトリー構成が存在する必要があります。`cd_node` は、Sterling Connect:Direct for UNIX のノード名になります。

注: Secure+ ディレクトリーは、Sterling Connect:Direct for UNIX Secure Plus がインストールされる場合にのみ利用可能です。



注: 以下の図を参照して、ノードの作業ディレクトリーを確認してください。

各ノードに対して、`d_dir/work/cd_node` が作成されます。以下の図は、複数のノードにおける作業ディレクトリー、および各ノードに対して作成された作業ファイル (例: TCQ ファイル) を示しています。



Sterling Connect:Direct for UNIX 文書

本書について

『IBM Sterling Connect:Direct for UNIX ユーザー・ガイド (IBM Sterling Connect:Direct for UNIX User Guide)』は、Sterling Connect:Direct for UNIX をインストールするネットワーク運用スタッフおよびプログラマーを対象としています。本書によって、Sterling Connect:Direct for UNIX のインストールに必要な知識を得ることができます。

本書は、アプリケーション、ネットワーク、環境を含めた、UNIX オペレーティング・システムの知識があることを前提としています。LU6.2 接続においては、独立 LU6.2 接続をサポートするための SNA パッケージの構成における高度な能力が必要となります。

タスクの概要

以下の表は、Sterling Connect:Direct for UNIX のインストールにおけるタスクの実行に必要な情報の参照先について案内しています。

タスク	参照先
Understanding Sterling Connect:Direct の理解	第 1 章: <i>Sterling Connect:Direct for UNIX</i>
Sterling Connect:Direct for UNIX のインストール	第 2 章: <i>Sterling Connect:Direct for UNIX のインストール</i>
Sterling Connect:Direct for UNIX のカスタマイズ	第 2 章: <i>Sterling Connect:Direct for UNIX のインストール</i>

タスク	参照先
Sterling Connect:Direct for UNIX サーバーのセットアップ	第 2 章: <i>Sterling Connect:Direct for UNIX のインストール</i>
Sterling Connect:Direct for UNIX クライアントのセットアップ	第 2 章: <i>Sterling Connect:Direct for UNIX のインストール</i>
強固なアクセス制御ファイルの作成、セッション・マネージャー、Process マネージャー、コマンド・マネージャー、ユーザー・マネージャー・プログラム上のファイル許可の変更	第 2 章: <i>Sterling Connect:Direct for UNIX のインストール</i>
Sterling Connect:Direct for UNIX File Agent のインストール	第 2 章: <i>Sterling Connect:Direct for UNIX のインストール</i>
Sterling Connect:Direct for UNIX Secure Plus のインストール	第 2 章: <i>Sterling Connect:Direct for UNIX のインストール</i>
Sterling Connect:Direct for UNIX File Agent を使用したファイルの管理	第 3 章: <i>Sterling Connect:Direct File Agent を使用したファイルの管理</i>
Sterling Connect:Direct for UNIX のインストールおよびカスタマイズに必要な情報の記入	付録 A: <i>インストール用ワークシート</i>
LU6.2 接続の構成	付録 B: <i>SNA LU6.2 接続</i>
マニュアル・ページへのアクセスおよび表示	付録 C: <i>Sterling Connect:Direct マニュアル・ページ</i>

Process の制御および監視

この章では、コマンド・ライン・インターフェース (CLI) を使用して Process およびコマンドを実行依頼する手順について記載しています。

注: この章に記載されている手順の中には、Sterling Connect:Direct Browser User Interface を用いて実行可能なものもあります。Sterling Connect:Direct Browser User Interface に関する詳細については、『IBM Sterling Connect:Direct Browser User Interface ユーザー・ガイド (IBM Sterling Connect:Direct Browser User Interface User Guide)』を参照してください。

コマンドライン・インターフェース (CLI) の使用

コマンドライン・インターフェース (CLI) を使用することで、ネイティブのコマンド・ライン環境から Sterling Connect:Direct Process およびコマンドを実行依頼することが可能となります。このセクションでは、CLI 操作時に使用するコマンドについて説明します。

CLI の起動

CLI を起動するには、以下のステップを実行します。

1. NDMAPICFG 環境変数をまだ定義していない場合は、以下の適切なシェルのコマンドを入力します。d_dir には、Sterling:Connect:Direct のサブディレクトリーへのパスが入ります。

注: cfgCheck ユーティリティーを実行するためには、NDMAPICFG 環境変数を設定する必要があります。cfgCheck に関する詳細については、77 ページの『構成ファイルの検証』を参照してください。

- C シェル:

```
% setenv NDMAPICFG d_dir/ndm/cfg/cliapi/ndmapi.cfg
```

- Bourne シェルまたは Korn シェル:

```
$NDMAPICFG=d_dir/ndm/cfg/cliapi/ndmapi.cfg
$ export NDMAPICFG
```

2. 以下のコマンドを入力して、Sterling Connect:Direct CLI を起動します。必要に応じてオプションを入力します。

```
$ direct [-P string -s -t n -e nn -n name -p nnnnn -x -r -h -z]
```

コマンド・オプションおよびコマンド・エントリーのサンプルに関する説明については、以下の表を参照してください。

オプション	説明	値	コマンド・エントリーのサンプル
-P	<p>コマンドライン・プロンプトで使用するカスタム・STRINGを指定します。</p> <p>プロンプト・STRINGにスペースまたは特殊文字が含まれる場合には、それを単一引用符または二重引用符で囲んでください。</p> <p>プロンプト・STRINGは、ndmapi.cfg ファイル内で指定することも可能です。プロンプト・STRINGがコマンドラインおよび ndmapi.cfg ファイル内の両方で指定された場合、-P が優先されます。</p> <p>デフォルトのプロンプト (「Direct」) がオーバーライドされると、コマンドライン・プロンプトおよび Welcome バナーに新しいプロンプト・STRINGが表示されます。</p>	<p>テキスト・STRING</p> <p>最大 32 文字</p>	<pre>\$ direct -PNewPrompt \$ direct -P"Test CD on Medea"</pre>
-s	<p>標準出力を抑制します。コマンドの完了ステータスのみを確認する場合に、このオプションを使用してください。</p>	なし	<pre>\$ direct -s</pre>

オプション	説明	値	コマンド・エントリーのサンプル
-t n	CLI/API トレース・オプションを有効化します。レベル番号である n によって、トレース出力の詳細レベルを指定します。	1 2 4 以下のいずれかのレベル番号を指定します。 1 - 関数入口および関数出口を提供します。これはデフォルトです。 2 - 関数入口および関数出口、ならびに実行フロー内の主要ポイントにおける内部データ構造の値の表示といった、基本的な診断情報を提供します。 4 - 完全なトレースを有効にします。すべての診断情報が表示されます。	\$ direct -t 4
-e nn	CLI が自動的に終了するエラー・レベルを定義します。返されたエラー・コードが指定されているエラー・レベルを超えた場合に、CLI は自動的に終了します。 このコマンドはシェル・スクリプト内で使用します。 このパラメーターにより、指定されたレベルを超えるエラーを生成したコマンドの後に、不要なコマンドが実行されることを防ぎます。 CLI の終了時に、シェルによるテストが可能な UNIX 終了コードを返します。	0 4 8 16 エラー・レベル・コードの有効な値: 0 - 正常な完了を示します。 4 - 警告を示します。 8 - エラーを示します。 16 - 重大なエラーを示します。	\$ direct -e 16
-n name	Sterling Connect:Direct サーバー (PMGR) が稼動しているコンピューターのホスト名を指定します。 注: -p または -n を使って direct を呼び出すことにより、ndmapi.cfg ファイル内の設定がオーバーライドされます。	Sterling Connect:Direct ホスト名	\$ direct -n ホスト名
-p nnnnn	Sterling Connect:Direct ノードの通信ポート番号を指定します。 注: -p または -n を使って direct を呼び出すことにより、ndmapi.cfg ファイル内の設定がオーバーライドされます。	1024 ~ 65535。 フォーマットは nnnnn。	\$ direct -p 2222
-x	標準出力上に入力コマンドを表示します。スクリプトのデバッグを行う際に、このコマンドを使用してください。	なし	\$ direct -x

オプション	説明	値	コマンド・エントリーのサンプル
-r	Process 番号がユーザー作成のシェル・スクリプトで使用できるようにします。CLI には、特定のストリング <code>_CDPNUM_</code> 、スペース、Process 番号の順で表示されます。	none	<code>\$direct -r grep "_CDPNUM_"</code>
-h	D間違った Sterling Connect:Direct コマンドが入力された場合に、コマンドの使用に関する情報を表示します。	none	<code>\$ direct -h</code>
-z	プロンプトの後に改行文字を追加します。	none	<code>\$ direct -z</code>

ジョブ制御の使用

Sterling Connect:Direct では、ジョブ制御をサポートするシェルのフォアグラウンドとバックグラウンド間で CLI Process を切り替えることができます。この機能によって、CLI を終了して再起動することなく、保存された Process のテキストの編集、UNIX コマンドの実行、および Process エラーの解決を行うことができます。CLI Process を切り替えるには以下のコマンドを使用します。CLI Process を切り替えるには以下のコマンドを使用します。

- ◆ CLI Process を停止または中断するには、中断文字 (**Control-Z**) を押します。
- ◆ CLI Process をフォアグラウンドに移行するには、**fg** コマンドを実行します。

注: ジョブ制御の使用において問題がある場合、システム管理者に連絡して、使用すべき追加的 UNIX コマンドに関するアドバイスを受けてください。

CLI での history の使用

Sterling Connect:Direct では、UNIX で使用可能な history コマンドを使用することができます。history コマンドでは、コマンドの末尾にセミコロン (;) を必要としません。以下の表は、使用できる history コマンドを記載しています。

コマンド	説明
!!	直前のコマンドを 1 回繰り返します。
!#n	ヒストリー・バッファーに保存されるコマンド数を設定します。デフォルトのヒストリー・バッファー・サイズは 50 コマンド。
!n	ヒストリー・バッファー中のコマンド番号 <n> を繰り返します。
!<string>	<string> のストリングで始まるコマンドを繰り返します。
!?	ヒストリー・バッファーの内容を一覧表示します。

CLI の停止

プロンプトで、**Control-D** または **quit** を入力して CLI 操作を停止します。

Sterling Connect:Direct コマンドの使用におけるガイドライン

Sterling Connect:Direct コマンドの使用に関するガイドラインについては、本セクションの情報を参照してください。

コマンド概要

以下のコマンドを使用して、Sterling Connect:Direct Process の制御および監視を行います。

コマンド	省略形	説明
submit	sub	Process が実行できるようにします。
change process	cha pro	TCQ 内の非実行中の Process のステータスの変更や特性の修正を行います。
delete process	del pro	TCQ から非実行中の Process を削除します。
flush process	flush pro	TCQ から実行中の Process を削除します。
stop	stop	Sterling Connect:Direct for UNIX を停止して、制御をオペレーティング・システムに戻します。
select process	sel pro	実行中の Process および実行待機中の Process を監視します。検索基準および情報の表示形式を指定することができます。
select statistics	sel stat	統計ファイルから情報を取り出します。検索基準および情報の表示形式を指定することができます。
view process	view pro	ローカル・ノードが PNODE の場合の TCQ 内の Process を表示します。Process の表示に必要な情報は PNODE のみが保持していることから、view process は、ローカル・ノード上で実行されている Process しか表示できません。

注: CMGR は現在、Process ファイルのサイズを 60キロバイト までに制限しています。

パラメーターの省略形

以下の表は、共通に使用されるパラメーターの有効な省略形を記載しています。

パラメーター	省略形
detail	det
quit	q
recids	rec
release	rel
pname	pnam, pna
pnumber	pnum
sunday	sun
monday	mon
tuesday	tue
wednesday	wed
thursday	thu
friday	fri
saturday	sat
today	tod
tomorrow	tom

プログラムおよびスクリプトの使用における制限

システム管理者およびその他のネットワーク運用担当者は、ユーザーが run task および run job Process ステートメントを使用して実行できるスクリプトおよび UNIX コマンドを制限することができます。

システム管理者およびその他のネットワーク運用担当者は、Sterling Connect:Direct で使用できる機能において以下の制限を設定することができます。

- ◆ ファイルの送受信機能。ユーザーは、ファイルの送信のみ、または受信のみのいずれかの機能に制限される可能性があります。
- ◆ ファイルを送信または受信する、送信先または送信元のロケーション。特定のローカル・ノードまたはリモート・ノードに制限される可能性があります。

ユーザー ID における特定の制限については、システム管理者に確認してください。

コマンド構文

CLI プロンプトで入力されるコマンド、または `ndmapi_sendcmd()` 関数のコマンド・テキスト・パラメーターとして使用されるコマンドには、同一のコマンド構文を使用します。関数呼び出しに関する詳細については、82ページの『カスタム・プログラムの作成』を参照してください。コマンドを入力する際には、以下の規則が適用されます。

- ◆ パスワードまたはユーザー ID を選択する際には、Sterling Connect:Direct のキーワードを使用しないでください。
- ◆ ユーザー名およびファイル名は大文字と小文字が区別されることに留意します。
- ◆ 大文字のみ、小文字のみ、または大文字/小文字が混在する文字で個別のコマンド・キーワードを入力します。
- ◆ すべてのコマンドはセミコロン (;) で終了します。
- ◆ コマンドを入力する際には、コマンド名全体、最初の 3 文字、または特定のパラメーターに関してはその省略形を入力します。パラメーターの省略形のリストについては、23 ページの表を参照してください。
- ◆ ステートメントおよびパラメーターを省略しないでください。
- ◆ ファイル名、グループ名、ユーザー ID、およびパスワードは、可変長ストリングであり、長さの指定はありません。
- ◆ Sterling Connect:Direct ノード名は、1 ~ 16 文字の長さで設定されます。リモート・ノードを記述する netmap 内のレコード名は、通常、リモート Sterling Connect:Direct ノード名ですが、1 ~ 256 文字までの任意の長さで設定されることが可能です。リモート・ノード名を IP アドレスまたはホスト名、およびポート番号またはポート名として指定することもできます。

汎用

構文定義内のパラメーター値として **汎用** を指定した場合は、アスタリスク (*) および疑問符 (?) の文字を含むストリングを使用できます。これらの文字は、パラメーター値においてパターン・マッチングまたはワイルドカードを提供します。アスタリスクは、0 個以上の文字にマッチし、疑問符は、任意の単一文字にマッチします。以下のサンプルは、アスタリスクおよび疑問符の使用を示しています。

```
PNAME = A?PROD5*
```

上記のサンプルで指定された汎用 Process 名は、**A** の文字で始まり、その後の 2 番目の位置に任意の単一文字、3 番目から 7 番目の位置に **PROD5** のストリングが続く Process のすべてに一致する指定を示しています。アスタリスクは、8 番目の位置から始まる 0 個以上の文字に取って代わるものです。

リスト

パラメーター値が **(リスト)** の場合、パラメーター値のグループを括弧で囲み、それぞれの値をコマンドで区切ることによって、複数のパラメーター値を指定することができます。以下のコマンドは、リストを示しています。

```
(pnumber1, pnumber2, pnumber3)
```

Process の実行依頼

Process を実行できるようにし、指定されたファイルに含まれる Process ステートメントをソフトウェアが解釈できるようにするには、**submit** コマンドを使用します。

submit コマンドのパラメーター

submit コマンド内で指定されたパラメーターは、Process ステートメントで指定された同一のパラメーターをオーバーライドします。必須のパラメーターはありませんが、ファイル・パラメーターにファイル名を指定しない場合、Sterling Connect:Direct Process のテキストは、submit コマンドに従う必要があります。以下は、submit コマンドのパラメーターです。

パラメーター	説明	値
file	Process ファイル名。ファイル名には、Process のロケーションを示すパス名を含めることができます。 このパラメーターは、最初のパラメーターとなる必要があります。	ファイル名 (パス名を含むことも可)
class	Process を実行できるノード間のセッション。Process は、指定されたクラス、またはより上位の任意のセッション・クラスで実行できます。デフォルトのクラスは、初期化パラメーター・ファイル内の local.node レコードの sess.default パラメーターとして指定されています。	1 n 1 から最大同時ローカル・ノード接続数 (sess.pnode.max) までの数値。デフォルト値は 1。この値は、基本制御でのローカル・セッションの最大数を超えることはできません。
crc	CRC 検査が実行されるかどうかを決定します。このパラメーターは、初期化パラメーター、ネットワーク・マップ、および Process での設定をオーバーライドします 注: ユーザーは、ユーザー権限ファイル内で CRC 設定を変更するための権限が割り当てられている必要があります。	on off on - CRC 検査を有効化 off - CRC 検査を無効化。 デフォルトは off です。
hold	Process が保留キュー内に配置されるかどうかを決定します。 Process が retain=yes または retain=call で実行依頼される場合、Sterling Connect:Direct は、この hold パラメーターを無視します。	yes <u>no</u> call yes - change process コマンドによって解放されるまで、Process が HI ステータスで保留キューに配置されるように指定します。hold=yes で実行依頼された Process は、開始時刻が指定されている場合でも、保留キューに配置されます。 no - リソースが利用可能となり次第、Process が実行されるように指定します。これはデフォルトです。 call - リモート・ノードとローカル・ノード間の接続が確立するまで、Process が保留されるように指定します。接続すると、Process が解放されて実行されます。

パラメーター	説明	値
maxdelay	submit コマンドが、実行依頼された Process の実行が完了するまで待機する時間。このパラメーターは、シェル・スクリプトによってコマンドが実行される場合に活用できます。このパラメーターを指定すると、スクリプトは Process が完了するまで待機してから、実行を継続します。Process の戻りコードは、Bourne シェルまたは Korn シェルを使用している場合、\$? 変数に格納され、C シェルを使用している場合には、\$status 変数に格納されます。シェル・スクリプトはこれを使用して、Process の実行結果を検証することができます。maxdelay を指定しない場合には、遅延は発生しません。 時間間隔の有効期限が切れると、submit コマンドは、警告ステータス・コードおよびメッセージ ID を実行元の Process または CLI/API に返します。Process は、時間間隔の期限切れによる影響は受けずに、正常に実行されます。	unlimited hh:mm:ss 0 unlimited - Process が実行を完了するまで待機します。 hh:mm:ss - 指定された時間、分、秒を超えない範囲で待機します。 0 - Process の実行が完了するまで待機します。maxdelay=0 に設定した場合、maxdelay=unlimited を指定した場合と同じ結果を得ることになります。
newname	実行依頼された Process 内の名前をオーバーライドする新規の Process 名。	最大で 256 文字の名前
notify	Process 完了メッセージを受け取るユーザーの e メール。このパラメーターは、UNIX System V メール機能で使用できる rmail ユーティリティを使用して、完了メッセージを送信します。 注: Sterling Connect:Direct は、notify パラメーターに提供された e メール・アドレスまたはユーザー ID の検証は行いません。無効な e メール・アドレスや失敗した eメールの試行は、ローカル・メール機能の設定に基づいて処理されます。	ユーザー名@ホスト名 または ユーザー@ローカルホスト
pacct	PNODE に関する情報を含むストリング。このストリングを二重引用符の記号で囲んでください。	"pnode accounting data"。最大 256 文字。
pnodeid	PNODE でのセキュリティ・ユーザー ID およびパスワード。pnodeid サブパラメーターには、1 ~ 64 文字の英数字を含めることができます。	ID [, パスワード] ID - PNODE 上のユーザー ID を指定します。 パスワード - PNODE 上のユーザー・パスワードを指定します。 pnodeid を指定した場合、ID も指定する必要があります。最初に ID、最後にパスワードを指定します。
prty	伝送制御キュー (TCQ) 内の Process の優先順位。より高い優先順位の Process が、より低い優先順位の Process よりも先に実行されるように選択されます。prty 値は、伝送時の優先順位には影響を与えません。	1 ~ 15。15 は最も高い優先順位を示します。デフォルトは 10 です。

パラメーター	説明	値
retain	<p>Sterling Connect:Direct が TCQ 内で Process のコピーを保持するかどうかを決定します。Sterling Connect:Direct は、Process が保留キューに配置された際に、その Process に対して Process 番号を割り当てます。Process が実行されると、保持 Process に割り当てられた Process 番号が、1 増加します。例えば、保持キュー内で Process に 1445 の Process 番号が割り当てられていた場合、Process が実行されると、その Process 番号は 1446 になります。</p> <p>開始時刻を指定し、retain=yes と設定している場合、Process は、HR ステータスでタイマー・キュー内に保持され、適切な間隔で実行依頼されます。例えば、startt=(Monday,2:00) と指定した場合、Process は、毎週月曜日の 2:00 AM に実行されます。startt=(,1:00) と指定した場合、Process は、毎日 1:00 AM に実行されます。IBM Sterling Connect:Direct では、1 時間毎に Process を実行する方法は提供されていません。これを行うには、UNIX cron ユーティリティを使用する必要があります。</p> <p>開始時刻が指定されていない場合、Process を解放して実行するためには、change process コマンドを実行する必要があります。retain=initial を指定する場合、startt パラメーターのコーディングは行わないでください。</p>	<p>yes <u>no</u> initial</p> <p>yes - Process 実行後、システムが Process を HR ステータスで保留キュー内に保持するように指定します。</p> <p>no - Process 実行後、システムが Process を TCQ から削除するように指定します。これはデフォルトです。</p> <p>initial - Process マネージャー初期化の度に、Process を自動的に実行するために、システムが Process を HR ステータスで保留キュー内に保持するように指定します。</p>
sacct	<p>SNODE のアカウントング・データを指定します。submit ステートメント内でこの値を設定することにより、Process 内で指定されたすべてのアカウントング・データがオーバーライドされます。</p>	<p>“<i>snode accounting data</i>”。最大 256 文字。このストリングを二重引用符で囲んでください。</p>

パラメーター	説明	値
snode	2 次ノードの名前を特定します。この値を設定することにより、Process ステートメント内の snode 値がオーバーライドされます。submit コマンドまたは Process ステートメントのいずれかにおいて、 snode パラメーターが必要となります。	<p>名前 ホスト名 nnn.nnn.nnn.nnn または nnnn:nnnn:nnnn:nnnn:nnnn:nnnn:nnn n:nnnn[:ポート名 nnnnn]]</p> <p>名前 - リモート・ノードのノード名を指定します。2 次ノード名は、ネットワーク・マップ・ファイル内のエントリーに一致します。</p> <p>ホスト名 - リモート Sterling Connect:Direct ノードが稼動しているホスト・コンピューター名を指定します。</p> <p>nnn.nnn.nnn.nnn または nnnn:nnnn:nnnn:nnnn:nnnn:nnnn:nnn n:nnnn - リモート・ノードの IP アドレスを IPv4 または IPv6 形式で指定します。</p> <p>nnn.nnn.nnn.nnn (IPv4) または nnnn:nnnn:nnnn:nnnn:nnnn:nnnn:nnn n:nnnn (IPv6)。</p> <p>[:ポート名 nnnnn] - 通信ポートを特定します。このパラメーターは、ホスト名または IP アドレス・パラメーターを使用時のみ使用可能です。nnnn の値は、1,024 ~ 65,535 の 10 進数になります。</p> <p>IP アドレスおよびホスト名の指定に関する詳細については、『IBM Sterling Connect:Direct for UNIX 管理ガイド (IBM Sterling Connect:Direct for UNIX Administration Guide)』の付録『IP アドレス、ホスト名、およびポートの指定 (Specifying IP Addresses, Host Names, and Ports)』を参照してください</p>

パラメーター	説明	値
snodeid	<p>SNODE 上のセキュリティ・ユーザー ID およびセキュリティ・パスワードを指定します。pnodeid サブパラメーターには、1 文字以上の英数字を含めることができます。</p> <p>Sterling Connect:Direct により、Process に snodeid パラメーターが存在しない、または、snodeid パラメーターが定義されており、proxy.attempt 初期化パラメーターが y に設定されていることが判明した場合、snodeid パラメーターに指定されているすべてのパスワードは無視されます。プロキシ・ユーザー・レコードは、userfile.cfg 内のリモート・ユーザー・レコードであり、snodeid パラメーター上で指定されたユーザー名に対応します。プロキシ・ユーザー・レコードが存在しない場合、snodeid パラメーターには、userfile.cfg 内に対応するローカル・ユーザー・レコードを持つ UNIX ユーザーの有効なユーザー名およびパスワードが含まれている必要があります。</p> <p>proxy.attempt=n が定義されており、snodeid の定義がない場合は、Sterling Connect:Direct は、実行依頼元 ID およびノードを使用して、ユーザー認証情報ファイル内でリモート・ユーザー情報レコードを検索します。Sterling Connect:Direct が一致する情報を見つけられなかった場合、そのユーザーは、ファイルの送受信を行うことはできません。</p> <p>初期化パラメーター・ファイルの proxy.attempt パラメーターが y に設定されている場合、ユーザーは、snodeid パラメーターのパスワードを指定する必要はありません。</p> <p>この機能により、id サブパラメーターに、リモート・システム上のローカル・ユーザー ID への変換に使用されるダミー・ユーザー ID を含めることが可能となります。ダミー・ユーザー ID の使用によって、送信側および受信側のどちらも実際のユーザー ID を使用する必要がなくなることから、セキュリティが向上します。</p> <p>snodeid フィールドに、予約キーワードを使用することはできません。</p>	<p>ID [,パスワード[,新パスワード]]</p> <p>ID - SNODE 上のユーザー ID を指定します。</p> <p>パスワード - SNODE 上のユーザー・パスワードを指定します。ID を指定した場合、パスワードを指定する必要はありません。</p> <p>この機能により、ID パラメーターに、リモート・システム上のローカル ID への変換に使用されるダミー ID を含めることが可能となります。</p> <p>新パスワード - 新しいパスワード値を指定します。SNODE 上でユーザー ID および古いパスワードが正しい場合に、ユーザー・パスワードが新しい値に変更されるプラットフォームもあります (特定のプラットフォーム向けの文書を参照してください)。</p> <p>パスワードを指定した場合、ID も指定する必要があります。新パスワードを指定した場合、パスワードを指定する必要があります。ID、パスワード、新パスワードの順で、値を入力してください。</p>

パラメーター	説明	値
startt	<p>Process を開始する日付、曜日、時間を特定します。</p> <p>Sterling Connect:Direct は、Process を WS (Waiting for Start Time) ステータスでタイマー・キュー内に配置します。date、時刻、および time は、定位置パラメーターです。date または 時刻 を指定しない場合は、time の前にコンマを付ける必要があります。</p> <p>retain=initial を指定する場合、startt パラメーターのコーディングは行わないでください。</p>	<p>[日付 曜日] [,hh:mm:ss [am pm]]</p> <p>日付 - 日 (dd)、月 (mm)、年 (yy) を指定します。mm/dd/yyyy または mm-dd-yyyy としてコーディングできます。</p> <p>日付のみを指定した場合、時刻は、デフォルトで 00:00:00 (午前零時) となります。現在の日付がデフォルトです。</p> <p>曜日 - 週の曜日を指定します。値は、today、tomorrow、yesterday、monday、tuesday、wednesday、thursday、friday、saturday、および sunday です</p> <p>hh:mm:ss [am pm] - 時刻を、時間 (hh)、分 (mm)、秒 (ss) で指定します。時間は、12 時間形式と 24 時間形式のいずれでも指定できます。</p> <p>12 時間形式を使用する場合は、am または pm を指定する必要があります。デフォルトは 24 時間形式です。デフォルト値は 00:00:00 で、午前零時を表します。</p> <p>曜日の値のみを指定すると、時刻は、00:00:00 にデフォルト設定されます。これは、月曜日に Process を実行依頼する場合に、startt パラメーターとして monday のみを指定すると、Process は次の月曜日の午前零時まで実行されないことを意味します。</p>
&symbolic name1 &symbolic name 2 . . &symbolic name n	<p>値が割り当てられたシンボリック・パラメーターを指定します。シンボリック・パラメーターの検出時に、Process 内で値が代入されます。</p> <p>シンボリック・パラメーターの値は、それがキーワードである場合、または特殊文字を含んでいる場合には、二重引用符で囲む必要があります。Process でシンボリック名が解決 されるときに二重引用符を保持したい場合は、二重引用符で囲まれたストリングを単一引用符で囲みます。例えば、以下のようになります。</p> <p>&filename = "filename with spaces"</p> <p>シンボリック名は、他のいかなるシンボリック名のサブセットとなることはできません。(例えば、&param という名のシンボリック名と、&parameter という名の別のシンボリック名が同一 Process 内に存在することはできません。)</p>	<p>variable string 1 variable string 2 variable string n</p> <p>シンボリック名は、最大 32 文字です。</p>

パラメーター	説明	値
tracel	<p>Process に対して実行するトレース・レベルを指定します。Process によるトレースは、submit コマンド内で、または、Process 定義の一部として有効にすることができます。</p> <p>トレース・レベルの定義の直後に、SNODE または PNODE を指定した場合、このトレース・レベルは、指定されたノードから実行依頼する Process、またはそのノードが実行依頼される Process のすべてに対して有効となります。</p>	<p>level = 0 1 2 4</p> <p>snode pnode</p> <p>file=ファイル名</p> <p>level - トレース出力で表示される詳細レベルを指定します。デフォルトは、4 です。</p> <p>0 - トレースを停止します。</p> <p>1 - 関数入口および関数出口を提供する基本的なレベルです。</p> <p>2 - includes level 1 plus function arguments. レベル 1 に関数の引数を加えます。</p> <p>4 - 完全なトレースを有効にします。実行フロー内の主要ポイントにおける内部データ構造の値といった、基本的な診断情報を表示します。</p> <p>snode - SNODE SMGR のみをトレースするように指定します。</p> <p>pnode - PNODE SMGR のみをトレースするように指定します。</p> <p>file - トレース出力が送信されるファイル名を指定します。ファイル名を指定しない場合、Sterling Connect:Direct の作業ディレクトリー内に、CMGR.TRC と命名された名前のファイルが作成されます。ファイル名の値の長さに制限はありません。</p>

submit コマンドのサンプル

以下は、**submit** コマンドのサンプルです。

毎週実行する Process の実行依頼

以下のコマンドは、payroll という名の Process を実行依頼します。

```
submit file=payroll retain=yes startt=monday pacct="1959,dept-27";
```

このサンプルでは、**retain=yes** が指定されているため、Process は、その実行後、TCQ 内に保持されます。Process は、次の月曜日の 00:00:00 に開始され、その後、毎週月曜日に実行されます。PNODE に対する Process アカウンティング・データが指定されます。TCQ に関する詳細については、『第 3 章: プロセス・キューイング』を参照してください。

開始時刻が指定された Process の実行依頼

以下のコマンドは、copyfil という名前の Process を実行依頼します。

```
submit file=copyfil snode=vmcent startt=(01/01/2008, 11:45:00 am);
```

startt が指定されているため、Process は、2008 年 1 月 1 日の 11:45 a.m. に実行されます。

file の値が存在しない Process の実行依頼

以下のコマンドは、**file** パラメーターの値が存在しないものの、CLI コマンド・プロンプトで Process ステートメントが入力されている Process を実行依頼します。

```
Direct> sub do_copy process snode=node1
step01 copy from (
                file=data.data
                pnode
            )
            to (
                file=b
                snode
            )
            pend ;
Process Submitted, Process Number = 5
```

Process の実行依頼およびトレースの有効化

以下のコマンドは、copy.cdp という名の Process を実行依頼します。

```
submit file=copy.cdp trachel=4 pnode;
```

trachel が指定されており、**pnode** パラメーターが含まれているため、SMGR および COMM の完全なトレースが Process 上で実行されます。トレース情報は、デフォルトのファイルである SMGR.TRC に書き込まれます。

Process パラメーターの変更

change process コマンドを使用して、*非実行中の* Process の指定されたパラメーターを修正します。

Process 名、Process 番号、2 次ノード名、または実行依頼者によって、変更される Process を指定します。

クラス、宛先ノード、優先順位を変更することができます。**release** パラメーターまたは **hold=no** パラメーターを指定して change process コマンドを実行することで、Process を保留キューに配置または保留キューから解放することができます。

startt パラメーターを指定して Process を実行依頼する場合、Sterling Connect:Direct は、Process をタイマー・キューに配置します。change process コマンドで hold=yes を指定することにより、Process が失敗した場合に、その Process を保留キューに移動することができます。

その後、Sterling Connect:Direct は、その Process を HO ステータスで保留キューに配置します。後から、Process を解放して実行することができます。

tracel パラメーターを 1、2、または 4 に設定することで、既存の Process に対してトレースを設定することができます。tracel を 0 に設定することで、Process に対するトレースを無効にすることができます。

以下の検索基準パラメーターのうち少なくとも 1 つのパラメーターを指定してください。

パラメーター	説明	値
pname	Process 名によって、変更する Process を検索します。 Sterling Connect:Direct for Microsoft Windows および Sterling Connect:Direct for z/OS 上では、Process 名は、8 文字に制限されています。	名前 汎用 (リスト) 名前 - Process 名を指定します。最大 8 文字の英数字。 汎用 - Process 名に非特定の値を指定します。パターン・マッチング文字を含むこの汎用値は、0 個以上の pname スtring のリストを評価します。 リスト - Process 名のリストを指定します。リストを括弧で囲み、それぞれの値をコンマで区切ります。
pnumber	Process 番号によって、変更される Process を検索します。Sterling Connect:Direct は、Process の実行依頼時に Process 番号を割り当てます。	1 ~ 99,999 の番号 (リスト) 番号 - Process 番号を指定します。 リスト - Process 番号のリストを指定します。 リストを括弧で囲み、それぞれの値をコンマで区切ります。
snode	2 次ノード名によって、変更される Process を検索します。このパラメーターを使用して、特定のリモート・ノード、リモート・ノード名のマッチング用の汎用値 (パターン・マッチングを使用)、または複数のリモート・ノード名のリストを指定できます。 2 次ノード名は、通常、1 ~ 16 文字のリモート Sterling Connect:Direct ノード名を含みますが、最大 256 文字の任意の英数字の String となることも可能です。リモート・ノード名を IP アドレスまたはホスト名、およびポート番号として指定することもできます。 IP アドレスおよびホスト名の指定に関する詳細については、「IBM Sterling Connect:Direct for UNIX 管理ガイド (IBM Sterling Connect:Direct for UNIX Administration Guide)」の付録「IP アドレス、ホスト名、およびポートの指定 (Specifying IP Addresses, Host Names, and Ports)」を参照してください。	リモート・ノード指定 汎用 (リスト) リモート・ノード指定 - 特定のリモート・ノード名を指定します。 汎用 - リモート・ノード名に非特定値を指定します。パターン・マッチング文字を含むこの汎用値は、0 個以上のリモート・ノード名のリストを評価します。 リスト - リモート・ノード指定のリストを指定します。リストを括弧で囲み、それぞれの値をコンマで区切ります。

パラメーター	説明	値
submitter	Process 所有者のノード指定 (Sterling Connect:Direct ノード名) およびユーザー ID によって、変更される Process を検索します。このパラメーターの文字長に制限はありません。	(ノード指定, ユーザー ID) 汎用 (リスト) ノード指定, ユーザー ID - ノード指定 (Sterling Connect:Direct ノード名) およびユーザー ID を指定します。 汎用 - ノード指定およびユーザー ID の非特定値を指定します。パターン・マッチング文字を含むこの汎用値は、0 個以上のノード指定およびユーザー ID のリストを評価します。 リスト - ノード指定およびユーザー ID のペアのリストを指定します。リストを括弧で囲み、それぞれの値をコンマで区切ります。

以下は、**change process** コマンドのオプションのパラメーターです。

パラメーター	説明	値
class	Process を実行できるノード間セッションを変更します。Process は、指定されたクラス、またはより上位の任意のセッション・クラスで実行できます。デフォルトのクラスは、初期化パラメーター・ファイル内の local.node レコードの sess.default パラメーターとして指定されています。	デフォルトは 1。
hold	Process を保留キューまたは待機キューに移動します。	yes <u>no</u> call yes - 別の change process コマンドによって解放されるまで、Process を HO (Held by Operator) ステータスで保留キュー内に配置します。 No - Process を WC (Waiting for Connection) ステータスで待機キュー内に配置します。この Process は、十分なリソースが利用可能となり次第、実行されます。これはデフォルトです。 call - リモート・ノード (SNODE) がローカル・ノード (PNODE) に接続されるか、または別の Process が実行依頼されるまで、Process を HC (Hold for Call) ステータスで保留キュー内に配置します。接続または実行依頼されると、Sterling Connect:Direct は、Process を解放して実行します。
newsnode	Process に割り当てる新規のリモート・ノード名を指定します。	新規のリモート・ノード指定

パラメーター	説明	値
prty	TCQ 上の Process の優先順位を変更します。Sterling Connect:Direct は、Process の選択に prty パラメーターを使用します。より高い優先順位の Process が、より低い優先順位の Process よりも先に実行されるように選択されます。 prty 値は、伝送時の優先順位には影響を与えません。	1 ~ 15。15 は最も高い優先順位を示します。 prty を指定しない場合、デフォルトは 10 です。
release	保留状態から Process を解放します。このパラメーターは、 hold=no に相当するものです	なし
tracel	Process に対して実行するトレース・レベルを変更します。トレース・レベルの定義の直後に、SNODE または PNODE を特定した場合、このトレース・レベルは、特定されたノードから実行依頼する Process、またはそのノードが実行依頼される Process のすべてに対して有効となります。	level = 0 1 2 4 level - トレース出力で表示される詳細レベルを指定します。デフォルトは、 4 です。 0 - トレースを停止します。 1 - 関数入口および関数出口を提供する基本的なレベルです。 2 - レベル 1 に関数の引数を加えます。 4 - 完全なトレースを有効にします。実行フロー内の主要ポイントにおける内部データ構造の値といった、基本的な診断情報を表示します。

以下のコマンドは、cdproc という名の Process のリモート・ノード名を新しいリモート・ノードの paris に変更します。

```
change process pname=cdproc newsnode=paris;
```

Process の削除

delete process コマンドを使用して、*非実行中*の Process を TCQ から削除します。

Process 名、Process 番号、2 次ノード名、実行依頼者、またはこれらの検索基準/パラメーターの任意の組み合わせによって、削除される Process を選択します。以下の検索基準/パラメーターのうち少なくとも 1 つのパラメーターを指定してください。

パラメーター	説明	値
pname	Process 名によって、削除する Process を特定します。 Sterling Connect:Direct for Microsoft Windows および Sterling Connect:Direct for z/OS 上では、Process 名は、8 文字に制限されています。	名前 汎用 (リスト) 名前 - Process 名を指定します。最大長 8 文字の英数字。 汎用 - Process 名に非特定の値を指定します。パターン・マッチング文字を含むこの汎用値は、0 個以上の pname スtring のリストを評価します。 リスト - Process 名のリストを指定します。リストを括弧で囲み、それぞれの値をコンマで区切ります。
pnumber	Process 番号によって、削除される Process を特定します。Sterling Connect:Direct は、Process の実行依頼時に Process 番号を割り当てます。有効な Process 番号の範囲は、1 ~ 99,999 です。	番号 (リスト) 番号 - Process 番号を指定します。 リスト - Process 番号のリストを指定します。 リストを括弧で囲み、それぞれの値をコンマ (,) で区切ります。
snode	2 次ノード名によって、削除される Process を特定します。このパラメーターを使用して、特定のリモート・ノード、リモート・ノード名のマッチング用の汎用値 (パターン・マッチングを使用)、または複数のリモート・ノード名のリストを指定できます。2 次ノード名は、通常、1 ~ 16 文字のリモート Sterling Connect:Direct ノード名を含みますが、最大 256 文字の任意の英数字の String となることも可能です。リモート・ノード名を IP アドレスまたはホスト名、およびポート番号として指定することもできます。 IP アドレスおよびホスト名の指定に関する詳細については、『IBM Sterling Connect:Direct for UNIX 管理ガイド (IBM Sterling Connect:Direct for UNIX Administration Guide)』の付録『IP アドレス、ホスト名、およびポートの指定 (Specifying IP Addresses, Host Names, and Ports)』を参照してください。	リモート・ノード指定 汎用 (リスト) リモート・ノード指定 - 特定のリモート・ノード名を特定します。 汎用 - リモート・ノード名に非特定値を指定します。パターン・マッチング文字を含むこの汎用値は、0 個以上のリモート・ノード名のリストを評価します。 リスト - リモート・ノード指定のリストを指定します。リストを括弧で囲み、それぞれの値をコンマで区切ります。 。

パラメーター	説明	値
submitter	Process 所有者のノード指定およびユーザー ID によって、削除される Process を特定します。このパラメーターの文字長に制限はありません。	(ノード指定, ユーザー ID) 汎用 (リスト) ノード指定, ユーザー ID - ノード指定およびユーザー ID を指定します。 汎用 - ノード指定およびユーザー ID の非特定値を指定します。パターン・マッチング文字を含むこの汎用値は、0 個以上のノード指定およびユーザー ID のリストを評価します。 リスト - ノード指定およびユーザー ID のペアのリストを指定します。リストを括弧で囲み、それぞれの値をコンマで区切ります。

以下のコマンドは、dallas というノード上の、cduser というユーザー ID によって実行依頼された、すべての非実行中の Process を削除します。

```
delete process submitter=(dallas, cduser);
```

実行キューからの Process の削除

flush process コマンドを使用して、Process を実行キューから削除します。Process 名、Process 番号、2 次ノード名、実行依頼者、またはこれらの検索基準パラメーターの任意の組み合わせによって、削除される Process を選択します。以下の検索基準パラメーターのうち少なくとも 1 つのパラメーターを指定してください。

パラメーター	説明	値
pname	Process 名によって、削除する Process を検索します。 Sterling Connect:Direct for Microsoft Windows および Sterling Connect:Direct for z/OS 上では、Process 名は、8 文字に制限されています。	名前 汎用 (リスト) 名前 - Process 名を指定します。最大 8 文字の英数字。 汎用 - Process 名に非特定の値を指定します。パターン・マッチング文字を含むこの汎用値は、0 個以上の pname スtring のリストを評価します。 リスト - Process 名のリストを指定します。リストを括弧で囲み、それぞれの値をコンマで区切ります。
pnumber	Process 番号によって、削除する Process を検索します。Sterling Connect:Direct は、Process の実行依頼時に Process 番号を割り当てます。	1~99,999 の番号 (リスト) 番号 - Process 番号を指定します。 リスト - Process 番号のリストを指定します。 リストを括弧で囲み、それぞれの値をコンマで区切ります。

パラメーター	説明	値
snode	<p>2 次ノード名によって、削除する Process を検索します。このパラメーターを使用して、特定のリモート・ノード、リモート・ノード名のマッチング用の汎用値 (パターン・マッチングを使用)、または複数のリモート・ノード名のリストを指定できます。</p> <p>2 次ノード名は、通常、1 ~ 16 文字のリモート Sterling Connect:Direct ノード名を含みますが、最大 256 文字の任意の英数字のストリングとなることも可能です。リモート・ノード名を IP アドレスまたはホスト名、およびポート番号として指定することもできます。</p> <p>IP アドレスおよびホスト名の指定に関する詳細については、『<i>IBM Sterling Connect:Direct for UNIX 管理ガイド (IBM Sterling Connect:Direct for UNIX Administration Guide)</i>』の付録『<i>IP アドレス、ホスト名、およびポートの指定 (Specifying IP Addresses, Host Names, and Ports)</i>』を参照してください。</p>	<p><i>リモート・ノード名指定 汎用 (リスト)</i></p> <p><i>リモート・ノード名指定</i> - 特定のリモート・ノード名を指定します。</p> <p><i>汎用</i> - リモート・ノード名に非特定値を指定します。パターン・マッチング文字を含むこの汎用値は、0 個以上のリモート・ノード名のリストを評価します。</p> <p><i>リスト</i> - リモート・ノード指定のリストを指定します。リストを括弧で囲み、それぞれの値をコンマで区切ります。</p>
submitter	<p>Process 所有者のノード指定 (Sterling Connect:Direct ノード名) およびユーザー ID によって、削除される Process を検索します。</p>	<p><i>(ノード指定, ユーザー ID) 汎用 (リスト)</i></p> <p><i>ノード指定, ユーザー ID</i> - ノード指定 (Sterling Connect:Direct ノード名) およびユーザー ID を指定します。</p> <p><i>汎用</i> - ノード指定およびユーザー ID の非特定値を指定します。パターン・マッチング文字を含むこの汎用値は、0 個以上のノード指定およびユーザー ID のリストを評価します。</p> <p><i>リスト</i> - ノード指定およびユーザー ID のペアのリストを指定します。リストを括弧で囲み、それぞれの値をコンマで区切ります。</p>

以下は、**flush process** コマンドのオプションのパラメーターです。

パラメーター	説明	値
force	実行中の Process を強制的に終了、または、Process を、ステップの完了と同時に、正しい順序で終了します。このパラメーターは、Process が実行状態にあり、利用不可のリソースを待機している場合に活用できます。	yes no yes - Process が強制的に即終了するように指定します。SMGR も即時に終了します。 no - Process がステップの完了と同時に、正しい順序で終了されるように指定します。SMGR は、統計ファイルを閉じた後に終了します。これはデフォルトです。
hold	終了された Process を、解放して再実行することができる保留キューに配置します。	yes no yes - Process の終了後、その Process を HS ステータスで保留キュー内に配置するように指定します。 no - Process の終了後、その Process を TCQ から削除するように指定します。これはデフォルトです。

以下のコマンドは、「rome」という名前が付いた実行中の Process のすべてを実行キューからフラッシュします。

```
flush process pname=rome force=yes;
```

以下のコマンドは、jones というユーザー ID によって実行依頼された、alma というノード上の実行中の Process のすべてをフラッシュします。

```
flush process submitter=(alma, jones);
```

Sterling Connect:Direct の停止

stop コマンドを使用して、Sterling Connect:Direct のシャットダウン・シーケンスを正しい順序で開始、またはソフトウェアを強制的に終了します。stop コマンドを実行した後に、新たな Process を実行することはできず、リモート・システムとの新規の接続は確立されません。サーバーが停止するまでは、コマンドの実行およびユーザーによるサインオンが可能です。

stop コマンドでは、force、immediate、quiesce、または step パラメーターを指定することができます。

注: AIX 以外の対応プラットフォームのすべてにおいて、LU6.2 機能を備えた Sterling Connect:Direct を実行する場合には、**force** パラメーターが必要です。

以下は、**stop** コマンドのパラメーターです。

パラメーター	説明
force	Sterling Connect:Direct を強制的に終了し、制御権をオペレーティング・システムに戻します。
immediate	即時かつ正しい順序ですべてのアクティビティーのシャットダウンを開始し、Sterling Connect:Direct を終了します。ソフトウェアは、すべての接続を終了して、統計レコードを作成し、ファイルを閉じた後、シャットダウンします。
quiesce	実行中の Process のすべてを完了するまで実行した後に、Sterling Connect:Direct をシャットダウンします。新しい Process は開始されません。これはデフォルト値です。
step	その時点で実行中の Process ステップのすべてが完了した後、Sterling Connect:Direct をシャットダウンします。ソフトウェアは、統計レコードを書き込み、ファイルを閉じた後、シャットダウンします。すべてのアクティブな Process は、TCQ 内に保持されます。Process は、ソフトウェアが再初期化された際に、再始動します。

以下のコマンドは、Sterling Connect:Direct を強制的に終了し、制御をオペレーティング・システムに戻します。

```
stop force;
```

TCQ 内の Process の表示

ローカル・ノードが Pnode の場合に、**view process** コマンドを使用して、TCQ 内の Process を表示します。Process 名、Process 番号、キュー、2 次ノード、ステータス、Process の所有者、またはこれらの検索基準パラメーターの任意の組み合わせによって検索することができます。

検索基準内で、複数の Process を指定することもできます。

このコマンドにおいては、必須のパラメーターはありません。オプションのパラメーターを指定しない場合、Sterling Connect:Direct は、実行中の Process または実行待機中の Process のすべてを選択します。以下は、**view process** コマンドのオプションのパラメーターです。

パラメーター	説明	値
pname	Process 名によって、表示する Process を検索します。 Sterling Connect:Direct for Microsoft Windows および Sterling Connect:Direct for z/OS 上では、Process 名は、8 文字に制限されています。	名前 汎用 (リスト) 名前 - Process 名を指定します。最大 8 文字の英数字。 汎用 - Process 名に非特定の値を指定します。パターン・マッチング文字を含むこの汎用値は、0 個以上の pname スtring のリストを評価します。 リスト - Process 名のリストを指定します。リストを括弧で囲み、それぞれの値をコンマで区切ります。

パラメーター	説明	値
pnumber	Process 番号によって、表示する Process を検索します。Sterling Connect:Direct は、Process の実行依頼時に Process 番号を割り当てます。	1~99,999 の番号 (リスト) 番号 - Process 番号を指定します。 リスト - Process 番号のリストを指定します。リストを括弧で囲み、それぞれの値をコンマで区切ります。
queue	指定されたキュー名によって、表示する Process を指定します。	all exec hold wait timer all - すべてのキューからの Process を選択します。これはデフォルトです。 exec - 実行キューからの Process を選択します。 hold - 保留キューからの Process を選択します。 timer - タイマー・キューからの Process を選択します。 wait - 待機キューからの Process を選択します。
snode	2 次ノード名によって、Process を表示します。このパラメーターを使用して、特定のリモート・ノード、リモート・ノード名のマッチング用の汎用値 (パターン・マッチングを使用)、または複数のリモート・ノード名のリストを指定できます。 2 次ノード名は、通常、1 ~ 16 文字のリモート Sterling Connect:Direct ノード名を含みますが、最大 256 文字の任意の英数字のストリングとなることも可能です。リモート・ノード名を IP アドレスまたはホスト名、およびポート番号として指定することもできます。 IP アドレスおよびホスト名の指定に関する詳細については、『IBM Sterling Connect:Direct for UNIX 管理ガイド (IBM Sterling Connect:Direct for UNIX Administration Guide)』の付録『IP アドレス、ホスト名、およびポートの指定 (Specifying IP Addresses, Host Names, and Ports)』を参照してください。	リモート・ノード指定 汎用 (リスト) リモート・ノード指定 - 特定のリモート・ノード名を特定します。 汎用 - リモート・ノード名に非特定値を指定します。パターン・マッチング文字を含むこの汎用値は、0 個以上のリモート・ノード名のリストを評価します。 リスト - リモート・ノード指定のリストを指定します。リストを括弧で囲み、それぞれの値をコンマで区切ります。

パラメーター	説明	値
status	Process ステータスによって、表示する Process を指定します。ステータス値を指定しない場合、すべてのステータス値に対する情報が生成されます。	<p>EX HC HE HI HO HR HS PE WC WR WS (リスト)</p> <p>EX (Execution) - 実行キューからの Process を選択するように指定します。</p> <p>HC (Held for Call) - hold=call で実行依頼された Process を選択するように指定します。</p> <p>HE (Held due to Error) - 接続エラーのため保留されている Process を選択するように指定します。</p> <p>HI (Held Initially) - hold=yes で実行依頼された Process を選択するように指定します。</p> <p>HO (Held by Operator) - hold=yes で実行された change process コマンドによって保留されている Process を選択するように指定します。</p> <p>HR (Held Retain) - retain=yes または retain=initial で実行依頼された Process を選択するように指定します。</p> <p>HS (Held Due to Execution Suspension) - hold=yes で実行された flush process コマンドによって停止された Process を選択するように指定します。</p> <p>PE (Pending Execution) - maxdelay パラメーターが指定され、かつ、Process を実行するためにセッション・マネージャーが作成される直前に、Process マネージャーによって PE ステータスが割り当てられて実行依頼された Process を選択するように指定します。Process は、セッション・マネージャーの初期化後、実行キューに置かれ、EX ステータスに変更されます。</p> <p>WC (Waiting for Connection) - 実行の準備ができていないものの、リモート・ノードに対する利用可能な接続のすべてが使用中である Process を選択するように指定します。</p> <p>WR (Waiting for Restart) - セッション・エラーの後、再始動を待機している Process を選択するように指定します。</p> <p>WS (Waiting for Start Time) - 開始時刻を待機している Process を選択するように指定します。これらの Process は、タイマー・キューに存在します。</p> <p>リスト - ステータス値のリストを指定します。リストを括弧で囲み、それぞれの値をコンマで区切ります。</p>
submitter	Process 所有者のノード指定 (Sterling Connect:Direct ノード名) およびユーザー ID によって、表示する Process を検索します。このパラメーターの長さに制限はありません。	<p>(ノード指定, ユーザー ID) 汎用 (リスト)</p> <p>ノード指定, ユーザー ID - ノード指定 (Sterling Connect:Direct ノード名) およびユーザー ID を指定します。</p> <p>汎用 - ノード指定およびユーザー ID に非特定値を指定します。パターン・マッチング文字を含むこの汎用値は、0 個以上のノード指定およびユーザー ID のリストを評価します。</p> <p>リスト - ノード指定およびユーザー ID のペアのリストを指定します。リストを括弧で囲み、それぞれの値をコンマで区切ります。</p>

以下のコマンドは、指定された Process 番号を表示します。

```
view process pnumber=1;
```

TCQ 上の Process ステータスの監視

select process コマンドを使用して、TCQ 内の Process に関する情報を表示します。

検索基準によって、Process の選択において柔軟性が提供されます。Process 名、Process 番号、キュー、2 次ノード、ステータス、Process の所有者、またはこれらの検索基準パラメータの任意の組み合わせによって Process を検索することができます。

検索基準内で、複数の Process を指定することもできます。選択された Process に関する詳細なレポートまたは短いレポートを要求することができます。

このコマンドにおいては、必須のパラメータはありません。オプションのパラメータを指定しない場合、Sterling Connect:Direct は、実行中のプロセスまたは実行待機中のプロセスのすべてを選択します。以下は、**select process** コマンドのオプションのパラメータです。

パラメーター	説明	値
pname	Process 名によって、選択する Process を検索します。Sterling Connect:Direct for Microsoft Windows および Sterling Connect:Direct for z/OS 上では、Process 名は、8 文字に制限されています。	名前 汎用 (リスト) 名前 - Process 名を指定します。最大 8 文字の英数字。 汎用 - Process 名に非特定の値を指定します。パターン・マッチング文字を含むこの汎用値は、0 個以上の pname スtring のリストを評価します。 リスト - Process 名のリストを指定します。リストを括弧で囲み、それぞれの値をコンマで区切ります。
pnumber	Process 番号によって、選択される Process を検索します。Sterling Connect:Direct は、Process の実行依頼時に Process 番号を割り当てます。	1~99,999 の番号 (リスト) 番号 - Process 番号を指定します。 リスト - Process 番号のリストを指定します。リストを括弧で囲み、それぞれの値をコンマで区切ります。
queue	指定されたキュー名によって、選択される Process を指定します。デフォルトは、all です。	all exec hold wait timer all - すべてのキューからの Process を選択します。これはデフォルトです。 exec - 実行キューからの Process を選択します。 hold - 保留キューからの Process を選択します。 timer - タイマー・キューからの Process を選択します。 wait - 待機キューからの Process を選択します。

パラメーター	説明	値
snode	<p>2 次ノード名によって、Process を検索します。このパラメーターを使用して、特定のリモート・ノード、リモート・ノード名のマッチング用の汎用値 (パターン・マッチングを使用)、または複数のリモート・ノード名のリストを指定できます。</p> <p>2 次ノード名は、通常、1 ~ 16 文字のリモート Sterling Connect:Direct ノード名を含みますが、最大 256 文字の任意の英数字のストリングとなることも可能です。リモート・ノード名を IP アドレスまたはホスト名、およびポート番号として指定することもできます。</p> <p>IP アドレスおよびホスト名の指定に関する詳細については、『<i>IBM Sterling Connect:Direct for UNIX 管理ガイド (IBM Sterling Connect:Direct for UNIX Administration Guide)</i>』の付録『<i>IP アドレス、ホスト名、およびポートの指定 (Specifying IP Addresses, Host Names, and Ports)</i>』を参照してください。</p>	<p><i>リモート・ノード指定 汎用 (リスト)</i></p> <p>リモート・ノード指定 - 特定のリモート・ノード名を特定します。</p> <p>汎用 - リモート・ノード名に非特定値を指定します。パターン・マッチング文字を含むこの汎用値は、0 個以上のリモート・ノード名のリストを評価します。</p> <p>リスト - リモート・ノード指定のリストを指定します。リストを括弧で囲み、それぞれの値をコンマで区切ります。</p>

パラメーター	説明	値
status	Process ステータスによって、選択される Process を指定します。ステータス値を指定しない場合、すべてのステータス値に対する情報が生成されます。	<p>EX HC HE HI HO HR HS PE WC WR WS (リスト)</p> <p>EX (Execution) - 実行キューからの Process を選択するように指定します。</p> <p>HC (Held for Call) - hold=call で実行依頼された Process を選択するように指定します。</p> <p>HE (Held due to Error) - 接続エラーのため保留されている Process を選択するように指定します。</p> <p>HI (Held Initially) - hold=yes で実行依頼された Process を選択するように指定します。</p> <p>HO (Held by Operator) - hold=yes で実行された change process コマンドによって保留されている Process を選択するように指定します。</p> <p>HR (Held Retain) - retain=yes または retain=initial で実行依頼された Process を選択するように指定します。</p> <p>HS (Held Due to Execution Suspension) - hold=yes で実行された flush process コマンドによって停止された Process を選択するように指定します。</p> <p>PE (Pending Execution) - maxdelay パラメーターが指定され、かつ、Process を実行するためにセッション・マネージャーが作成される直前に、Process マネージャーによって PE ステータスが割り当てられて実行依頼された Process を選択するように指定します。Process は、セッション・マネージャーの初期化後、実行キューに置かれ、EX ステータスに変更されます。</p> <p>WC (Waiting for Connection) - 実行の準備ができていないものの、リモート・ノードに対する利用可能な接続のすべてが使用中である Process を選択するように指定します。</p> <p>WR (Waiting for Restart) - セッション・エラーの後、再始動を待機している Process を選択するように指定します。</p> <p>WS (Waiting for Start Time) - 開始時刻を待機している Process を選択するように指定します。これらの Process は、タイマー・キューに存在します。リスト - ステータス値のリストを指定します。リストを括弧で囲み、それぞれの値をコンマで区切ります。</p>
submitter	Process 所有者のノード指定 (Sterling Connect:Direct ノード名) およびユーザー IDによって、選択される Process を検索します。このパラメーターの長さに制限はありません。	<p>(ノード指定, ユーザー ID) 汎用 (リスト)</p> <p>ノード指定, ユーザー ID - ノード指定 (Sterling Connect:Direct ノード名) およびユーザー ID を指定します。</p> <p>汎用 - ノード指定およびユーザー ID に非特定値を指定します。パターン・マッチング文字を含むこの汎用値は、0 個以上のノード指定およびユーザー ID のリストを評価します。</p> <p>リスト - ノード指定およびユーザー ID のペアのリストを指定します。リストを括弧で囲み、それぞれの値をコンマで区切ります。</p>

パラメーター	説明	値
detail	選択された Process に関して、Sterling Connect:Direct が生成する、レポートのタイプ (簡易レポートまたは詳細レポート) を指定します。	yes no yes - 詳細レポートを生成します。 no - 簡易レポートを生成します。これはデフォルトです。

以下のコマンドは、指定された Process 番号に関する簡易レポートを表示します。

```
select process pnumber=9 detail=no;
```

このコマンドによる出力は、以下の表のように表示されます。

```
=====
                          SELECT PROCESS
=====
PROCESS NAME  NUMBER  USER   SUBMITTER  NODE      QUEUE    STATUS
-----
PR01          9       root   cd.unix.pj      EXEC     EX
=====
```

以下のコマンドは、指定された Process 番号に関する詳細レポートを表示します。

```
select process pnumber=9 detail=yes;
```

このコマンドによる出力は、以下の表のように表示されます。

```
=====
                          SELECT PROCESS
=====
Process Name   => pr01           Class           => 9
Process Number => 9              Priority        => 8
Submitter Node => cd.unix.pj    PNODE          => cd.unix.pj
Submitter      => sub1       SNODE          => cd.unix.pj
Retain Process => no         Header Type    => p

Submit Time    => 19:52:35   Schedule Time =>
Submit Date    => 05/22/1996  Schedule Date =>

Queue          => EXEC
Process Status => EX
Message Text   =>
=====
```

Processの出力の決定

select statistics cコマンドを使用して、Sterling Connect:Direct 統計ファイルからの Process 統計を調査します。統計レポート内の情報のタイプには、コピー・ステータスおよび実行イベントが含まれます。

この検索基準によって、表示したい情報の選択における柔軟性が提供されます。select statistics コマンドで使用されるパラメーターは、検索基準および情報の表示形式を決定します。条件コード、Process 名、Process 番号、ID タイプ、カテゴリー、2 次ノード、開始時刻、終了時刻、実行依頼者のノード指定およびユーザー ID によって、選択するレコードを指定できます。

このコマンドにおいては、必須パラメーターはありません。オプションのパラメーターを使用して検索要件を示さない場合、Sterling Connect:Direct は、すべての統計レコードを選択しますが、過剰なレコード量となる可能性があります。以下は、select statistics コマンドのパラメーターです。

パラメーター	説明	値
ccode	ステップ終了に関連付けられた完了コード演算子および戻りコード値に基づいて統計ファイルを選択します。戻りコードを指定する必要があります。	演算子, <i>nn</i> 演算子 - 完了コード演算子を指定します。 以下は、有効な完了コード演算子です。 eq または = または == (等しい) これはデフォルトです。 ge または >= または => (以上) gt または > (より大きい) le または <= または =< (以下) lt または < (より小さい) ne または != (等しくない) 戻りコードは、UNIX コマンド、あるいは Sterling Connect:Direct Process またはコマンドの終了ステータスです。 nn - ステップ終了に関連付けられた戻りコード値を指定します。
destfile	宛先ファイル名に基づいて統計を選択します。 注: このパラメーターは、dest と省略できます。	dest=/パス/ファイル名 例: sel stat dest=/sci/payroll/june.payroll; このパラメーターを、srcfile パラメーターと組み合わせて使用することで、ソース・ファイル名および宛先ファイル名に基づいた統計の選択が可能となります。例: sel stat srcf=/sci/accounting/june.payroll dest=/sci/payroll/june.payroll

パラメーター	説明	値
pname	Process 名によって、選択する統計を検索します。 Sterling Connect:Direct for Microsoft Windows および Sterling Connect:Direct for z/OS 上では、Process 名は、8 文字に制限されています。	名前 汎用 (リスト) 名前 - Process 名を指定します。最大 8 文字の英数字。 汎用 - Process 名に非特定の値を指定します。パターン・マッチング文字を含むこの汎用値は、0 個以上の pname スtring のリストを評価します。 リスト - Process 名のリストを指定します。リストを括弧で囲み、それぞれの値をコンマで区切ります。
pnumber	Process 番号によって、選択する統計を検索します。 Sterling Connect:Direct は、Process の実行依頼時に Process 番号を割り当てます。	1 ~ 99,999 の番号 (リスト) 番号 - Process 番号を指定します。 リスト - Process 番号のリストを指定します。リストを括弧で囲み、それぞれの値をコンマで区切ります。
reccat	統計ファイル・レコードがイベントに基づくのか、あるいは Process に関連するのかを指定します。	CAEV CAPR (CAEV, CAPR) CAEV - Sterling Connect:Direct のシャットダウンといったイベントに関連する統計ファイル・レコードを選択するように指定します。 CAPR - 1 個以上の Sterling Connect:Direct Process に関連する統計ファイル・レコードを選択するように指定します。

パラメーター	説明	値
recids	レコード ID によって、選択する統計ファイル・レコード ID を指定します。このパラメーターは、コピー終了レコードまたは Sterling Connect:Direct 初期化イベント・レコードといった、特定のタイプの統計レコードを特定します。	<p>レコード ID (リスト)</p> <p>レコード ID - 指定されたレコード ID の統計ファイル・レコードを選択します。</p> <p>リスト - Process 名のリストを指定します。リストを括弧で囲み、それぞれの値をコンマで区切ります。</p> <p>以下は、有効なレコード ID の値です。</p> <p>APSM - 生成されたライセンス管理エラー</p> <p>CHGP - 実行された change process コマンド</p> <p>CRHT - 著作権表示</p> <p>COAC - API またはリモート・ノードのいずれかに対して有効にされた listen 接続</p> <p>CSPA - 生成された Sterling Connect:Direct Secure Plus エラー</p> <p>CSTP - 停止された子 Process</p> <p>CTRC - コピー終了レコード</p> <p>CTRM - 強制終了された子 Process</p> <p>CUKN - 子 Process の不明なステータス</p> <p>CXIT - 終了された子 Process</p> <p>DELP - 実行された delete Process コマンド</p> <p>FLSP - 実行された flush Process コマンド</p> <p>FMRV - 関数管理で発生したエラー。情報受信操作</p> <p>FMSD - 関数管理で発生したエラー。情報送信操作</p> <p>GPRC - Process 取得中に発生したエラー</p> <p>IFED - 終了された if ステートメント</p> <p>LSST - ローカル・ノード上のステップのレコード ID</p> <p>LIEX - 有効期限が切れたライセンス</p> <p>LWEX - 14 日以内に有効期限が切れるライセンス</p> <p>NINF - スタートアップ時に生成される Sterling Connect:Direct 情報</p> <p>NMOP - 開かれたネットワーク・マップ・ファイル</p> <p>NMPR - Sterling Connect:Direct Browser User Interface、Sterling Control Center Console、または KQV Interface を通じてネットワーク・マップを更新</p> <p>NUIC - 完了された初期化</p> <p>NUIS - 開始された初期化</p> <p>NUTC - 完了された終了</p> <p>NUTS - 開始された終了</p> <p>PERR - Process エラー</p> <p>PFLS - フラッシュされた Process</p> <p>PRED - 終了された Process</p> <p>PRIN - 中断された Process</p> <p>PSAV - 保存された Process</p>

パラメーター	説明	値
recids (続)		PSTR - 開始された Process QCEX - 別のキューから実行キューに移動された Process QCWA - 別のキューから待機キューに移動された Process QCTI - 別のキューからタイマー・キューに移動された Process QCHO - 別のキューから保留キューに移動された Process RJED - 終了された run job RNCF - 失敗したリモート・ノード接続 RSST - リモート・ノード上のステップのレコード ID RTED - 終了された run task RTSY - 再始動された run task。実行されていた run task との再同期 SBED - 終了された submit SELP - 実行された select Process コマンド SELS - 実行された select statistics コマンド SEND - 終了されたセッション SERR - システム・エラー SFSZ - 実行依頼されたファイルのサイズ SGON - KQV Interface または CLI を使用してサインオンしたユーザー SHUD - 発生したシャットダウン SIGC - 取得されたシグナル SSTR - セッションの開始 STOP - 実行された stop コマンド SUBP - 実行された submit コマンド TRAC - 実行された trace コマンド TZDI - ローカル・ノードの時間と協定世界時との時間差 (秒) で示されるローカル・ノードの時間帯 UNKN - 実行された不明なコマンド USEC - ユーザー ID のセキュリティー・チェックにおけるエラー USMG - シャットダウン中の Sterling Connect:Direct XCMM - コマンド・マネージャー (CMGR) メッセージ XCPR - コピーの受信 XCPS - コピーの送信 XIPT - 通信エラー XLKL - 低レベルの TCQ レコード・ロック・エラー XMSG - ユーザー出口に送信されたメッセージ XPAE - Process またはコマンドの実行依頼時に発生した解析エラー XPAM - Process またはコマンドの実行依頼時に発生した解析エラー

パラメーター	説明	値
recids (続)		<p>XPMC - Process マネージャー (PMGR) 接続エラー・メッセージ</p> <p>XPML - PMGR 統計ログ・エラー・メッセージ</p> <p>XPMP - Sterling Connect:Direct プログラム上の権限チェック時の、PMGR エラー・メッセージ</p> <p>XPMP - PMGR RPC およびその他のエラー・メッセージ</p> <p>XPMT - PMGR 終了エラー・メッセージ</p> <p>XRPM - run task または run job エラー・メッセージ</p> <p>XRRF - 相対レコード・ファイルのアクセス・エラー・メッセージ。TCQ に使用されるファイル構造</p> <p>XSMG - セッション・マネージャー (SMGR) エラー・メッセージ</p> <p>XSQF - ファイル・アクセス・エラー・メッセージ</p> <p>XSTA - 開始されたユーザー出口プログラム</p> <p>XTQG - 単一の TCQ エラー・メッセージ・グループ</p> <p>XTQZ - 単一の TCQ エラー・メッセージ・グループ</p>
snode	<p>2 次ノード名によって、統計ファイル・レコードを検索します。このパラメーターを使用して、特定のリモート・ノード、リモート・ノード名のマッチング用の汎用値 (パターン・マッチングを使用)、または複数のリモート・ノード名のリストを指定できます。</p> <p>2 次ノード名は、通常、1 ~ 16 文字のリモート Sterling Connect:Direct ノード名を含みますが、最大 256 文字の任意の英数字のストリングとなることも可能です。リモート・ノード名を IP アドレスまたはホスト名、およびポート番号として指定することもできます。</p> <p>IP アドレスおよびホスト名の指定に関する詳細については、『IBM Sterling Connect:Direct for UNIX 管理ガイド (IBM Sterling Connect:Direct for UNIX Administration Guide)』の付録『IP アドレス、ホスト名、およびポートの指定 (Specifying IP Addresses, Host Names, and Ports)』を参照してください。</p>	<p>リモート・ノード指定 汎用 (リスト)</p> <p>リモート・ノード指定 - 特定のリモート・ノード名を特定します。</p> <p>汎用 - リモート・ノード名に非特定値を指定します。パターン・マッチング文字を含むこの汎用値は、0 個以上のリモート・ノード名のリストを評価します。</p> <p>リスト - リモート・ノード指定のリストを指定します。リストを括弧で囲み、それぞれの値をコンマで区切ります。</p>

パラメーター	説明	値
srcfile	<p>ソース・ファイル名に基づいて統計を選択します。</p> <p>注: このパラメーターは、srcf と省略できます。</p>	<p>srcf=/パス/ファイル名</p> <p>例:</p> <pre>sel stat srcf=/sci/accounting/june.payroll;</pre> <p>このパラメーターを、destfile パラメーターと組み合わせて使用することで、ソース・ファイル名および宛先ファイル名に基づいた統計の選択が可能となります。例:</p> <pre>sel stat srcf=/sci/accounting/june.payroll dest=/sci/payroll/june.payroll</pre>
startt	<p>指定された時間以降に生成されたレコードを選択します。日付または曜日、および時刻は、位置パラメーターです。日付または曜日を指定しない場合は、時刻の前にコンマを付ける必要があります。</p>	<p>[日付 曜日] [, hh:mm:ss [am pm]]</p> <p>日付 - 日 (dd)、月 (mm)、年 (yy) を指定します。mm/dd/yyyy または mm-dd-yyyy としてコーディングできます。日付のみを指定すると、時刻は、00:00:00 にデフォルト設定されます。現在日付がデフォルト設定されます。</p> <p>曜日 - 週の曜日を指定します。値は、today、monday、tuesday、wednesday、thursday、friday、saturday、および sunday です。</p> <p>hh:mm:ss [am pm] - 時刻を、時間 (hh)、分 (mm)、秒 (ss) で指定します。時間は、12 時間形式と 24 時間形式のいずれでも指定できます。12 時間形式を使用する場合は、am または pm を指定する必要があります。デフォルトは 24 時間形式です。デフォルト値は 00:00:00 で、午前零時を表します。曜日の値のみを指定すると、時刻は、00:00:00 にデフォルト設定されます。</p>
stopt	<p>Sterling Connect:Direct が指定された日付、曜日、時刻の位置パラメーターまでの統計レコードを検索するように指定します。日付または曜日を指定しない場合は、時刻の前にコンマを付ける必要があります。</p>	<p>[日付 曜日] [, hh:mm:ss [am pm]]</p> <p>日付 - 日 (dd)、月 (mm)、年 (yy) を指定します。mm/dd/yyyy または mm-dd-yyyy としてコーディングできます。日付のみを指定すると、時刻は、00:00:00 にデフォルト設定されます。現在日付がデフォルト設定されます。</p> <p>曜日 - 週の曜日を指定します。値は、today、monday、tuesday、wednesday、thursday、friday、saturday、および sunday です。</p> <p>hh:mm:ss [am pm] - 時刻を、時間 (hh)、分 (mm)、秒 (ss) で指定します。時間は、12 時間形式と 24 時間形式のいずれでも指定できます。12 時間形式を使用する場合は、am または pm を指定する必要があります。デフォルトは 24 時間形式です。デフォルト値は 00:00:00 で、午前零時を表します。曜日の値のみを指定すると、時刻は、00:00:00 にデフォルト設定されます。</p>

パラメーター	説明	値
submitter	Process 所有者のノード指定 (Sterling Connect:Direct ノード名) およびユーザー ID によって、選択する 統計レコードを検索します。このパラメーターの文字長に制限はありません。	(ノード指定, ユーザー ID) 汎用 (リスト) ノード指定, ユーザー ID - ノード指定 (Sterling Connect:Direct ノード名) およびユーザー ID を指定します。 汎用 - ノード指定およびユーザー ID に非特定値を指定します。パターン・マッチング文字を含むこの汎用値は、0 個以上のノード指定およびユーザー ID のリストを評価します。 リスト - ノード指定およびユーザー ID のペアのリストを指定します。リストを括弧で囲み、それぞれの値をコンマで区切ります。
detail	選択された Process に関して、Sterling Connect:Direct が生成する、レポートのタイプ (簡易レポートまたは詳細レポート) を指定します。	yes no yes - 詳細レポートを生成します。 no - 簡易レポートを生成します。 これはデフォルトです。

以下のセクションは、**select statistics** コマンドによって生成される詳細レポートおよび簡易レポートのサンプルを示しています。

詳細な出力のサンプル

以下のコマンドは、Process 番号 9 の詳細レポートを生成します。

```
select statistics pnumber=9 detail=yes start=(08/10/2008);
```

このレポートには、2008 年 8 月 10 日以降のすべてのレコードが含まれます。

以下のセクションに、2 つのステップのみに関する統計出力のサンプルがリストされています。レコード ID の解釈については、49 ページの *recids* 内の表を使用してください。レコード ID は、表示される各 Process ステップによって異なる可能性があります。完了コードは、Process が正常に実行されたのか、またはエラーが発生したのかどうかを示します。

ロング・テキスト・メッセージを表示するには、『第 4 章: Using Sterling Connect:Direct for UNIX ユーティリティの使用』で説明されている **ndmmsg** コマンドを実行してください。

サマリー出力のサンプル

以下のコマンドは、Process 番号 9 のサマリー統計を生成します。

```
sel stat pnumber=9 detail=no start=(08/10/2008);
```

このレポートには、2008 年 8 月 10 日以降のすべてのレコードが含まれます。

以下の表は、すべての Process ステップをサマリー形式で示した出力のサンプルを示しています。

SELECT STATISTICS					
P RECID	LOG TIME	PNAME	PNUMBER	STEPNAME	CCOD FDBK MSGID
P PSTR	08/10/2008 09:10:39	PR01	9		0 XSMG200I
P IFED	08/10/2008 09:10:44	PR01	9		0 XSMG405I
P CTRC	08/10/2008 09:10:44	PR01	9		0 XSMG405I
P IFED	08/10/2008 09:10:45	PR01	9		4 XSMG400I
P RTED	08/10/2008 09:10:45	PR01	9		0 XSMG400I
P IFED	08/10/2008 09:10:45	PR01	9		4 XSMG400I
P CTRC	08/10/2008 09:10:45	PR01	9		0 XSMG405I
P CTRC	08/10/2008 09:10:45	PR01	9		8 XSMG405I
P CTRC	08/10/2008 09:10:45	PR01	9		8 XSMG405I

select statistics コマンドの実行の際に要する検索時間を短縮するためには、定期的に統計ファイルをアーカイブまたは削除してください。また、**startt** パラメーターおよび **stopt** パラメーターを使用して、出来るだけ厳密に求める統計をまとめます。Process の実行によって、複数の統計レコードが生成されます。Sterling Connect:Direct は、毎日午前零時に、現行の統計ファイルを閉じて、新規の統計ファイルを作成します。ファイル・サイズが初期化パラメーターの `file.size` に設定されている値を超えた場合には、午前零時を待たずに、現行のファイルを閉じることも可能です。デフォルトのファイル・サイズは、1 メガバイトです。

統計ファイルは、`d_dir/work/cd_node` ディレクトリーに存在します。統計ファイルの名前は、**Syyyymmdd.ext** 形式となっており、**yyyy** は年、**mm** は月、**dd** は日を示します。拡張子 (**ext**) は、001 で始まります。この拡張子は、単一の日において新規の統計ファイルが作成されるたびに、1 増加します。

システム診断の実行

診断コマンドである `trace` によって、システムの診断や操作上の問題のトラブルシューティングを実行することができます。以下の表に記載されている適切なパラメーターを使った `trace` コマンドを使用することで、ソフトウェアの Process マネージャー、コマンド・マネージャー、およびセッション・マネージャーのコンポーネント内のランタイム・トレースを有効または無効にすることができます。セッション・マネージャー・トレースにおいては、特定のノードに対するトレースを実行することができます。

コマンド・マネージャー・トレースは、`trace` コマンドを実行したクライアントに対して、即時に有効にされます。`trace` コマンドを実行すると、接続を確立するすべてのクライアントもトレースされるようになります。セッション・マネージャー・トレースは、即時に有効となります。

以下は、**trace** コマンドのパラメーターです。

パラメーター	説明	値
cmgr	コマンド・マネージャーをトレースするためのパラメーターです。	<p>level=0 1 2 4 file=ファイル名</p> <p>level - トレース出力で表示される詳細レベルを指定します。デフォルトは、4 です。</p> <ul style="list-style-type: none"> 0 - トレースを停止します。 1 - 関数入口および関数出口を提供する基本的なレベルです。 2 - レベル 1 および関数引数を含めます。 4 - 完全なトレースを有効にします。実行フロー内の主要ポイントにおける内部データ構造の値といった、基本的な診断情報を表示します。 <p>file - トレースを出力する先のファイル名を指定します。ファイル名を指定しない場合、Sterling Connect:Direct の作業ディレクトリー内に、CMGR.TRC と命名されたファイルが作成されます。ファイル名の値の長さに制限はありません。</p>
comm	<p>セッション・マネージャー内で、リモート Sterling Connect:Direct システムとの間で送受信されるデータをトレースするためのパラメーターです。このトレースは、smgr トレースと一緒に設定することも、単独で設定することも可能です。</p> <p>comm および smgr トレースの両方を実行する場合、両トレースのトレースは、最後に指定されたトレースのファイル名に対して出力されます。</p>	<p>level=0 1 2 4 file=ファイル名</p> <p>level - トレース出力で表示される詳細レベルを指定します。デフォルトは、4 です。</p> <ul style="list-style-type: none"> 0 - トレースを停止します。 1 - 関数入口および関数出口を提供する基本的なレベルです。 2 - レベル 1 および関数引数を含めます。 4 - 完全なトレースを有効化します。実行フロー内の主要ポイントにおける内部データ構造の値といった、基本的な診断情報を表示します。 <p>file - トレースが出力される先のファイル名を指定します。ファイル名を指定しない場合、Sterling Connect:Direct の作業ディレクトリー内に、COMM.TRC と命名されたファイルが作成されます。ファイル名の値の長さに制限はありません。デフォルトのファイル名は、COMM.TRC です。</p>

パラメーター	説明	値
pmgr	Process マネージャーをトレースするためのパラメーターです。	<p>level=0 1 2 4 file=ファイル名</p> <p>level - トレース出力で表示される詳細レベルを指定します。デフォルトは 4 です。</p> <ul style="list-style-type: none"> 0 - トレースを停止します。 1 - 関数入口および関数出口を提供する基本的なレベルです。 2 - レベル 1 および関数引数を含めます。 4 - 完全なトレースを有効化します。実行フロー内の主要ポイントにおける内部データ構造の値といった、基本的な診断情報を表示します <p>file - トレースが出力される先のファイル名を指定します。ファイル名を指定しない場合、Sterling Connect:Direct の作業ディレクトリー内に、PMGR.TRC と命名されたファイルが作成されます。ファイル名の値の長さに制限はありません。</p>
smgr	<p>コマンドを実行した後に作成されるセッション・マネージャーに対するトレースを実行するためのパラメーターです。その時点で実行中のセッション・マネージャーに影響は与えません。</p> <p>comm および smgr トレースの両方を実行する場合、両トレースのトレースは、最後に指定されたトレースのファイル名に対して出力されます。</p>	<p>level=0 1 2 4 snode pnode tnode file=ファイル名</p> <p>level - トレース出力で表示される詳細レベルを指定します。デフォルトは、4 です。</p> <ul style="list-style-type: none"> 0 - トレースを停止します。 1 - 関数入口および関数出口を提供する基本的なレベルです。 2 - レベル 1 および関数引数を含めます。 4 - 完全なトレースを有効化します。実行フロー内の主要ポイントにおける内部データ構成の値といった、基本的な診断情報を表示します。 <p>snode - SNODE SMGR のみをトレースするように指定します。</p> <p>pnode - PNODE SMGR のみをトレースするように指定します。</p> <p>tnode - トレースを実行するノードを特定します。複数のノードに関するトレース情報を収集したい場合は、このパラメーターに複数のノードを指定します。</p> <p>file - トレースが出力される先のファイル名を指定します。ファイル名を指定しない場合、Sterling Connect:Direct の作業ディレクトリー内に、SMGR.TRC と命名されたファイルが作成されます。ファイル名の値の長さに制限はありません。デフォルトのファイル名は、SMGR.TRC です。</p>

以下の trace コマンドのサンプルは、ath3500ry と呼ばれるノードのセッション・マネージャー上でレベル 2 のトレースを実行し、ファイル Smgp.trcに出力を書き込みます。

```
trace smgr pnode tnode=ath3500ry level=2 file=Smgp.trc;
```

以下のセクションは、トレース出力のサンプルの一部を示しています。トレースは、Process ID、関数、月と日、および時刻 (マイクロ秒) で特定します。最初の列には Process ID が含まれます。2 番目の列は、月および日を、MM/DD の形式で表し、3 番目のカラムは、時刻を HH:MM:SSSS の形式で表します。最後のカラムは、関数を示します。右矢印は、関数が入力されたことを示し、左矢印は、関数が終了したことを示します。

一部の関数は字下げされており、ネスティングを示しています。字下げされた矢印は、その関数が先行の関数によって呼び出されたことを示します。

```

=====
498    05/18 15:13:0104    cm_sendcmd_1 entered.
498    05/18 15:13:0206    -> ndm_error_destroy
    <- ndm_error_destroy: ok
498    05/18 15:13:0506    -> ndm_error_create
    <- ndm_error_create: ok
498    05/18 15:13:0708    ndm_cmds_free entered.
    ndm_cmds_free exited.
498    05/18 15:13:0801    -> ndm_parser_jdi
498    05/18 15:13:0806    -> ndm_error_create
    <- ndm_error_create: ok
498    05/18 15:13:0916    ->Parser: SELPRO
498    05/18 15:13:0926    ->bldexp
    <-bldexp: Null argument value,
    don't add.
498    05/18 15:13:1116    ->bldexp
498    05/18 15:13:1136    -> ndm_crit_comp
498    05/18 15:13:1155    ->compile

    <-compile
    <- ndm_crit_comp: Handle
    <-bldexp: ok
...
=====

```

Process キューイング

この章では、以下を含めた伝送制御キュー (TCQ) に関する情報が記載されています。

- ◆ TCQ の説明
- ◆ Sterling Connect:Direct アクティビティのスケジューリング
- ◆ 待機キュー、実行キュー、保留キュー、タイマー・キューのステータス値を含めた、TCQ を通した Process の進捗

伝送制御キュー (TCQ)

TCQ は、Sterling Connect:Direct が稼動する際に、Process 実行を制御します。Process を実行依頼した後、その Process は TCQ 内に保存されます。TCQ は、実行キュー、待機キュー、タイマー・キュー、および保留キューの 4 つのキューで構成されています。

Process を実行依頼した後、適切なコマンドを使用することによって、ステータスの監視、特定の特性の修正、および実行の停止を行うことができます。以下の表に記載されているコマンドによって、これらのタスクを実行することができます。

コマンド	定義
change process	TCQ 内の <i>非実行中</i> の Process のステータスの変更や特性の修正を行います。
delete process	待機キュー、タイマー・キュー、および保留キューから <i>非実行中</i> の Process を削除します。
flush process	実行キューから <i>実行中</i> の Process を削除します。
select process	実行中の Process を含めた、TCQ 内の Process を監視します。
view process	TCQ 内の Process を表示します。

Sterling Connect:Direct アクティビティのスケジューリング

Sterling Connect:Direct は、スケジューリングに影響を与えるパラメーターに基づいて、Process をキューに配置します。**Process** ステートメントまたは **submit** コマンド内でスケジューリング・パラメーターを指定できます。

スケジューリング・パラメーターは、以下のセクションに記載されています。

- ◆ retain=yes|no|initial
- ◆ hold=yes|no|call
- ◆ startt=[[([date|day] [, hh:mm:ss | [am | pm]])]

以下の表は、スケジューリング・パラメーターが論理キューに与える影響について説明しています。

スケジューリング・パラメーター	キュー	説明
スケジューリング・パラメーターの指定なし	待機	Process は、Sterling Connect:Direct とリモート・ノードとのセッションが確立されるまで、待機キューに保持されます。セッション確立後、Process は実行キューに移動されます。
retain=yes	保留	startt パラメーター値を指定している場合を除き、Process のコピーが一度実行されます。Process が開始される日または時刻、あるいはその両方を指定します。
retain=no	待機 (他のパラメーターが指定されていない場合)	Process は、Sterling Connect:Direct とリモート・ノードとのセッションが確立されるまで、待機キューに保持されます。デフォルトは no です。
retain=initial	保留	Process のコピーが保留キューに保持され、Process マネージャーが初期化される度に実行されます。
retain=yes かつ、hold=no または hold=call	保留	Process のコピーが保留キューに保持され、解放時に実行されます。
hold=yes	保留	release パラメーターを使用して change process コマンドを指定することにより、Process を実行することができます。
hold=no	待機 (他のパラメーターが指定されていない場合)	hold のデフォルト値は no です。
hold=call	保留	Process は、リモート・ノードがローカル・ノードとのセッションを開始するか、または別の Process がそのリモート・ノードとのセッションを開始するまで、保留キューに保持されます。
startt	タイマー	スケジュールされた日および時刻に到達すると、Process は、待機キューに移動されます。

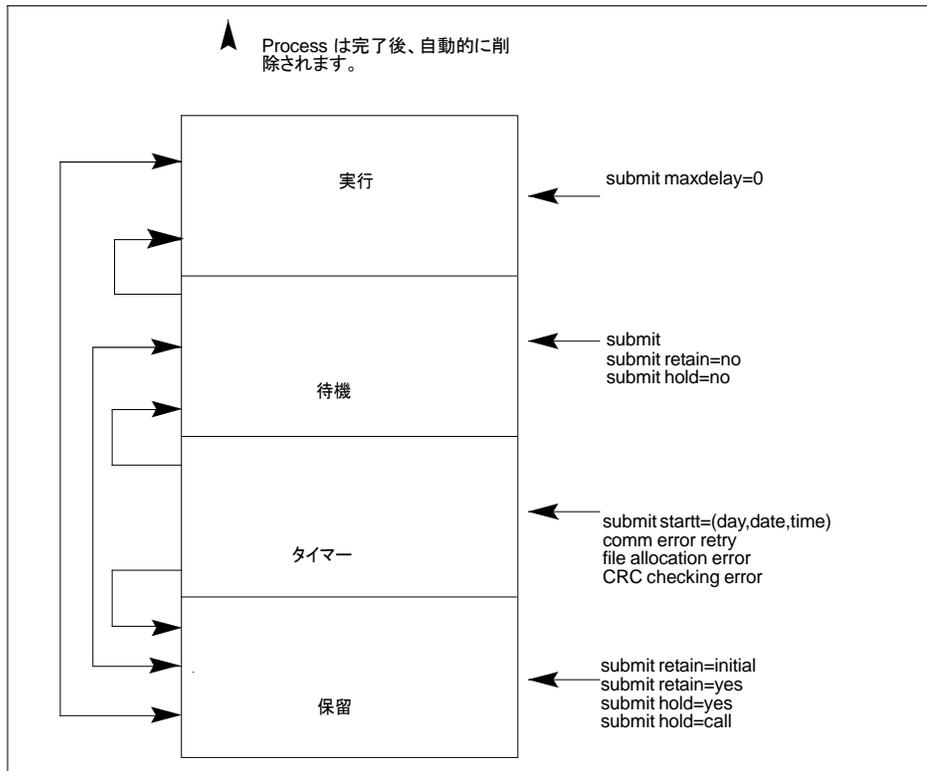
EaTCQ 内の各 Process には、関連付けられたステータス値が存在します。各ステータス値は、Process が配置される論理キューによって影響される特有の意味を持ちます。各キューのステータス値は、以下のセクションの表に示されています。**select process** コマンドを使用すること

で、TCQ 内の Process のステータスを調べることができます。例えば、以下のコマンドは、TCQ 内のEX ステータスを持つ Process のすべてを表示します。

```
select process status=EX;
```

TCQ を通した Process の進捗

このセクションでは、TCQ の各論理キューおよびこれらのキューを通した Process の進捗について説明しています。以下の図は、4 つの論理キューおよびこれらの論理キューに関連付けられたパラメーター値を示しています。



実行キュー

Process は、Sterling Connect:Direct がリモート・ノードに接続されると、実行キューに配置されます。Process は通常、待機キューから移動されて来ますが、maxdelay=0 が指定された submit コマンドによって実行キューに配置される場合もあります。

実行キュー内の Process は、Execution (EX) ステータスまたは Pending Execution (PE) ステータスである可能性があります。EX ステータスの Process は、2 つの Sterling Connect:Direct ノード間でデータ交換を行っています。

PE ステータスの Process は、ローカル・ノードおよびリモート・ノード間で Process 開始メッセージが交換されるのを待機しています。Process には通常、非常に短い時間、PE ステータスが割り当てられます。

Process が正常に完了すると、その Process は自動的に実行キューから削除されます。hold=yes が指定されている **flush process** コマンドは、Process を実行キューから保留キューに移動します。再試行 (retry) の値が指定されている場合、セッションが中断されると、Process は実行キューからタイマー・キューに移動されます。再試行の値が消費する前に接続が確立されない場合、または再試行の値が指定されていない場合には、**conn.retry.exhaust.action** のパラメーターに基づいたアクションが取られます。デフォルトでは、Process は保留キューに移動されます。

以下の表は、実行キューのステータス値を示しています。

ステータス	説明
PE	Pending Execution (PE) は、maxdelay=0 で Process が実行依頼された際の初期キュー・ステータスです。
EX	Execution (EX) ステータスは、Process が実行中であることを示します。

待機キュー

待機キュー内の Process は、ローカル・ノードおよびリモート・ノード間で新規の接続または既存の接続が利用可能になるのを待機しています。

Process は、保留キューまたはタイマー・キューから移動されて来る可能性があります。Process はまた、submit コマンドのパラメーターの指定なし、retain=no、または hold=no で実行依頼されることで、待機キューに配置される場合もあります。

Process は、接続が確立されると、自動的に実行キューに移動されます。

以下の表は、待機キューのステータス値を示しています。

ステータス	説明
WC	このステータスは、Process がすぐに実行できる状態にあるものの、利用できるセッションがないことを示します。SNODE が他の Process を実行している可能性があり、その他に利用できるセッションが存在しません。この Process は、新しいセッションが作成されるか、または既存のセッションが利用可能になり次第、すぐに実行されます。
WR	このステータスは、Process が再試行ステータスにあることを示します。再試行の回数および再試行の間隔は、ネットワーク・マップ内で指定されます。
WA	このステータスは、hold 値または retain 値を指定せずに Process が実行依頼される際の最初のキュー・ステータスを示します。この Process は、すぐに実行できる状態にあります。
WS	このステータスは、Process が PNODE によるセッション続行を待機していることを示します。

タイマー・キュー

Process は、startt パラメーターが指定された submit コマンドによって、タイマー・キューに配置されます。Waiting for Start Time (WS) ステータスにある Process は、待機キューに移動するにあたって、開始時刻に到達するのを待機しています。Process はまた、以下のいずれかのエラー条件が発生した場合にも、再試行 (WC) ステータスでタイマー・キューに配置されます。

- ◆ ローカル・ノードまたはリモート・ノードでの Process の実行中に、ファイル割り当てエラーが発生し、そのファイル割り当てエラーが再試行の条件として指定されている場合、その Process はタイマーキューに配置されます。この Process はその後、short-term および long-term の再試行パラメーター定義に従って再試行されます。この機能により、呼び出したファイルが利用できなかったために実行に失敗した Process を後から再試行することが可能となります。
- ◆ Process の実行中に、接続エラーが発生すると、インテリジェント・セッション再試行機能によって、ノードに対してスケジュールされている Process のすべてが、実行中の Process を含めて、タイマー・キューに配置されます。この機能により、接続が失われた場合でも、ノード上の各 Process の再試行に不可避のオーバーヘッドが除去されます。
- ◆ CRC 検査が有効にされている場合、CRC エラーを生成した Process はタイマー・キューに配置されます。

Sterling Connect:Direct は、ネットワーク・マップ・パラメーターで指定されている再試行の回数および再試行の間の遅延に基づいて、自動的に Process を再試行します。

Process は、タイマー・キューから待機キューに移動します。hold=yes が指定された **change process** コマンドは、指定された Process をタイマー・キューから保留キューに移動します。以下の表は、タイマー・キューのステータス値を示しています。

ステータス	説明
WR	このステータスは、Process が再試行ステータスにあることを示します。再試行の回数および再試行の間隔は、ネットワーク・マップ内で指定されます。
WS	このステータスは、期限にまだ到達していない開始時刻または日付を指定して Process が実行依頼されたことを示します。startt に到達すると、Process は待機キューに配置されて、実行がスケジュールされます。
HR	このステータスは、retain=yes または retain=initial を指定して Process が実行依頼されており、Process が既に実行済みであることを示します。この Process は、release が指定された change process コマンドによって、後から解放することができます。
WC	このステータスは、Process がすぐに実行できる状態にあるものの、利用できるセッションがないことを示します。他の Process が SNODE によって実行されている可能性があり、その他に利用できるセッションが存在しません。この Process は、新しいセッションが作成されるか、または既存のセッションが利用可能になり次第、すぐに実行されます。

保留キュー

保留キュー内の Process は、待機キューに進む前にオペレーターの介入を待機します。このキューによって、ローカル・ノードおよびリモート・ノードのオペレーターによる、Process 実行の調整および制御が可能となります。

Process は、retain=initial、retain=yes、または hold=yes パラメーターが指定された submit コマンドによって、保留キュー内に配置されます。hold=call を指定して実行依頼された Process も保留キューに配置されます。Process は、hold=yes が指定された **change process** コマンドによって、タイマー・キューから保留キューに移動されます。さらに、Process は、hold=yes が指定された **flush process** コマンドによって、実行キューから保留キューに移動されます。

Process は、**release** パラメーターを指定した **change process** コマンドによって、保留キューから実行キューに移動されます。

以下の表は、保留キューのステータス値を示しています。

ステータス	説明
HC	HC (Held for Call) は、hold=call を指定して Process が実行依頼されたことを示します。いずれかのノードでセッションが開始されることにより、Process は、WC ステータスで待機キューに移動されます。この Process は、実行に向けて選択されると、実行キューに配置されます。
HI	HI (Held Initially) は、hold=yes を指定して Process が実行依頼されたことを示します。この Process は、release または hold=no が指定された change process コマンドによって、後から解放することができます。
HE	HE (Held due to Error) は、セッション・エラーまたはその他の異常の発生を指定します。
HO	HO (Held by Operator) は、change process の hold=yes が指定されたことを示します。
HR	HR (Held Retain) は、retain=yes または retain=initial を指定して Process が実行依頼されており、Process が既に実行済みであることを示します。この Process は、release が指定された change process コマンドによって、後から解放することができます。
HS	HS (Held for Suspension) は、オペレーターが hold=yes を指定して flush process コマンドを実行したことを示します。この Process は、release が指定された change process コマンドによって、後から解放することができます。

Using Sterling Connect:Direct for UNIX ユーティリティの使用

Sterling Connect:Direct は、以下のタスクの実行に使用できるツールおよびユーティリティを提供します。

- ◆ 変換テーブルの作業
- ◆ Sterling Connect:Direct メッセージへのアクセス
- ◆ スタンドアロン型バッチ圧縮ユーティリティを使用した、ファイルの事前圧縮/圧縮解除
- ◆ 構成ファイルの検証
- ◆ 構成レポートの生成

変換テーブルの作業

Sterling Connect:Direct は、`d_dir/ndm/xlate` ディレクトリー内の文字変換テーブルに基づいて、ASCII から EBCDIC といったように、1 つの文字セット・コードから異なる文字セット・コードにデータを変換します。Sterling Connect:Direct は、ファイル転送の操作時に使用するデフォルトの文字変換テーブルを提供しますが、`ndmxlt` と呼ばれるユーティリティ・プログラムを使用してこの文字変換テーブルを変更することができます。

変換テーブルを作成するには、`/cd_dir/cdunix/ndm/src/def_send.sxlt` または `/cd_dir/cdunix/ndm/src/def_recv.sxlt` (`cd_dir` は Sterling Connect:Direct がインストールされているディレクトリー) と呼ばれるファイルをコピーして、名前を変更またはそのファイルを修正します。テキスト・エディターを使用して、作成したファイル内のテーブルに新しい値を追加し、`ndmxlt` ユーティリティを使用して、更新されたファイルをコンパイルします。`d_dir/ndm/xlate` 内のデフォルトの変換テーブルを、更新されたテーブルに置き換えます。各テーブルは、256 バイト長です。

以下は、変換テーブルのサンプルです。

```
# This file contains an example of defining an ASCII-to-EBCDIC translation table and
# then changing it to translate lowercase to uppercase.
#
# Define the ASCII-to-EBCDIC table. offset=0

00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F
40 5A 7F 7B 5B 6C 50 7D 4D 5D 5C 4E 6B 60 4B 61
F0 F1 F2 F3 F4 F5 F6 F7 F8 F9 7A 5E 4C 7E 6E 6F
7C C1 C2 C3 C4 C5 C6 C7 C8 C9 D1 D2 D3 D4 D5 D6
D7 D8 D9 E2 E3 E4 E5 E6 E7 E8 E9 AD E0 BD 5F 6D
79 81 82 83 84 85 86 87 88 89 91 92 93 94 95 96
97 98 99 A2 A3 A4 A5 A6 A7 A8 A9 C0 4F D0 A1 7F
80 81 82 83 84 85 86 87 88 89 8A 8B 8C 8D 8E 8F
90 91 92 93 94 95 96 97 98 99 9A 9B 9C 9D 9E 9F
A0 A1 A2 A3 A4 A5 A6 A7 A8 A9 AA AB AC AD AE AF
B0 B1 B2 B3 B4 B5 B6 B7 B8 B9 BA BB BC BD BE BF
C0 C1 C2 C3 C4 C5 C6 C7 C8 C9 CA CB CC CD CE CF
D0 D1 D2 D3 D4 D5 D6 D7 D8 D9 DA DB DC DD DE DF
E0 E1 E2 E3 E4 E5 E6 E7 E8 E9 EA EB EC ED EE EF
F0 F1 F2 F3 F4 F5 F6 F7 F8 F9 FA FB FC FD FE FF
#
# Change the lowercase characters to uppercase.
offset=61
C1 C2 C3 C4 C5 C6 C7 C8 C9 D1 D2 D3 D4 D5 D6 D7
D8 D9 E2 E3 E4 E5 E6 E7 E8 E9
```

各バイトには、ターゲット文字セットの文字値が格納されています。ソース文字セットは、テーブルへのインデックスとして使用されます。例えば、ASCII ブランク (16 進数 20) は、変換テーブル内のオフセット 16 進数 20 にあるバイトを検索します。16 進数 20 のロケーションにあるバイトに 16 進数 40 が含まれていると、これがブランク文字を示す EBCDIC コードになります。

変換テーブルの作成および修正

ndmxmlt ユーティリティ・プログラムを使用して、要件を満たすようにカスタマイズされた変換テーブルを作成または修正することができます。**ndmxmlt** ユーティリティを呼び出すには、UNIX プロンプトで以下のコマンドを入力します。

```
$ ndmxmlt -ssourcefile -ooutputfile [ -rradix ] [ -ffiller ] -mxmlatefile
```

以下のセクションは、**ndmxmlt** コマンドのパラメーターを示しています。

パラメーター	説明	値
-ssourcefile	変換テーブル・ソース・ファイルのパスおよびファイル名。値が指定されない場合、入力標準入力から読み込まれます。	変換テーブルのパスおよび名前
-ooutputfile	変換テーブル出力ファイルのパスおよびファイル名。	変換出力ファイルのパスおよび名前。

パラメーター	説明	値
-rradix	ソース・ファイル入力データの基数または底。コマンドライン・オプションからの数値または入力データからの数値に関わらず、すべての数値が、基数設定に基づいて解釈されます。	x d o x - 16 進数。これはデフォルトです。 d - 10 進数。 o - 8 進数。 デフォルトは、x です。
-ffiller	フィルラー・バイト値。入力データがスキャンされて、テーブルに適用される前に、テーブル全体がこの値に初期化されます。	あらゆるキーボード上の文字、数字、または特殊文字。さらに、先頭にスラッシュを使用して入力される制御文字。 例えば、「¥0」はヌルです。
-m	モデル変換テーブルのパスおよびファイル名。指定した場合、モデル・テーブルが読み込まれた後、入力データのスキャンが行われ、テーブルに適用されます。この機能によって、各テーブルに対して 256 バイトの入力データをすべて指定する必要なく、単一の基本テーブルのパリエーションとなる異なるテーブルを多数作成することができます。	モデル変換テーブルのパスおよびファイル名。

このセクションでは、変換テーブルの作成および既存の変換テーブルの修正を行う方法の例を記載しています。

例: 変換テーブルの作成

以下の手順を実行して、小文字を大文字に変更する変換テーブルのサンプルを作成します。

1. `cd_dir/ndm/src/def_send.sxlt` に置かれている変換テーブルのサンプルのコピーを作成します。
2. テキスト・エディターを使用して、新規の変換テーブルを開きます。
3. テーブルの一番下に、以下のラインを追加します。この情報を追加すると、65 ページに示されているテーブルのようになります。

```
#
# Change the lowercase characters to uppercase.
offset=61
C1 C2 C3 C4 C5 C6 C7 C8 C9 D1 D2 D3 D4 D5 D6 D7
D8 D9 E2 E3 E4 E5 E6 E7 E8 E9
```

4. 修正されたファイルを `cd_dir/ndm/src` にコピーして、`UpperCaseEBC.sxlt` と命名します。
5. 以下の構文を使用して、新規の変換テーブルをコンパイルします。

```
ndmxlt -s./src/UpperCaseEBC.sxlt -oUpperCaseEBC.xlt
```

6. この変換テーブルを使用するには、以下の sysopts パラメーターを copy ステートメントに追加します。

```
copy from file=filename
to file=filename
sysopts=":xlate.tbl=pathname/UpperCaseEBC.xlt:"
```

例: モデル変換テーブルの修正

以下の手順を実行して、モデル変換テーブルを修正します。この方法が実装されると、モデル・テーブルが読み込まれて、新規のファイルに書き出されます。その後、入力データが読み込まれて、作成されたテーブルが変更されます。

1. FourLinesUpperCase.sxlt と呼ばれるファイルを作成し、そのファイルに以下の行を追加します。

```
#
# Change the lowercase characters to uppercase.
offset=61
C1 C2 C3 C4 C5 C6 C7 C8 C9 D1 D2 D3 D4 D5 D6 D7
D8 D9 E2 E3 E4 E5 E6 E7 E8 E9
```

2. 修正したファイルを *cd_dir/ndm/src* にコピーします。
3. 以下のコマンドを入力することにより、このファイルをコンパイルし、*fourLineUpperCase.xlt* と呼ばれる変換テーブルを作成します。

```
ndmxlt -s. ./src/FourLineUpperCase.sxlt -oFourLineUpperCase.xlt -mdef_send.xlt
```

4. この変換テーブルを使用するには、以下の sysopts パラメーターを copy ステートメントに追加します。

```
copy from file=filename
to file=filename
sysopts=":xlate.tbl=pathname/FourLineUpperCase.xlt:"
```

ファイル転送操作時の変換の使用

変換は Sterling Connect:Direct Process の copy ステートメント内で指定されます。デフォルトの変換テーブルを使用するか、または新規のテーブルを作成することができます。デフォルトの変換テーブルを使用するには、以下の copy ステートメントを入力します。

```
copy from file=abc to file=xyz sysopts=":xlate=yes:"
```

データ変換においてカスタマイズされたテーブルを指定するには、copy ステートメント内に以下の sysopts サブパラメーターを含めます。*pathname/filename* は、変換テーブルを指定します。

```
copy from file=filename
to file=filename
sysopts=":xlate.tbl=pathname/filename:"
```

copy ステートメントでの変換テーブルの指定に関する詳細については、次の URL から、IBM Sterling Connect:Direct Process の Web サイトの UNIX のセクションを参照してください：
<http://www.sterlingcommerce.com/documentation/processes/processhome.html>

診断

以下の表は、ndmxmlt によって生成されたエラー・メッセージを表示しています。

診断番号	説明
XXLT001I	無効なディレクティブ
XXLT002I	入力ファイルのオープン・エラー
XXLT003I	モデル・ファイルのオープン・エラー
XXLT004I	無効なフィルター値
XXLT005I	無効なオフセット値
XXLT006I	無効な基数値
XXLT007I	無効なテーブル値
XXLT008I	範囲外のテーブル・データ

Sterling Connect:Direct メッセージへのアクセス

Sterling Connect:Direct メッセージ・ファイルには、ホスト・サーバー以外の Sterling Connect:Direct サーバーからのメッセージおよびエラーを含む、すべてのメッセージ・テキストを備えたレコードが含まれています。テキスト・エディターを使用して、メッセージ・レコードを追加および削除することができます。メッセージ・ファイルは、
`d_dir/ndm/cfg/cd_node/msgfile.cfg` にあります。**ndmmsg** コマンドを使用して、メッセージ・テキストを表示することができます。

メッセージ・ファイルの内容

メッセージ・ファイルは、他の Sterling Connect:Direct 構成ファイルとほぼ同じ形式で構成されています。各レコードは、1 つ以上の物理行で構成されるテキスト・ファイル内の論理行です。各レコードには、一意的な名前、メッセージ ID、およびメッセージ・テキストを構成するフィールドが存在します。

メッセージ・レコード定義はシンボリック置換を提供します。これにより、問題をより明示的に特定するために、実際のファイル名または他の変数情報をテキスト内に含めることが可能となります。シンボリック変数は、アンパーサンド文字 (&) で始まります。

以下の表は、Sterling Connect:Direct メッセージ ID のフォーマットをリストしています。

```
Xxxxxnnl Where:

X Indicates Sterling Connect:Direct
xxx is a 3-character Sterling Connect:Direct component identifier
nnn is a 3-digit decimal number
l is the standard, though not required, suffix
```

メッセージ・ファイル・レコードのフォーマットの理解

以下の例は、メッセージ・ファイル・レコードのフォーマットを示しています。各レコードは、最大 4 キロバイト長になることができます。オプションのパラメーターおよび値は、括弧で囲まれています。

```
message id [long.text detailed message explanation] [mod.name issuing module name] short.text message summary
```

以下は、メッセージ・ファイル・レコードのパラメーターです。

パラメーター	説明	値
long.text	メッセージを詳細に説明するストリング。	テキスト・ストリング
mod.name	メッセージ ID を発行するソース・モジュール名。	ソース・モジュール名
short.text	メッセージのサマリー。このフィールドは必須です。	サマリー・メッセージ。最大 72 文字

以下の例は、XCPS008I のメッセージ・レコードのサンプルを示しています。

```
XCPS008I:mod.name=NUSMCP00.C:
:short.text=File is not VB datatype.:
:long.text=File is not variable block. Change sysopts datatype to
either binary or text to transfer this file.
%nSYSTEM ACTION-> the copy step failed and CD processing
continued with the next process step.
%nRESPONSE-> change the sysopts datatype to either
binary or text.:
```

メッセージ・テキストの表示

ndmmsg コマンドを使用して、メッセージ・ファイル内のテキストを表示します。ショート・テキストおよびロング・テキストの両方を表示することができます。

以下のコマンドは、ndmmsg のフォーマットを示します。

```
ndmmsg -f msgfname [-l | -s] msgid1 [msgid2 [msgid3 [...]]]
```

以下は、**ndmmsg** コマンドのパラメーターです。l または s のパラメーターを指定しない場合、ショート・テキストとロング・テキストの両方が表示されます。

パラメーター	説明
-f	メッセージ・ファイル名を指定します。
-l	メッセージのロング・テキストを表示します。
-s	メッセージのショート・テキストを表示します。

以下は、**ndmmsg** コマンドのサンプルです。

```
ndmmsg -f /usr/ndmunix/msgfile.cfg XCMG000l
```

以下の例では、上記のコマンドによる出力を表示しています。

```
rc=&rc
fdbk=&fdbk
mod.name=NUCMRG00.C
func.name=ndmapi_sendcmd
short.text=CMGR RPC call returns NULL
long.text=The ndmapi_sendcmd RPC call made by the API to the CMGR returns a NULL
pointer. There is probably an RPC error.
ndm.action=None
user.action=First, check if the ndmcmgr is still running; it could have been killed
accidently. If so, then abort the current CLI and restart the CLI. If the same
problem occurs again, try to increase the value of wait time (if set) in the API
configuration file (ndmapi.cfg).
```

スタンドアロン型バッチ圧縮ユーティリティを使用した、ファイルの事前圧縮/圧縮解除

スタンドアロン型のバッチ圧縮ユーティリティ (cdsacomp) によりファイルを事前圧縮した後、Sterling Connect:Direct Process を使用して、事前圧縮したファイルをリモート Sterling Connect:Direct ノードに転送することができます。ファイルの圧縮解除においては、以下のオプションが存在します。ファイルは、以下のいずれかが可能です。

- ◆ リモート・ノードによる受信時に圧縮解除 (すべての Sterling Connect:Direct プラットフォームで利用可能)
- ◆ リモート・ノード上に保存して、後で cdsacomp を使用してオフラインで圧縮解除 (Sterling Connect:Direct および Sterling Connect:Direct for z/OS でのみ利用可能)

cdsacomp はオフラインで使用できることから、圧縮に伴うオーバーヘッドの一部をピーク外の時間に割り当てることができます。例えば、同一ファイルを複数のリモート・ノードに送信する必要がある場合、このユーティリティを使用することで、ファイルが一度のみ事前圧縮されるようにすることができます。cdsacomp を使用することで、ファイルを伝送することなく、そのファイルにおい

スタンドアロン型バッチ圧縮ユーティリティを使用した、ファイルの事前圧縮/圧縮解除
てどのくらいの圧縮を行うことができるかを決定することもできます。

cdsacomp ユーティリティは、Sterling Connect:Direct /bin ディレクトリーに配置されています。

スタンドアロン型バッチ圧縮ユーティリティの使用における特別な考慮事項

cdsacomp を使用してファイルを事前圧縮する際には、以下を考慮してください。

- ◆ cdsacomp ユーティリティを使用してファイルを事前圧縮する場合、そのファイルをコピーする際に、Sterling Connect:Direct Process で圧縮オプションを指定することはできません。
- ◆ オンザフライでの圧縮解除のために :precompress=yes: sysopts を使用して、事前圧縮されたファイルを送信する際には、データ変換 (xlate、codepage、strip blanks など) を指定することはできません。以下の変換オプションを使用することができます。これらのオプションは、以下のパラメーター表に記述されています。
 - -x
 - -p
 - -s
 - -a
- ◆ Sterling Connect:Direct ノード上で cdsacomp ユーティリティを使用してファイルを事前圧縮し、リモート・ノードによるファイルの受信時にそのファイルを圧縮解除する場合には、Sterling Connect:Direct Process でチェックポイント間隔を指定することはできません。
- ◆ 延期された圧縮解除のために :precomp=yes: を指定せずに、事前圧縮されたファイルを z/OS にコピーする場合:
 - コピー操作において、宛先ファイルの DCB 情報を指定する必要があります。Sterling Connect:Direct for z/OS 上の宛先ファイルの物理ブロック・サイズは、Sterling Connect:Direct for UNIX 上の事前圧縮されたソース・ファイルの論理ブロック・サイズと一致する必要があります。
 - ソース・ファイルの論理ブロック・サイズは、-b パラメーターによってオーバーライドされない限り、27920 にデフォルト設定されます。

スタンドアロン型バッチ圧縮ユーティリティの使用

スタンドアロン型バッチ圧縮ユーティリティ (cgsacomp) を呼び出すには、UNIX プロンプトで以下のコマンドを入力します。

```
cgsacomp
```

以下は、cgsacomp ユーティリティのパラメーターです。

パラメーター	説明	値
-m	事前圧縮または圧縮解除のどちらのモードを使用するかを指定します。この引数は必須です。	<u>compress</u> decompress デフォルトは、 compress です。
-i	事前圧縮または圧縮解除する入力ファイルを指定します。この引数は必須です。	入力ファイルの完全パスまたは相対パス。

パラメーター	説明	値
-o	保存する出力ファイルを指定します。出力ファイルが既に存在する場合は、上書きされます。この引数は必須です。	出力ファイルの完全パスまたは相対パス。
-z	このオプションに「-m compress」と指定して、デフォルトの圧縮値をオーバーライドします。この引数はオプションです。 圧縮解除の際には、圧縮時に使用された値が使用されます。	レベル, ウィンドウ, メモリー レベル - 圧縮レベル。範囲は 1 ~ 9 です。デフォルトは 1 です。 1 - 最小レベルの圧縮が実行されるものの、最も迅速に実行されます。 9 - 最大レベルの圧縮が実行されるものの、最も長い時間がかかります。 ウィンドウ - 圧縮ウィンドウおよびヒストリー・バッファーのサイズ。ウィンドウを増やすことで、圧縮レベルも高まりますが、より多くの仮想メモリーを使用します。範囲は 9 ~ 15 です。デフォルトは 13 です。 メモリー - 割り当てる仮想メモリーの量。範囲は 1 ~ 9 です。デフォルトは 4 です 1 - 最小仮想メモリー使用量。 9 - 最大仮想メモリー使用量。
-x	このオプションを使用して、ファイルを変換します。 このオプションを使用して、ファイルを変換します。 このパラメーターは、-codepage と同時に指定することはできません。	変換テーブル・ファイルへの完全パス 変換テーブル・ファイルへの相対パス
-p	このオプションを使用して、ファイル変換用のコード・ページを指定します。デフォルトでは、コード・ページ変換の指定はありません。 このパラメーターは、-xlate と同時に指定することはできません。	ソース・コード・ページ、 宛先コード・ページ

パラメーター	説明	値
-d	<p>ファイルの datatype を指定します</p> <p>「-m compress」を使用すると、datatype の値は、以下になります。</p> <ul style="list-style-type: none"> ◆ text <ul style="list-style-type: none"> ● 各レコードから改行文字を削除します。 ● -s および -a パラメーターをサポートします。 ● blocksize=23040、recfm=vb、lrecl=23036、および dsorg=ps のデータ属性を使用します。 ◆ binary <ul style="list-style-type: none"> ● blocksize=23040、recfm=u、lrecl=0、および dsorg=ps のデータ属性を使用します。 ● -s および -a パラメーターをサポートしません。 ◆ VB <ul style="list-style-type: none"> ● -x、-p、-s および -a パラメーターをサポートしません。 ● blocksize=23040、recfm=vb、lrecl=23036、および dsorg=ps のデータ属性を使用します。 <p>「-m decompress」を使用すると、datatype の値は、以下になります。</p> <ul style="list-style-type: none"> ◆ text <ul style="list-style-type: none"> ● 各レコードに改行文字を挿入します。 ● -s パラメーターをサポートします。 ◆ binary <ul style="list-style-type: none"> ● -s パラメーターをサポートしません。 ◆ VB <ul style="list-style-type: none"> ● -x、-p、および -s パラメーターをサポートしません。 	<p><u>text</u> binary VB</p> <p>デフォルトは text です。</p>
-b	<p>出力ファイルのブロック・サイズを指定します。</p> <p>このパラメーターは、圧縮オプションに「-m compress」を指定した場合のみ有効です。</p>	<p><i>nnnnn</i></p> <p>範囲は 4096 ~ 32760 です。デフォルトは 27920 です。</p>
-s	<p>このオプションを使用して、末尾空白を削除します。このパラメーターは、ファイルの datatype に「-d text」を指定した場合のみ有効です。</p>	<p><u>y</u> n</p> <p>y - yes n - no</p> <p>デフォルトは y です。</p>

パラメーター	説明	値
-a	このオプションを使用して、ゼロ長のレコードを単一の空白文字に置き換えます。 このパラメーターは、「-d text」および「-m compress」を指定した場合のみ有効です。	y n y - yes n - no デフォルトは y です。 データが i5OS またはメインフレーム・ノードにコピーされる場合には、n を指定します。
-h	このオプションを使用して、このユーティリティのオンライン・ヘルプを表示します。	このパラメーターに利用できる値はありません。

例: テキスト・ファイルの事前圧縮

この例では、ソース・ファイルは、compressed.file という名の宛先ファイルに事前圧縮された、source.file という名のテキスト・ファイルです。このファイルは、デフォルトの変換テーブル /home/cd/ndm/xlate/def_send.xlt を使用して変換されます。末尾空白は削除されます。ZLIB チューニング、チェックポイント間隔、およびブロック・サイズのデフォルトの設定が使用されます。

```
cdsacomp -m compress
-d text
-i source.file
-o compressed.file
-x/home/cd/ndm/xlate/def_send.xlt
-s y
```

例: コード・ページ変換を使用したテキスト・ファイルの事前圧縮

この例では、ソース・ファイルは、zzz.txt という名のファイルに事前圧縮された、zzz.sac という名のテキスト・ファイルです。このファイルは、コード・ページ・オプションを使用して、EBCDIC-US から ASCII に変換されます。指定のないパラメーターに対しては、デフォルトの設定が使用されます。

```
cdsacomp -m compress
-d text
-i zzz.txt
-o zzz.sac
-p EBCDIC-US,ASCII
```

例: バイナリー・ファイルの事前圧縮

この例では、ソース・ファイルは、compressed.file という名の宛先ファイルに事前圧縮された、source.file という名のバイナリー・ファイルです。指定のないパラメーターに対しては、デフォルトの設定が使用されます。

```
cdsacomp -m compress
          -d binary
          -infile source.file
          -outfile compressed.file
```

例: バイナリー・ファイルの圧縮解除

この例では、ソース・ファイルは、dest.file という名の宛先ファイルに圧縮解除された、compressed.file という名の事前圧縮されたテキスト・ファイルです。このファイルは、デフォルト変換テーブル /home/cd/ndm/xlate/def_recv.xlt を使用して変換されます。指定のないパラメーターに対しては、デフォルトの設定が使用されます。

```
cdsacomp -m decompress
          -d text
          -i compressed.file
          -o dest.file
          -x/home/cd/ndm/xlate/def_recv.xlt
```

例: cdsacomp コマンドのヘルプ

cdsacomp コマンド・パラメーターおよびヘルプ・オプションのサマリーを要求します。

```
cdsacomp -h
```

例: Copy ステップ実行時のリモート・ノードでのファイルの圧縮解除

この Process の前に、cdsacomp ユーティリティによってファイルが圧縮される場合には、“precomp=yes” パラメーターを使用します。ファイルは、このプロセスによって、事前圧縮されたファイルとして転送された後、リモート・ノードで受信された際に、特別な処理によって圧縮解除されます。

```
sample process snode=cdunix1

step01 copy
from
(
file=/home/cd/upload/compressed.file
sysopts=":precomp=yes:"
pnode
)

to
(
file=/home/cd/download/decompressed.file
snode
disp=rpl
)

pend;
```

例: 事前圧縮されたファイルの z/OS への送信および事前圧縮済みとしての保存

事前圧縮されたファイルは、“datatype=binary” の PNODE sysopts を使用して、z/OS ノードにコピーされます。宛先ファイルは圧縮解除されません。事前圧縮された元ファイルの DCB 設定は、z/OS ノード上に保存されます。指定されたチェックポイント間隔がファイル転送時に使用されます。このファイルは、z/OS cdsacomp ユーティリティによって圧縮解除することができます。

```
sample process snode=cdunix1

step01 copy
from
(
file=/home/cd/upload/compressed.file
sysopts=":datatype=binary:"
pnode
)

chkpt=2M

to
(
file=upload.compressed.file
dcb=(blksize=27920,lrecl=0,dsorg=ps,recfm=u)
snode
disp=(new,catlg)
)

pend;
```

構成ファイルの検証

テキストベースの 5 つの Sterling Connect:Direct 構成ファイルのいずれかを手動で編集する際に、構成検査ユーティリティ (cfgCheck) によって、これらのファイルをオフラインで検証することができます。このユーティリティを使用して、userfile.cfg、initparm.cfg、netmap.cfg、ndmapi.cfg、および sysacl.cfg のファイルを検証することができます。

注: 強固なアクセス制御ファイル (sysacl.cfg) は、構成検査ユーティリティを実行するユーザーが root ユーザーである場合のみ、検証されます。

デフォルトでは、cfgCheck は、引数を指定せずに実行され、現行作業ディレクトリーにおいて、5 つの構成ファイルすべての検索を試みます。Sterling Connect:Direct コンポーネントのすべてがインストールされていない場合には、検出されないファイルもあります。例えば、コマンドライン・インターフェース (CLI) がインストールされているものの、Sterling Connect:Direct サーバーがインストールされていない場合、ndmapi.cfg ファイルのみがインストール・ディレクトリーに存在することになります。そのため、ndmapi.cfg ファイルのみが検証されます。引数を使用せずに cfgCheck を実行した場合、このユーティリティは、他の構成ファイルが検出されなかったことをレポートします。

注: cfgCheck を実行する前に、NDMAPICFG 環境変数を設定する必要があります。詳細については、18 ページの『CLI の起動』を参照してください。

cfgCheck を呼び出すには、UNIX プロンプトで以下のコマンドを入力します。

```
$ cfgCheck -v -h -f filename.cfg
```

以下の表は、cfgCheck コマンドの引数を示しています。

引数	説明
引数の指定なし (デフォルト)	引数の指定なしに、非 root ユーザーによって cfgCheck が実行される場合、このユーティリティは cfg/ ディレクトリー内で initparm.cfg、netmap.cfg、userfile.cfg、および ndmapi.cfg を検索します。root ユーザーによって cfgCheck が実行される場合には、このユーティリティは、sysacl.cfg ファイルを見つけるために、SACL/ ディレクトリーも検索します。
-h	ヘルプ画面を出力して、終了します。
-t	トレースを有効にして、詳細なデバッグ情報を出力します。
-f filename.cfg	検証する構成ファイル名を指定します。filename は、構成ファイルのうちの 1 つの構成ファイルの名前です。複数の -f 引数を指定することができます。-f 引数が使用されると、cfgCheck が、指定されたファイル以外の構成ファイルを自動的に検索することはありません。 注: sysacl.cfg ファイルを検証するには、root アクセスまたは root の有効な権限を有している必要があります。

構成レポートの生成

構成レポート・ユーティリティ (cdcustrpt) を使用して、システム情報および Sterling Connect:Direct の構成情報のレポートを生成することができます。以下の Sterling Connect:Direct コンポーネントに関する構成レポートの生成が可能です。

- ◆ Sterling Connect:Direct の基本インストール
- ◆ Sterling Connect:Direct Secure Plus for UNIX
- ◆ Sterling Connect:Direct for SWIFTNet for UNIX

cdcustrpt は、Sterling Connect:Direct のインストール時に、<installation>/etc/ ディレクトリーにインストールされます。

基本インストールに関するレポート

cdcustrpt を使用して、Sterling Connect:Direct の基本インストールに関するレポートを生成する場合、以下のタイプのシステム情報がレポートされます。

- ◆ オペレーティング・システムの名前およびその他の情報
- ◆ ファイル・システム上の容量
- ◆ 仮想メモリー統計
- ◆ Sterling Connect:Direct のインストール・ディレクトリーの内容

システム情報のレポートに加えて、cdcustrpt は、構成検査ユーティリティ (cfgCheck) を呼び出して、テキスト・ベースの 5 つの構成ファイルの構文を検証し (これらのファイルが利用可能であり、ユーザーがこれらのファイルに対するアクセス権を保持している場合)、構成ファイルの内容についてレポートします。cfgCheck に関する詳細については、77 ページの『構成ファイルの検証』を参照してください。

cdcustrpt を呼び出して、基本インストールに関するレポートを生成する手順:

1. UNIX プロンプトで、以下のコマンドを入力します。

```
$ cdcustrpt
```

注: デフォルト値は、このユーティリティによって、インストールされている Sterling Connect:Direct のロケーションおよび名前に基づいて計算され、[] 内に表示されます。Enter を押して、デフォルト値を受け入れます。

2. Sterling Connect:Direct がインストールされているロケーションの完全パスを入力して、Enter を押します。
3. 生成されるレポートの完全パスおよび名前を入力して、Enter を押します。

4. 指定したロケーションにレポートが生成され、以下の例に示されているように、すべてのエラー・メッセージが表示されます。

```
% cdcustrpt
Enter full path of Connect:Direct destination
directory:[/sci/users/jbrown1/cd40]:
Enter full path and name for this support report
file:[/sci/users/jbrown1/cd40/etc/cd.support.rpt]:
ls: /sci/users/jbrown1/cd40/ndm/SACL: Permission denied cdcustrpt ended
```

この例では、ユーザーが root アクセス権を保持していないため、強固なアクセス制御ファイル (sysacl.cfg) にアクセスすることはできません。以下の例は、サンプル・レポートからの抜粋を示しています。

```
#####
#####      Connect:Direct for UNIX 4.0.00 configuration report      #####
#####
Connect:Direct for UNIX Version 4000, Build 00, IBM/RS6000 AIX, Fix date: 01OCT2007
Install directory: /sci/users/jbrown1/cd40
Local Node name: jb_aix40

Report for: jbrown1

=====
=====      Begin: Environment and system information      =====
=====

System: AIX skyglass 3 5 00CE208E4C00

Disk usage:
Filesystem      512-blocks      Free %Used      lused %lused Mounted on
/dev/hd4          262144          64216 76%         2479  4% /
/dev/hd2          8126464         2708688 67%         37802 4% /usr
/dev/hd9var       262144          18448 93%          613  2% /var
/dev/hd3          786432          363600 54%          424  1% /tmp
/dev/fwdump       524288          507752  4%           17  1% /var/adm/ras/platform
/dev/hd1          262144          216520 18%          167  1% /localhome
/proc              -                -      -            -    - /proc
/dev/hd10opt      524288          52168 91%          3688 6% /opt
/dev/fslv00       121634816       13629040 89%         264984 15% /sci
scidalnis01:/export/nis01 1677670392 512499192              70%          0  -1% /home/nis01

Memory statistics:
System Configuration: lcpu=4 mem=3824MB

kthr      memory              page              faults              cpu
-----
 r b   avm  fre re pi po fr          sr cy in          sy cs us sy id wa
1 1 400072 232777          0  0  0  0          0  0  0  4 1805 197 0 1 99 0
```

Sterling Connect:Direct Secure Plus for UNIX に関するレポートニング

cdcustrpt は、インストール・ディレクトリー内に、Sterling Connect:Direct Secure Plus ディレクトリー <installation>/ndm/secure+/ を検知した場合、Sterling Connect:Direct Secure Plus コマンド・ライン・ユーティリティー (splicli.sh) を呼び出して、Secure+ パラメーターに関してレポートします。Sterling Connect:Direct Secure Plus が検知されると、Sterling Connect:Direct Secure Plus パラメーター・ファイルに対するパスの入力が求められます (デフォルトのロケーションが、[] 内に表示されます)。以下は、その例です。

```
Enter full path of Secure+ parmfile
directory:[/sci/users/jbrown1/cd40/ndm/secure+/nodes]:
```

以下の例は、サンプル・レポートからの抜粋を示しています。

```
===== Begin: Secure+ parameters =====
=====

All secure+ nodes:
*****
*                Secure+ Command Line Interface                *
*                Connect:Direct for UNIX v4.0.00                *
*-----*
* Copyright (c) 1999, 2008 Sterling Commerce Inc.                *
* All Rights Reserved.                                           *
*****

SPCLI> display all;

name=.Local
baserecord=brown_aix38
type=l
protocol=tls
override=n authtimeout=120
stsenablesig=n
stsenableautoupdate=n
stslimitexportversion=y
stsenableenc=y
stsencalgs=(ideacbc128,tdescbc112,descbc56)
stsauthlocalkey=0305.095A.44E3.BD87.F476.45E8.09B1.FCCA.45ED.67B0.01AD
stsprevauthkeyexpdatetime=
stssiglocalkey=0204.BABA.613D.2FA5.AAE6.0BD4.5847.B610.A17F.C7DD.0AA2
stsprevsigkeyexpdatetime=
ssltsseaenable=n
seacertvaldef=
ssltstrustedrootcertfile=/home/nis01/jbrown1/CertificateWizard/cert.crt
ssltscertfile=/home/nis01/jbrown1/CertificateWizard/athena.selfsigned.keycert.txt
ssltsenablefipsmode=n
ssltsenableclientauth=n
ssltsenablecipher=(TLS_RSA_WITH_AES_256_CBC_SHA,TLS_RSA_WITH_AES_128_CBC_SHA)

2007/10/19 14:27:37 parmfile upgraded: SPV4
2007/03/27 09:25:14 jbrown1
2007/03/22 09:54:55 jbrown1
```

Sterling Connect:Direct for SWIFTNet for UNIX に関するレポートिंग

cdcustrpt は、インストール・ディレクトリー内に、SWIFTNet ディレクトリー <installation>/ndm/SwiftNet/を検知した場合、CDSwiftnet.cfg ファイルの内容をレポートに含めます。パスワード・パラメーターの値は、アスタリスク (*) 列に置き換えられます。

以下の例は、サンプル・レポートからの抜粋を示しています。

```

=====
Begin: /sci/users/jbrown1/swift31/ndm/SwiftNet/Version3/cfg/CDSwiftnet.cfg
=====

==== Content of
/sci/users/jbrown1/swift31/ndm/SwiftNet/Version3/cfg/CDSwiftnet.cfg =====
o
# Connect:Direct UNIX for SWIFTNet 3.1.00 configuration file.
#

[Directory.Info] CD.HomeDir="/sci/users/jbrown1/swift31"
CDSwiftnet.HomeDir="/sci/users/jbrown1/swift31/ndm/SwiftNet/Version3"

# Concatenate the RequestorDN and ResponderDN to these directories for the Request
Handler.
Reception.Dir="/sci/users/jbrown1/reception"
Download.Dir="/sci/users/jbrown1/download"

# This directory must be specified to use the #OLDEST_FILE feature.
Success.Dir="/sci/users/jbrown1/success"

[Log.Info]
#Log.MaxSize="1048576"
Log.MaxSize="35000"
Log.MaxVersions="5"

[connection.info]
#Connection information for Connect:Direct's API port. (Used when forwarding files to the back office.)
Comm.Info="spyglass;10102"
Userid="jbrown1" Passwd="*****"# this is a
test
#ClientInfo="/sci/users/jbrown1/swift31/ndm/SwiftNet/Version3/program/<Encrypted
userid/password file generated by the LCU>"

```

カスタム・プログラムの作成

Sterling Connect:Direct アプリケーション・プログラミング・インターフェース (API) によって、Sterling Connect:Direct で使用するカスタム・プログラムを、C または C++ で作成することができます。C 関数または C++ クラスを使用することで、以下のタスクを実行するためのプログラムを作成することができます。

- ◆ Sterling Connect:Direct サーバーへの接続の確立
- ◆ サーバーからの切断
- ◆ サーバーからのコマンド応答の受信
- ◆ サーバーへのコマンドの送信

この章では、Sterling Connect:Direct API の関数およびクラスのフォーマットを説明するとともに、その使用におけるサンプルを提供します。Sterling Connect:Direct API の関数およびクラスを使用して、コマンドの実行および Sterling Connect:Direct サーバーからの応答の受信を行うサンプル・プログラムが提供されています。

カスタム・プログラムのコンパイル

カスタム・プログラムを作成後、C コンパイラーまたは C++ コンパイラーを使用してコンパイルする必要があります。各プラットフォームに使用する最小の C++ コンパイラーのバージョンを特定するには、以下の表を参照してください。

プラットフォーム	C++ コンパイラー
AIX	IBM XL C++ for AIX バージョン 8.0
Sun Solaris	SPARC/x86 C++5.7
HP	aCC: HP ANSI C++ B3910B A.03.73
HP-Itanium	aCC: HP ANSI C++ B3910B A.06.07
Linux	c++ バージョン 3.3.3

C++ API 呼び出しを使用したカスタム C++ プログラムをコンパイルするには、以下の表に定義されたコマンドを使用します。

プラットフォーム	C++コンパイル・コマンド
AIX	
32-bit	<code>/usr/vacpp/bin/xlC -qinline -I../include ++ -o sdksample sdksample.C ../lib/ndmapi.a -lbsd -ldl -lsrc -lpthreads</code>
64-bit	<code>/usr/vacpp/bin/xlC-q64 -qinline -I../include ++ -o sdksample sdksample.C ../lib/ndmapi64.a -lbsd -ldl -lsrc -lpthreads</code>
Sun	
32-bit	<code>/opt/SUNWspro/bin/CC -DBSD_COMP -I../include -o sdksample sdksample.C ../lib/ndmapi.a -L/usr/ucblib -L/usr/lib -lsocket -lrpcsoc -lnsl -lelf -ldl</code> 注: LD_LIBRARY_PATH 変数内に /usr/ucblib がない場合は、-R/usr/ucblib をコンパイル・コマンドに追加します。
64-bit	<code>/opt/SUNWspro/bin/CC -xarch=generic64 -DBSD_COMP -I../include -o sdksample sdksample.C ../lib/ndmapi64.a -L/usr/ucblib/sparcv9 -L/usr/lib/sparcv9 -L/usr/ucblib/amd64 -lsocket -lrpcsoc -lnsl -lelf -ldl -R/usr/ucblib/sparcv9 -R/usr/ucblib/amd64</code>
HP	
32-bit	<code>/opt/aCC/bin/aCC -AA -I../include -o sdksample sdksample.C ../lib/ndmapi.a -lrpcsoc -lnsl -ldld -Wl,+s</code>
64-bit	<code>/opt/aCC/bin/aCC -AA +DD64 -I../include -o sdksample sdksample.C ../lib/ndmapi64.a -L/usr/lib/pa20_64 -lnsl -ldld -Wl,+s</code>
HP-Itanium	
32-bit	<code>/opt/aCC/bin/aCC -I../include -o sdksamplesdksample.C ../lib/ndmapi.a -lrpcsoc -lnsl -ldld -Wl,+s -lunwind</code>
64-bit	<code>/opt/aCC/bin/aCC +DD64 -I../include -o sdksample sdksample.C ../lib/ndmapi64.a -L/usr/lib/hpux64 -lrpcsvc -lnsl -ldld -Wl,+s -lunwind</code>
Linux	
32-bit	<code>g++ -m32 -I../include -O -DLINUX -o sdksample sdksample.C ../lib/ndmapi.a -ldl -lnss_nis /usr/lib/libstdc++.so.5</code> 注: Linux z/OS 向けにコンパイルする場合には、-m32 を -m31 に変更します。
64-bit	<code>g++ -I../include -O -DLINUX -o sdksample sdksample.C ../lib/ndmapi64.a -ldl -lnss_nis /usr/lib64/libstdc++.so.5</code> 注: 64 ビットのバイナリーをコンパイルするには、64 ビットの Linux OS をインストールする必要があります。

C API 呼び出しを使用したサンプル・プログラムである apicheck.C といった C++ プログラムを構築するには、sdksample.C パラメーターを、C++ プログラムの名前に置き換え、出力ファイル・パラメーター -o sdksample を apicheck といった作成したい出力ファイル名に変更します。

C プログラムをコンパイルするには、以下の表に定義されたコマンドを使用します。

プラットフォーム	Cコンパイル・コマンド
AIX	
32-bit	<code>/usr/vacpp/bin/xlc -I../include ++ -o apicheck apicheck.c ../lib/ndmapi.a -lbsd -ldl -lsrc -lC -lpthreads</code>
64-bit	<code>/usr/vacpp/bin/xlc -q64 -I../include ++ -o apicheck apicheck.c ../lib/ndmapi64.a -lbsd -ldl -lsrc -lC -lpthreads</code>
Sun	
32-bit	<code>/opt/SUNWspro/bin/cc -DBSD_COMP -I../include -o apicheck apicheck.c ../lib/ndmapi.a -L/usr/ucblib -L/usr/lib -lCstd -lsocket -lrpcsoc -lnsl -lelf -ldl -lCrun</code> 注: LD_LIBRARY_PATH 変数 内に /usr/ucblib がいない場合は、-R/usr/ucblib をコンパイル・コマンドに追加します。
64-bit	<code>/opt/SUNWspro/bin/cc -xarch=generic64-DBSD_COMP -I../include -o apicheck apicheck.c ../lib/ndmapi64.a -L/usr/ucblib/sparcv9 -L/usr/lib/sparcv9 -L/usr/ucblib/amd64 -lsocket -lCstd -lCrun -lrpcsoc -lnsl -lelf -ldl -lCrun -R/usr/ucblib/sparcv9 -R/usr/ucblib/amd64</code>
HP	
32-bit	<code>/opt/ansic/bin/cc -I../include -o apicheck apicheck.c ../lib/ndmapi.a -lrpcsoc -lnsl -ldld -lWI,+s -lcl -lstdd_v2 -lCsup_v2</code>
64-bit	<code>/opt/ansic/bin/cc +DD64 -I../include -o apicheck apicheck.c ../lib/ndmapi64.a -L/usr/lib/pa20_64 -lnsl -ldld -lWI,+s -lcl -lstdd_v2 -lCsup_v2</code>
HP-Itanium	
32-bit	<code>/opt/ansic/bin/cc -I../include -o apicheck apicheck.c ../lib/ndmapi.a -lrpcsoc -lnsl -ldld -lWI,+s -lcl -lstdd_v2 -lCsup -lunwind</code>
64-bit	<code>/opt/ansic/bin/cc +DD64 -I../include -o apicheck apicheck.c ../lib/ndmapi64.a -L/usr/lib/hpux64 -lrpcsvc -lnsl -ldld -lWI,+s -lcl -lstdd_v2 -lCsup -lunwind</code>
Linux	
32-bit	<code>gcc -m32 -I../include -O -DLINUX -o apicheck apicheck.c ../lib/ndmapi.a -ldl -lnss_nis /usr/lib/libstdc++.so.5</code>
64-bit	<code>gcc -I../include -O -DLINUX -o apicheck apicheck.c ../lib/ndmapi64.a -ldl -lnss_nis /usr/lib64/libstdc++.so.5</code> 注: 64 ビットのバイナリーをコンパイルするには、64 ビットの Linux OS をインストールする必要があります。
LinuxS390	
32-bit	<code>gcc -m31 -I../include -O -DLINUX -o apicheck apicheck.c ../lib/ndmapi.a -ldl -lnss_nis /usr/lib/libstdc++.so.5</code>
64-bit	<code>gcc -I../include -O -DLINUX -o apicheck apicheck.c ../lib/ndmapi64.a -ldl -lnss_nis /usr/lib64/libstdc++.so.5</code> 注: 64 ビットのバイナリーをコンパイルするには、64 ビットの Linux OS をインストールする必要があります。

カスタム C プログラムの作成

C API 呼び出しを使用したカスタム・プログラムを作成する場合、ヘッダー・ファイル `ndmapi.h` をインクルードするとともに、そのファイルを `ndmapi.a` にリンクする必要があります。 `apicheck.c` と呼ばれるサンプル・プログラムが提供されています。

Java プログラミングにおいては、JNI および `libndmapi` 共有オブジェクト (HP の場合は `libndmapi.sl`、その他の対応プラットフォームの場合は `libndmapi.so`) を使用して C API 関数を呼び出すことができます。

注: JNI がサポートされていますが、Sterling Connect:Direct のサービス呼び出す Java プログラムには、Sterling Connect:Direct Java アプリケーション・インターフェースが推奨されています。

注: クライアント構成ファイルの `pathname` に `NDMAPICFG` の環境変数が設定される必要があります。環境変数の設定手順については、18 ページの『CLI の起動』を参照してください。

C プログラムおよび C++ プログラムにおいては、以下の Sterling Connect:Direct API 関数を使用します。

C++ 関数	C 関数	説明
<code>ndmapi_connect()</code>	<code>ndmapi_connect_c()</code>	サーバーとの接続を確立します。 <code>ndm_nodspec</code> ポインターまたは CLI/API 構成情報ファイル内で、接続するノードを指定します。呼び出しが正常に行われると、 <code>NDM_NO_ERROR</code> が返されます。接続が確立されて、最初の API 要求に回答できる状態になると、制御がアプリケーションに戻されます。
<code>ndmapi_sendcmd()</code>	<code>ndmapi_sendcmd_c()</code>	Sterling Connect:Direct にコマンドを送信します。コマンド・テキストを入力する必要があります。 <code>resp_moreflag</code> は、実行されたコマンドにおいて、応答がさらに保留されていることを示すフラグに対するポインターです。C プログラムの場合は <code>ndmapi_recvresp_c()</code> 、C++ プログラムの場合は <code>ndmapi_recvresp()</code> を呼び出して、さらなる応答を取得します。 <code>select process</code> および <code>select statistics</code> コマンドにおいてのみ、C を使用する場合には <code>ndmapi_recvresp_c()</code> 、C++ をする場合には <code>ndmapi_recvresp()</code> を使用する必要があります。
<code>ndmapi_recvresp()</code>	<code>ndmapi_recvresp_c()</code>	Sterling Connect:Direct に送信されたコマンドに対する応答を受信します。応答バッファ内容が返されます。
<code>ndmapi_disconnect()</code>	<code>ndmapi_disconnect_c()</code>	API 接続を終了します。

これらの関数は、以下の 3 種の Sterling Connect:Direct コマンド応答を返します。

- ◆ これらの関数は、以下の 3 種の Sterling Connect:Direct コマンドに関する情報を返します。
- ◆ データ応答: `resp_buffer` に保存されており、データ・レコードを含みます。
- ◆ エラー応答: 検出された全エラーのリンク・リストに対するポインターである、 `ERROR_H` を返します。ID フィールド値は、デバッグでの使用のために固定されています。 `msgid`、 `feedback`、および `rc` フィールドは、Sterling Connect:Direct によって指定され、メッセージ・

テキスト内で参照されます。subst フィールドは、メッセージ・テキスト内に適切に挿入される置換変数情報を含むストリングを指します。

エラー制御構造体によって、現在のエラー数およびエラー数の合計の記録がとられます。エラー・エントリー・ブロック内で next ポインターを使用することによって、エラー間を移動することができます

以下のコードは、ERROR_H 構造体を定義しています。

```
#define NDM_ERR_ENT_T struct NDM_ERR_ENT_S
#define NDM_ERR_ENT_H NDM_ERR_ENT_T *
#define NDM_ERR_CTL_T struct NDM_ERR_CTL_S
#define ERROR_H          NDM_ERR_CTL_T
* struct NDM_ERR_ENT_S
{
    int32    id;
    char     msgid[MSGIDLEN];
    int32    feedback;
    int32    rc;
    char     *subst;
    NDM_ERR_ENT_H next;
};
struct NDM_ERR_CTL_S
{
    int32    id;
    int32    cur_entry; int32
            num_entries;
    NDM_ERR_ENT_H next;
};
```

ndmapi_connect() または ndmapi_connect_c() を使用した、Sterling Connect:Direct への接続の作成

アプリケーションによるコマンドの送信およびコマンドからの応答の受信を可能にするには、**ndmapi_connect()** または **ndmapi_connect_c()** を使用して、Sterling Connect:Direct に対する接続を作成します。接続が確立されて、Sterling Connect:Direct が最初の API 要求に応答できる状態となるか、あるいはエラー条件が設定されると、制御がアプリケーションに戻されます。

以下は、**ndmapi_connect()** 関数または **ndmapi_connect_c()** 関数のフォーマットを示しています。

```
int32 ndmapi_connect ERROR_H error, char * ndm_hostname, char * ndm_portname
```

以下の表は、**ndmapi_connect()** 関数または **ndmapi_connect_c()** 関数のパラメーターを説明しています。

パラメーター	説明	値
error	エラー情報またはステータス情報を含む、Sterling Connect:Direct によって定義された構造体に対するポインター。	ポインター
ndm_hostname	接続を確立する Sterling Connect:Direct ホストのテキスト指定に対するポインター。このパラメーターが指定されていない場合、ホスト名は、まず環境変数の TCPHOST をチェックすることによって決定されます。値が指定されていない場合は、CLI/API 構成ファイル内の host.name フィールドをチェックしますが、値の指定がない場合は、 gethostbyname() コマンドが呼び出され、デフォルト値の ndmhost が使用されます。	ヌル終了ストリング

パラメーター	説明	値
ndm_portname	ホスト・ポート番号に対するポインター。このパラメーターが指定されていない場合、環境変数 TCPSPORT がチェックされます。値が指定されていない場合は、CLI/API 構成ファイル内の tcp.port の値がチェックされますが、値の指定がない場合、デフォルト値の 1363 が使用されます。	ポインター

以下は、**ndmapi_connect()** 関数または **ndmapi_connect_c()** 関数の戻りコードを示しています。

戻りコード	説明
NDM_NO_ERROR	サーバーとのセッションが確立されたことを示します。
NDM_ERROR	サーバーとのセッションが確立されなかったことを示します。エラー・ステータスの詳細については、エラー構造体を確認してください。

以下のサンプル関数は、**ndmapi_connect()** を使用した、sun1 ホストへの接続を示しています。

```
rc=ndmapi_connect(error, "sun1", "3122");
```

ndmapi_disconnect() または **ndmapi_disconnect_c()** を使用した、接続の終了

ndmapi_connect() または **ndmapi_connect_c()** の呼び出しによって確立された Sterling Connect:Direct への接続を終了するには、**ndmapi_disconnect()** または **ndmapi_disconnect_c()** を使用します。以下は、**ndmapi_disconnect()** または **ndmapi_disconnect_c()** のフォーマットを示しています。

```
void ndmapi_disconnect
```

ndmapi_disconnect() または **ndmapi_disconnect_c()** においては、パラメーターおよび戻りコードは存在しません。以下は、**ndmapi_disconnect()** 関数のサンプルです。

```
ndmapi_disconnect();
```

ndmapi_recvresp() または **ndmapi_recvresp_c()** を使用した、応答の受信

アプリケーションから送信された前のコマンドに関連付けられている応答を受信するには、**ndmapi_recvresp()** または **ndmapi_recvresp_c()** を使用します。以下は、**ndmapi_recvresp()** または **ndmapi_recvresp_c()** のフォーマットです。

```
int32 ndmapi_recvresp ERROR_H error int32 * resp_length, char * resp_buffer, int32 * resp_moreflag
```

以下は、`ndmapi_recvresp()` または `ndmapi_recvresp_c()` のパラメーターです。

パラメーター	説明	値
<code>error</code>	エラー情報またはステータス情報を含む、Sterling Connect:Direct によって定義された構造体に対するポインター。	ポインター
<code>resp_length</code>	応答を受信するためのアプリケーション・バッファの長さ (バイト) に対するポインター。API は、返されたバイト数をこのパラメーターに設定します。	返されたバイト数、またはゼロ (応答の受信を停止したい場合) に対するポインター。このフィールドをゼロに設定すると、キューにある すべての応答がパーシされます。
<code>resp_buffer</code>	<p>コマンドまたは実行依頼の応答を受信するアプリケーション・バッファに対するポインター。このバッファには、4096 バイトが割り当てられる必要があります。</p> <p>resp_buffer のフォーマットは、フリー・フォーム・テキスト・レコード構造体です。フィールド名は、4 文字長で、すべて大文字です。データの長さに制限はなく、ブランクを含めることができます。構造体は以下のとおりです。</p> <pre>field name=data field name=data ...</pre> <p>For example: SUBM = username PNUM = 12 PNAME = proc1 ...</p>	<p>ndmapi_recvresp() または ndmapi_recvresp_c() によって入力された、<code>resp_length</code> による設定の値以上のサイズを持つローカル・バッファ。</p> <p>CLI は、<code>resp_buffer</code> を AWK に渡して解析を行います。有効な値には以下が含まれます。</p> <p>ADMN - Sterling Connect:Direct 管理者の名前</p> <p>ADPH - Sterling Connect:Direct 管理者の電話番号</p> <p>CCOD - 完了コード</p> <p>CKPT - チェックポイント</p> <p>CLAS - クラス</p> <p>DBYW - 書き込まれたバイト</p> <p>DBYX - 受信されたバイト</p> <p>DCOD - 宛先完了コード</p> <p>DDAY - 実行依頼日</p> <p>DDS1 - 宛先の処理 1</p> <p>DDS2 - 宛先の処理 2</p> <p>DDS3 - 宛先の処理 3</p> <p>DESC - Sterling Connect:Direct 管理者の説明</p> <p>DFIL - 宛先ファイル</p> <p>DMSG - 宛先メッセージ ID</p> <p>DNVL - ポリュームの宛先番号</p> <p>DRCW - 書き込まれたレコード</p> <p>DRUX - 受信された RU</p> <p>DVOL - 宛先ポリューム配列</p> <p>ECMP - 拡張圧縮 ON または OFF</p>

パラメーター	説明	値
resp_buffer (続)		FROM - コピー送信ノード LCCD - ローカル完了コード LCLP - ローカル IP アドレスおよび ポート番号 LKFL - リンク失敗 LMSG - ローカル・メッセージ ID LNOD - ローカル・ノード MSGI - メッセージ ID MSGT - メッセージ・テキスト MSST - ショート・テキスト OCCD - その他の完了コード OERR - その他のエラー状態のノード OMSG - その他のメッセージ ID PACC - PNODE アカウント PFIL - Process ファイル PNAM - Process 名 PNOD - PNODE PNUM - Process 番号 PPMN - PDS メンバー名 PRTY - 優先順位 QUEU - キュー RECC - レコード・カテゴリ RECI - レコード ID RETA - 保持 Process RMTP - リモート IP アドレスおよびポ ート番号 RSTR - 再始動された Process RUSZ - RU サイズ SACC - SNODE アカウント SBID - 実行依頼者ノード ID SBND - 実行依頼者ノード名 SBYR - 読み込まれたバイト SBYX - 送信されたバイト SCMP - 標準圧縮 SCOD - ソース完了コード SDDY - スケジュール日 SDS1 - ソースの処理 1 SDS2 - ソースの処理 1 SDS2 - ソースの処理 2 SDS3 - ソースの処理 3

パラメーター	説明	値
resp_buffer (続)		SELA - イベントの経過時間 SFIL - ソース・ファイル SMSG - ソース・メッセージ ID SNAM - ステップ名 SSTA - イベントの開始時刻 STAR - レコードのログ開始日/時刻 STAT - Process ステータス SNOD - SNODE SNVL - ボリュームのソース番号 SOPT - SYSOPTS レコード SRCR - 読み込まれたレコード SRUX - 送信された RU STIM - スケジュール時刻 STOP - イベントの終了時刻 SUBM - 実行依頼者 ID SUBN - 実行依頼者ノード SUMM - サマリー出力セクター SVOL - ソース・ボリューム配列 TIME - 実行依頼時刻 XLAT - 変換
resp_moreflag	詳細情報を取得するには、さらなる ndmapi_recvresp() または ndmapi_recvresp_c() 呼び出しの実行が必要であることを示します。このフラグは、select process および select statistics コマンドにおいてのみ使用されます。	なし

以下は、**ndmapi_recvresp()** または **ndmapi_recvresp_c()** の戻りコードを示しています。

戻りコード	説明
NDM_NO_ERROR	関数が正常に完了したことを示します。
NDM_ERROR	エラーが発生したことを示します。エラー・ステータスの詳細については、エラー構造体を確認してください。
TRUNCATED	受信バッファが小さすぎるため、データが切り捨てられたことを示します。

以下は、**ndmapi_recvresp()** 関数のサンプルです。

```
int32 rc, resp_length;
int32 resp_moreflag;
char resp_buffer[makbuf];

rc=ndmapi_recvresp (error,
                   &resp_length,
                   resp_buffer,
                   &resp_moreflag
                   );
```

ndmapi_sendcmd() または **ndmapi_sendcmd_c()**を使用した、Sterling Connect:Direct へのコマンドの送信)

Sterling Connect:Direct アプリケーションへのコマンドの送信を可能にするには、**ndmapi_sendcmd()** または **ndmapi_sendcmd_c()** を使用します。以下は、**ndmapi_sendcmd()** または **ndmapi_sendcmd_c()** のフォーマットを示しています。

```
int32 rc, resp_moreflag;
struct sendcmd_data ret_data;
rc=ndmapi_sendcmd (error,
                  "select process pnumber=2, ",
                  &resp_moreflag,
                  &ret_data
                  );
```

以下は、**ndmapi_sendcmd()** または **ndmapi_sendcmd_c()** のパラメーターです。

パラメーター	説明	値
error	エラー情報またはステータス情報を含む、Sterling Connect:Direct によって定義された構造体に対するポインター。	ポインター
cmd_text	Sterling Connect:Direct に対して送信するコマンドを指定する、ヌル終了テキスト・ストリングに対するポインター。コマンド・テキストの後に、セミコロンが続き、ヌルで終了する必要があります。 API から submit=filename コマンドを使用する場合、Process テキストに十分なストレージを割り当てるようにします。関数の完了時に、このパラメーターに関連付けられたテキスト・ストリング内に、実行依頼された Process のテキストが返されます。Process テキストに十分なストレージが割り当てられていない場合、コア・ダンプが発生します。	テキスト・ストリングに対するポインター
resp_moreflag	実行されたコマンドにおいて、さらに多くの応答が保留されていることを示すフラグに対するポインター。さらなる応答を取得するには、 ndmapi_recvresp() または ndmapi_recvresp_c() を呼び出します。	フラグに対するポインター

パラメーター	説明	値
ret_data	コマンドの内部応答情報を含む構造体に対するポインター。構造体は、以下のとおりです。 <pre>struct sendcmd_data { char * cmd_name; ulong cmd_id; long data1; long data2; long data3; };</pre>	構造体に対するポインター
sendcmd_data	ユーザー要求に関する一部の情報を呼び出し元に提供します。コマンド・テキストの解析は CMGR で行われるため、エンド・ユーザー・アプリケーション (EUA) は、テキストを生成しない限り、実行依頼されたコマンドを特定することはできません。	ユーザー要求に関する情報
cmd_name	実行依頼されたコマンドの名前を持つストリングに対するポインター。CLI は、このポインターを使用して、完了メッセージを表示します。このフィールドによって、EUA において固有コマンド知識を一切必要とすることなく、独自の完了メッセージを表示させることができます。	コマンド名に対するポインター
cmd_id	コマンド・テキスト内で検出されたコマンドの 4 バイトの識別子。以下は、4 バイトの識別子です。 <pre>/******CommandIDs******/ #define CHANGE_PROCESS 0x43484750 /* "CHGP" */ #define DELETE_PROCESS 0x44454c50 /* "DELP" */ #define FLUSH_PROCESS 0x464c5350 /* "FLSP" */ #define SELECT_PROCESS 0x53454c50 /* "SELP" */ #define SELECT_STATISTICS 0x53454c53 /* "SELS" */ #define SUBMIT 0x5355424d /* "SUBM" */ #define TRACE_API 0x41504920 /* "API" */ #define TRACE_CMGR 0x434d4752 /* "CMGR" */ #define TRACE_SMGR 0x534d4752 /* "SMGR" */ #define TRACE_PMGR 0x504d4752 /* "PMGR" */ #define TRACE_COM 0x434f4d4d /* "COMM" */ #define TRACE 0x54524143 /* "TRAC" */ #define STOPNDM 0x53544F50 /* "STOP" */</pre>	4 バイトの識別子
	CLI は、これらの識別子を使用して、ルールが準拠されるようにします。例えば、 <code>ndmapi_sendcmd</code> が <code>resp_moreflag</code> を指定して返され、 <code>cmd_id</code> が <code>SELECT_STATISTICS</code> または <code>SELECT_PROCESS</code> でない場合、CLI はエラーを生成します。	
data1, data2, and data3	今後の拡張用。data1 は、Process 番号を返すために、 <code>submit</code> コマンドで使用されます。data2 は、Process の結果 (0、4、8、または 16) を返すために、 <code>submit</code> コマンドで使用されます。	

以下は、`ndmapi_sendcmd()` または `ndmapi_sendcmd_c()` の戻りコードです。

戻りコード	説明
NDM_NO_ERROR or Process Number	関数が正常に完了したことを示します。
NDM_ERROR	エラーが発生したことを示します。エラー・ステータスの詳細については、エラー構造体を確認してください。

以下は、`ndmapi_sendcmd()` 関数のサンプルです。

```
int32 rc, resp_moreflag;
struct sendcmd_data ret_data;
rc=ndmapi_sendcmd(error,
                  "select process pnumber=2;",
                  &resp_moreflag,
                  &ret_data
                  );
```

カスタム C++ プログラムの作成

C++ API 呼び出しを使用したカスタム・プログラムを作成する場合、`ConnectDirectSession` と呼ばれるクラスを含める必要があります。呼び出しプログラムは、`ConnectDirectSession` をインスタンス化し、`send` および `receive` 関数を呼び出す必要があります。sdksample.C と呼ばれるサンプル・プログラムが提供されています。カスタム C++ プログラムを作成するには、`ConnectDirectSession` クラスを作成します。このクラスには、`ConnectDirectSession` インターフェース、およびこのクラスに関連付けられたストレージの割り当てや解放を行うためのコンストラクターおよびデストラクター呼び出しが含まれています。このクラスは、Sterling Connect:Direct メソッドに対するインターフェースであり、接続、コマンド、データ取得、およびエラーのサービスを提供します。各メソッドは、`CD_SUCCESS` または `CD_FAILURE` のいずれかを返します。

注: クライアント構成ファイルの `pathname` に `NDMAPICFG` 環境変数が設定される必要があります。環境変数の設定手順については、18 ページの『*CLI の起動*』を参照してください。

ConnectDirectSession クラスを使用するには、インストールで提供された cdunxsdk.h というヘッダー・ファイルがアプリケーションにインクルードされているとともに、ndmapi.a ファイルにリンクされている必要があります。以下は、ConnectDirectSession クラスのプログラム・サンプルです。

```

#include "cdunxsdk.h"
#include <iostream.h>
#include <string.h>
void getError(ConnectDirectSession& cdSess);

main()
{
    ConnectDirectSession cdSess;
    char processText[16384];

    if (cdSess.SessionINF->Connect() == CD_SUCCESS)
    {
        strcpy(processText, "submit maxdelay=unlimited sdksample process snode=SNODENAME");
        strcat(processText, "copy00 copy from (file=sample.txt pnode)");
        strcat(processText, "to (file=sample.000 snode disp=rpl)");

        if (cdSess.SessionINF->SendCommand(processText) == CD_SUCCESS)
        {
            printf("%s completed, pnumber = %ld.\n",
                cdSess.SessionINF->GetCommandName(),
                cdSess.SessionINF->GetProcessNumber());
            sprintf(processText, "SELECT STATISTICS PNUMBER=%ld DETAIL=YES;",
                cdSess.SessionINF->GetProcessNumber());
            if (cdSess.SessionINF->SendCommand(processText) == CD_SUCCESS)
            {
            }
            else
            {
                getError(cdSess);
            }
        }
        else
        {
            getError(cdSess);
        }
    }
    cdSess.SessionINF->Disconnect();
}
else
{
    getError(cdSess);
}

void getError(ConnectDirectSession& cdSess)
{
    if (cdSess.SessionINF->GetFirstError())
    {
        printf("\nError Message: %s", cdSess.SessionINF->GetMsgID());
        printf("\nError Feedback: %d", cdSess.SessionINF->GetFeedBackCode());
        printf("\nError RC: %d", cdSess.SessionINF->GetReturnCode());
        printf("\nError SUBST: %s\n", cdSess.SessionINF->GetSubstitute());
    }

    while (cdSess.SessionINF->GetNextError())
    {
        printf("\nError Message: %s", cdSess.SessionINF->GetMsgID());
        printf("\nError Feedback: %d", cdSess.SessionINF->GetFeedBackCode());
        printf("\nError RC: %d", cdSess.SessionINF->GetReturnCode());
        printf("\nError SUBST: %s\n", cdSess.SessionINF->GetSubstitute());
    }
}
}

```

以下の表は、ConnectDirectSession クラス・メソッドを説明しています。

メソッド	説明	パラメーター	戻り値
Connect	Sterling Connect:Direct サーバーへの接続を提供します。 void パラメーターを指定した Connect() は、クライアント構成ファイル内で指定されたホスト名およびポートに接続します。	void または IP アドレスおよびポートに対するポインター	CD_SUCCESS または CD_FAILURE
DisConnect	現行のセッションを切断します。	void	CD_SUCCESS または CD_FAILURE
SendCommand	処理のため、Sterling Connect:Direct コマンドをサーバーに送信します。	コマンド・テキスト・バッファに対するポインター	CD_SUCCESS または CD_FAILURE
ReceiveResponse	select statistics コマンドなど、前に実行されたコマンドからの応答を受信します。	void	CD_SUCCESS または CD_FAILURE

メソッド	説明	パラメーター	戻り値
GetResponse	ReceiveResponse 呼び出しからの応答を取得します。	void	応答バッファーに対するポインター
GetResponseLength	前に受信した応答バッファーの長さを返します。	void	前に実行された呼び出しからの応答バッファーの長さ
MoreData	前に実行した send コマンドの呼び出しからの未処理データが利用可能かどうかを示す値を返します。 戻り値が TRUE の場合、ReceiveResponse を再度呼び出して、データをさらに取得します。	void	TRUE - 未処理のデータがさらに存在する場合 FALSE - 未処理のデータが存在しない場合
GetCommandName	submit コマンドなど、前に実行された send コマンドのコマンド名を返します。	void	コマンド名バッファーに対するポインター
GetProcessNumber	前に実行された submit コマンドの Process 番号を返します。	void	submit コマンドの Process 番号 -1 - submit コマンドが見つからない場合
GetProcessCount	delete process、change process、または flush process を実行した、直前の send コマンドによる影響を受ける Process の番号を返します。	void	delete process、change process、または flush process を実行した、submit コマンドの Process 数。 -1 - submit コマンドが見つからない場合
GetCurrentError	エラー・データ・ポインターをリスト内の現在のエラーに移動します。	void	TRUE - 正常に実行された場合 FALSE - 現行エラーが存在しない場合
GetNextError	エラー・データ・ポインターをリスト内の次のエラーに移動します。	void	TRUE - 正常に実行された場合 FALSE - エラーがこれ以上見つからない場合
GetPreviousError	エラー・データ・ポインターをリスト内の直前のエラーに移動します。	void	TRUE - 正常に実行された場合 FALSE - 直前のエラーが存在しない場合
GetFirstError	エラー・データ・ポインターをリスト内の最初のエラーに移動します。	void	TRUE - 正常に実行された場合 FALSE - エラーが見つからない場合
GetLastError	エラー・データ・ポインターをリスト内の最後のエラーに移動します。	void	TRUE - 正常に実行された場合。それ以外は FALSE

メソッド	説明	パラメーター	戻り値
GetMsgID	現在のエラー・データ・ブロックのメッセージを取得します。 適切な結果を取得するためには、このメソッドを呼び出す前に、GetXXXXErrorメソッドのうちの1つを呼び出す必要があります。	void	戻り値: データ・ブロックに値が入っている場合のメッセージ ID に対するポインター
GetFeedBackCode	現在のエラー・データ・ブロックのフィードバック・コードを返します。	void	フィードバック・コード
GetReturnCode	Connect Direct の戻りコードを返します。	void	次の有効な Sterling Connect:Direct 戻りコードのうちの1つ: 1、4、8、16
GetStatus	ステータスを返します。	void	Sterling Connect:Direct ステータス・コード
GetSubstitute	エラーに関連づけられた現在の置換バッファを返します。	void	置換バッファに対するポインター。
DisplayError	出力ロケーションに対する現行のエラー・チェーンを表示します。	パラメーター: ファイル I/O 構造体に対するポインター	戻り値: エラー・チェーン内で見つかった最も上位のエラー、またはエラーが存在しない場合は -1 を返す。

以下は、ConnectDirectSession クラス・ヘッダーです。

```

#include <stdio.h>

// Error enumeration.
typedef enum CDErrorCode
{
    CD_SUCCESS = 0,
    CD_FAILURE = -1
} CDErrorCode;

// <<Interface>> class
CDSession
{
public:
    // Communication methods...
    virtual CDErrorCode Connect(void) = 0;
    virtual CDErrorCode Connect(char *IpAddress, char *IpPort) = 0;
    virtual CDErrorCode DisConnect(void) = 0;
    virtual CDErrorCode SendCommand(char *CmdText) = 0;
    virtual CDErrorCode ReceiveResponse(void) = 0;

    // Methods for retrieving ReceiveResponse data...
    virtual const char *GetResponse(void) = 0;
    virtual int          GetResponseLength(void) = 0;
    virtual bool         MoreData(void) = 0;

    // Methods for retrieving SendCommand return data...
    virtual const char *GetCommandName(void) = 0;
    virtual long        GetProcessNumber(void) = 0;
    virtual long        GetProcessCount(void) = 0;

    // Methods to iterate over error collection ...
    virtual bool        GetCurrentError(void) = 0;
    virtual bool        GetNextError(void) = 0;
    virtual bool        GetPreviousError(void) = 0;
    virtual bool        GetFirstError(void) = 0;
    virtual bool        GetLastError(void) = 0;

    // Methods to retrieve error data...
    virtual const char *GetMsgID(void) = 0;
    virtual int         GetFeedBackCode(void) = 0;
    virtual int         GetReturnCode(void) = 0;
    virtual int         GetStatus(void) = 0;
    virtual const char *GetSubstitute(void) = 0;

    // Method to display error collection... virtual int
    DisplayError(FILE *Output) = 0;
};

class ConnectDirectSession
{
public:
    // Interface classes CDSession
    *SessionINF;

    ConnectDirectSession();
    ~ConnectDirectSession();
};

```

ユーザー出口の作成

ユーザー出口 API 関数によって、Sterling Connect:Direct で使用するカスタム・プログラムを作成することができます。

ユーザー出口関数の理解

ユーザー出口プログラムは Sterling Connect:Direct 実行中の戦略上重要な処理時点で、ユーザー固有のロジックを呼び出すために、Sterling Connect:Direct によって使用されます。ユーザー出口プログラムは C 言語 または C++ 言語のプログラムでなければならず、シェル・スクリプトは使用できません。PMGR は、Sterling Connect:Direct 始動時に統計ユーザー出口プログラムを呼び出し、Sterling Connect:Direct の実行中、この出口の実行が継続されます。SMGRはセッション毎にファイル・オープン出口およびセキュリティー ユーザー出口を呼び出し、特定のセッションの終了時にこれらの出口を停止します。

注: exit_skeleton.c および exit_skeleton.C は 3 つの出口すべての作業例を含み、make_exit_c Make ファイルおよび make_exit_C Make ファイルを用いて作成することができます。

以下の表は、ユーザー出口プログラムを説明しています。

プログラム	説明
ファイル・オープン出口	<p>Sterling Connect:Direct は、copy ステートメントの処理時に、ソース・ファイルまたは宛先ファイルを開くようこのユーザー出口プログラムにメッセージを送信します。ファイル・オープン出口はソース・ファイルを開き、ファイル記述子を指定します。この出口では、ファイル名またはディレクトリー名に対するあらゆる種類の処理の実行が可能です。また、必要に応じて他のファイルへのオープン要求のリダイレクトも可能です。</p> <p>ファイル・オープン出口プログラム (この例では、「exit_skeleton」) は、root によって所有され、setuid ビットの設定されている必要があります。以下のコマンドを使用してください。</p> <pre>% chown root exit_skeleton % chmod u+s exit_skeleton</pre>
セキュリティー出口	<p>セキュリティー出口によって、独自のセキュリティー・システムの実装またはサード・パーティー・セキュリティー・システムへのアクセス提供が可能となります。</p>

プログラム	説明
統計出口	統計出口は、何らかのレコードが統計ファイルに書き込まれた後の、ユーザー出口プログラムへの通知です。統計レコードが統計ファイルに書き込まれる度に、正確なコピーがこの出口へ渡されます。

ユーザー出口関数

ユーザー出口プログラムが `exit_child_init()` 関数または `exit_child_init_c()` 関数を呼び出すと、ユーザー出口および Sterling Connect:Direct 間の接続が確立されます。接続は特別に指定された停止メッセージを通して終了されます。メッセージのタイプはインクルード・ファイル `user_exit.h` で定義されています。以下の関数は、ユーザー出口および Sterling Connect:Direct 間の通信に役立つものです。

C++ 関数	C 関数	説明
<code>exit_child_init()</code>	<code>exit_child_init_c()</code>	Sterling Connect:Direct およびユーザー出口プログラム間の通信を初期化するため、ユーザー出口プログラムの先頭行にこの関数を使用します。
<code>recv_exit_msg()</code>	<code>recv_exit_msg_c()</code>	Sterling Connect:Direct およびユーザー出口プログラムの両方で使用され、他の Process からのメッセージを受信します。受信出口メッセージは、他の Process からの応答を待ちます。
<code>send_exit_file()</code>	<code>send_exit_file_c()</code>	ユーザー出口プログラムは、Sterling Connect:Direct 用に既にファイルを開いている場合にこの関数を使用します。この関数は 1 つの Process から他の Process へオープン・ファイル記述子を渡すのに、基本的な UNIX のメソッドを使用します。
<code>send_exit_msg()</code>	<code>send_exit_msg_c()</code>	この関数は Sterling Connect:Direct およびユーザー出口プログラムの両方で使用され、他の Process へメッセージを送信します。送信メッセージの後には、他の Process からの応答を取得するための受信メッセージが続きます。

`exit_child_init()` または `exit_child_init_c()` を使用した通信の初期化

通信の初期化をするには、ユーザー出口プログラムのコードの先頭行に `exit_child_init()` 関数または `exit_child_init_c()` 関数を使用します。この関数は両サイドにおいて通信の準備が整っていることを確認するためのチェックを行います。`exit_child_init()` 関数のフォーマットは以下の通りです。

```
int exit_child_init( char * logfile )
```

exit_child_init() 関数または **exit_child_init_c()** 関数のパラメーターは以下の通りです。

パラメーター	説明	値
logfile	ユーザー出口プログラムによる使用のために開かれるログ・ファイルまたはトレース・ファイルの名前。ファイル・オープン出口およびセキュリティー出口は、root として実行される SMGR によって開始されるため、出口も root として実行されます。root としての出口の実行はログ・ファイルのファイル権限の問題を引き起こす可能性があるため、 logfile は、その所有者または権限の変更が容易に行えるようになっています。詳細については、 <code>d_dir/ndm/src/exit_skeleton.c</code> にある出口のサンプルを参照してください。	ログ・ファイルまたはトレース・ファイルの名前

exit_child_init() 関数または **exit_child_init_c()** 関数の戻りコードは以下の通りです。この関数の戻りコードは `ndmapi.h` で定義されています。

戻りコード	説明
GOOD_RC	Sterling Connect:Direct および ユーザー出口プログラム間の通信の初期化に成功。
ERROR_RC	Sterling Connect:Direct および ユーザー出口プログラム間の通信の初期化に失敗。

recv_exit_msg() または **recv_exit_msg_c()** を使用したメッセージの待機

recv_exit_msg() 関数または **recv_exit_msg_c()** 関数は、Sterling Connect:Direct からメッセージを受信するまで待機します。メッセージの受信もしくはエラーの発生まで、制御は保留されます。**recv_exit_msg()** のフォーマットは以下の通りです。

```
int recv_exit_msg( int exit_flag )
int * msg_type,
char * recv_buf, int *
recv_buf_len
```

recv_exit_msg() 関数または **recv_exit_msg_c()** 関数のパラメーターは以下の通りです。

パラメーター	説明	値
exit_flag	受信者 ID を指定するフラグ。ユーザー出口プログラムが使用できる有効な値は EXIT_PROGRAM のみです。	EXIT_PROGRAM
msg_type	受信メッセージ名へのポインター。メッセージは、Sterling Connect:Direct からの要求、およびそれに関連付けられるユーザー出口プログラムからの応答です。	メッセージへのポインター

パラメーター	説明	値
recv_buf	メッセージのメモリー・ロケーションへのポインター。	メッセージへのポインター
recv_buf_len	受信するメッセージの長さ (バイト)。	メッセージの長さ

recv_exit_msg() または **recv_exit_msg_c()** の戻りコードは以下の通りです。この関数の戻りコードは `ndmapi.h` で定義されています。

戻りコード	説明
GOOD_RC	メッセージの受信に成功。
ERROR_RC	エラーが発生し、メッセージの受信に失敗。原因の可能性には、Sterling Connect:Direct の終了、 exit_flag パラメーターでの無効値の使用、もしくは受信メッセージを一時保存するための受信バッファのサイズ不足が含まれます。

send_exit_file() または **send_exit_file_c()** を使用したファイル記述子の引き渡し

send_exit_file() 関数または **send_exit_file_c()** 関数を使用して、1 つの Process から他の Process へファイル記述子を渡します。**send_exit_file()** のフォーマットは以下の通りです。

```
int send_exit_file(int exit_flag
                 int fd)
```

send_exit_file() または **send_exit_file_c()** のパラメーターは以下の通りです。

パラメーター	説明	値
exit_flag	送信者 ID を指定するフラグ。ユーザー出口プログラムが使用できる有効な値は EXIT_PROGRAM のみです。	EXIT_PROGRAM
fd	Sterling Connect:Direct の代わりにユーザー出口プログラムが開いたファイルのファイル記述子。これは、 <code>open(2)</code> 関数で返されるものと同様です。	ファイル記述子

send_exit_file() または **send_exit_file_c()** の戻りコードは以下の通りです。関数の戻りコードは `ndmapi.h` で定義されています。

戻りコード	説明
GOOD_RC	ファイル記述子の受信に成功。

戻りコード	説明
ERROR_RC	エラーが発生し、ファイル記述子の送信に失敗。原因の可能性には、Sterling Connect:Direct の終了、 exit_flag パラメーターまたは fd パラメーターでの無効値の使用、もしくは最終送信メッセージが send_exit_msg 以外であることが含まれます。

send_exit_msg() または **send_exit_msg_c()** を使用した Sterling Connect:Direct へのメッセージ送信

send_exit_msg() 関数または **send_exit_msgc()** 関数によって、ユーザー出口プログラムによる Sterling Connect:Direct へのメッセージ送信が可能となります。この関数は、メッセージがキューに入れられた直後に、制御を呼び出し元に戻します。

send_exit_msg() 関数のフォーマットは以下の通りです。

```
int send_exit_msg(int exit_flag,
                 int msg_type,
                 char * send_buf, int
                 send_buf_len)
```

send_exit_msg() または **send_exit_msg_c()** のパラメーターは以下の通りです。

パラメーター	説明	値
exit_flag	送信者 ID を指定するフラグ。ユーザー出口プログラムが使用できる有効な値は EXIT_PROGRAM のみです。	EXIT_PROGRAM
msg_type	メッセージ名。メッセージは、Sterling Connect:Direct からの要求およびそれに関連付けられるユーザー出口プログラムからの応答です。	メッセージへのポインター
send_buf	送信されるメッセージのメモリー・ロケーションへのポインター。	メッセージへのポインター
send_buf_len	送信されるメッセージの長さ (バイト)。	メッセージの長さ

send_exit_msg() または **send_exit_msg_c()** の戻りコードは以下の通りです。関数の戻りコードは ndmapi.h で定義されています。

戻りコード	説明
GOOD_RC	メッセージの送信に成功。
ERROR_RC	エラーが発生し、メッセージの送信に失敗。原因の可能性には、Sterling Connect:Direct の終了、もしくは exit_flag パラメーターまたは msg_type パラメーターでの無効な値の使用が含まれます。

ユーザー出口メッセージの理解

Sterling Connect:Direct は、C++ プログラムでは `send_exit_msg()` 関数および `recv_exit_msg()` 関数、C プログラムでは `send_exit_msg_c()` 関数および `recv_exit_msg_c()` 関数をそれぞれ使用して、メッセージの送受信を行います。各メッセージで送信されるデータの完全な定義については、`d_dir/ndm/include/user_exit.h` および `d_dir/ndm/include/user_exit2.h` にあるインクルード・ファイルを参照してください。

注: copy 制御ブロックは `user_exit2.h` で定義されています。

統計出口メッセージ

統計出口には、1 つのメッセージ・タイプ `STATISTICS_LOG_MSG` のみが存在します。

Sterling Connect:Direct は、`STATISTICS_LOG_MSG` をユーザー出口プログラムに送信します。Sterling Connect:Direct が統計レコードを書き込む度に、このメッセージは、文字ストリングの正確なコピーを提供します。`STATISTICS_LOG_MSG` には Sterling Connect:Direct 統計レコードが含まれます。

ファイル・オープン出口メッセージ

ファイル・オープン出口には、以下の 4 つのメッセージ・タイプが存在します。

- ◆ `FILE_OPEN_OUTPUT_MSG`
- ◆ `FILE_OPEN_OUTPUT_REPLY_MSG`
- ◆ `FILE_OPEN_INPUT_MSG`
- ◆ `FILE_OPEN_INPUT_REPLY_MSG`

ファイル・オープン出口には、以下の制限が存在します。

- ◆ ユーザー出口に渡される `oflag` パラメーターは、`copy` ステートメントで明示的に指定されている処理に基づくか、あるいはデフォルト値を使用して、既に計算されています。ユーザー出口で `oflag` が `truncate` (切り捨て) に変更され、元々の処理指定が `MOD` (ファイルが既に存在する場合に、ファイルの終わりにコピーを付加するような指定) である場合、このユーザー出口によって、Process は、その Process における記述とは異なる動作をするようになります。
- ◆ Process によって指定されたファイル・タイプを変更しないでください。例えば、Process が通常のファイルを指定する場合には、ユーザー出口はパイプへのファイル記述子を開いたり返したりすることはできません。`copy` 制御ブロックのコンテンツを修正し、それを Sterling Connect:Direct へ返すような機能は提供されていません。
- ◆ `oflag` が書き込み権限のある状態でファイルを開くよう指定され、ユーザー出口がアクセス権限を読み取り専用に変更された場合、Sterling Connect:Direct は、読み取り専用ファイルへの書き込みの試行に失敗します。
- ◆ ディレクトリーへのアクセスを制限するアップロードおよびダウンロード・パラメーターは、このユーザー出口に対しては無効となります。

これらのパラメーターに関する詳細については、『*IBM Sterling Connect:Direct for UNIX 管理ガイド (IBM Sterling Connect:Direct for UNIX Administration Guide)*』の『*アクセス情報ファイルの保守 (Maintaining Access Information Files)*』を参照してください。

FILE_OPEN_OUTPUT_MSG

copy ステートメントの処理時に、Sterling Connect:Direct はユーザー出口プログラムに FILE_OPEN_OUTPUT_MSG を送信し、宛先 ファイルを開きます。FILE_OPEN_OUTPUT_MSG には、以下が含まれます。

- ◆ open 関数の oflag パラメーター (例: O_CREAT|O_RDWR|O_TRUNC)
- ◆ ファイル権限の制御を行う、open 関数の mode パラメーター
- ◆ ファイルを所有する UNIX ユーザー ID
- ◆ ファイルを所有する UNIX グループ ID
- ◆ UNIX ユーザー名
- ◆ Sterling Connect:Direct の copy 制御ブロックのコピー
- ◆ Sterling Connect:Direct の構文解析された sysopts 構造のコピー (copy 制御ブロックには、その処理からの実際の未加工バージョンが含まれます)

FILE_OPEN_OUTPUT_REPLY_MSG

ユーザー出口プログラムは、Sterling Connect:Direct FILE_OPEN_OUTPUT_MSG に応答メッセージを送信します。FILE_OPEN_OUTPUT_REPLY_MSG には、以下が含まれます。

- ◆ 成功の場合はゼロ (0)、失敗の場合はゼロ以外のステータス値
- ◆ ステータス・テキスト・メッセージ (ステータス値が失敗を示す場合には、ステータス・テキスト・メッセージはエラー・メッセージに含まれます)
- ◆ パイプ ID pid (パイプ I/O においては、パイプ I/O に対するシェル・コマンドを実行しているシェル・プロセスの UNIX プロセス ID)
- ◆ 開かれたファイルの実際のファイル名 (統計ログ・メッセージで使用されます)

ステータス値が成功を示すゼロの場合、ユーザー出口プログラムは直ちに **send_exit_file()** または **send_exit_file_c()** を呼び出し、開かれたファイルのファイル記述子を Sterling Connect:Direct に送信する必要があります。

FILE_OPEN_INPUT_MSG

copy ステートメントの処理時に、Sterling Connect:Direct は FILE_OPEN_INPUT_MSG をユーザー出口プログラムに送信し、ソース・ファイルを開きます。FILE_OPEN_INPUT_MSG には、以下が含まれます。

- ◆ open 関数の oflag パラメーター (例: O_RDONLY)
- ◆ ファイル権限の制御を行う、open 関数の mode パラメーター
- ◆ ファイルを所有する UNIX ユーザー ID
- ◆ ファイルを所有する UNIX グループ ID
- ◆ UNIX ユーザー名
- ◆ Sterling Connect:Direct の copy 制御ブロックのコピー
- ◆ Sterling Connect:Direct の構文解析された sysopts 構造のコピー (copy 制御ブロックには、その処理からの実際の未加工バージョンが含まれます)

FILE_OPEN_INPUT_REPLY_MSG

このメッセージ・タイプは、ユーザー出口プログラムが Sterling Connect:Direct FILE_OPEN_INPUT_MSG に応答メッセージを送信する際に使用されます。FILE_OPEN_INPUT_REPLY_MSG には、以下が含まれます。

- ◆ 成功の場合はゼロ (0)、失敗の場合はゼロ以外のステータス値
- ◆ ステータス・テキスト・メッセージ (ステータス値が失敗を示す場合には、ステータス・テキスト・メッセージはエラー・メッセージに含まれます)
- ◆ パイプ ID pid (パイプ I/O においては、パイプ I/O に対するシェル・コマンドを実行しているシェル・プロセスの UNIX プロセス ID)
- ◆ 開かれたファイルの実際のファイル名 (統計ログ・メッセージで使用されます)

セキュリティー出口メッセージ

セキュリティー出口には、以下の 4 つのメッセージ・タイプが存在します。

- ◆ GENERATE_MSG
- ◆ GENERATE_REPLY_MSG
- ◆ VALIDATE_MSG
- ◆ VALIDATE_REPLY_MSG

注意: セキュリティー出口が使用されている場合、Sterling Connect:Direct はユーザー ID 認証の際、この出口に依存します。セキュリティー出口が正しく実装されていないと、セキュリティーが侵害される可能性があります。

GENERATE_MSG

Sterling Connect:Direct は、セッション開始時にユーザー出口プログラムに対して生成メッセージを送信し、セキュリティー環境を確立します。PNODE はセキュリティー出口に GENERATE_MSG を送信し、SNODE での認証に使用されるユーザー ID およびセキュリティー・トークンを決定します。GENERATE_MSG には、以下が含まれます。

- ◆ 実行依頼者 ID
- ◆ PNODE ID
- ◆ PNODE ID パスワード (ユーザーが指定した場合)
- ◆ SNODE ID
- ◆ SNODE ID パスワード (ユーザーが指定した場合)
- ◆ PNODE 名
- ◆ SNODE 名

GENERATE_REPLY_MSG

ユーザー出口プログラムは、Sterling Connect:Direct に対して応答メッセージを送信します。GENERATE_REPLY_MSG には、以下が含まれます。

- ◆ 成功の場合はゼロ (0)、失敗の場合はゼロ以外のステータス値
- ◆ ステータス・テキスト・メッセージ (ステータス値が失敗を示す場合には、ステータス・テキスト・メッセージはエラー・メッセージに含まれます)
- ◆ SNODE 側でセキュリティー・コンテキストで使用される ID (生成メッセージ中の ID と同一の ID である場合もあります)
- ◆ SNODE 側のセキュリティー・コンテキストの ID と併用されるセキュリティー・トークン

VALIDATE_MSG

Sterling Connect:Direct は、ユーザー出口プログラムに対して検証メッセージを送信します。

SNODE は、セキュリティー出口に対して VALIDATE_MSG を送信し、PNODE から受信したユーザー ID およびセキュリティー・トークンを検証します。VALIDATE_MSG には、以下が含まれます。

- ◆ 実行依頼者 ID
- ◆ PNODE ID
- ◆ PNODE ID パスワード (ユーザーが指定した場合)
- ◆ SNODE ID
- ◆ SNODE ID パスワード (ユーザーが指定した場合)
- ◆ PNODE 名
- ◆ SNODE 名
- ◆ セキュリティー・トークンと併用される ID
- ◆ セキュリティー・トークン (パスワード、パスチケット、またはその他のセキュリティー・トークン)

VALIDATE_REPLY_MSG

ユーザー出口プログラムは、Sterling Connect:Direct VALIDATE_MSG に対して応答メッセージを送信します。VALIDATE_REPLY_MSG には、以下が含まれます。

- ◆ 成功の場合はゼロ (0)、失敗の場合はゼロ以外のステータス値
- ◆ ステータス・テキスト・メッセージ (ステータス値が失敗を示す場合には、ステータス・テキスト・メッセージはエラー・メッセージに含まれます)
- ◆ セキュリティー・コンテキストで使用される ID
- ◆ セキュリティー・コンテキストの IDと併用されるセキュリティー・トークン

ユーザー出口停止メッセージ

Sterling Connect:Direct は、ユーザー出口の実用的な作業がすべて完了した時点で、停止メッセージ STOP_MSG を送信し、ユーザー出口に終了するよう通知します。ユーザー出口 は、停止メッセージを受信した時、もしくは上記にリストされたユーザー出口関数がエラー・コードを返した時にのみ停止する必要があります。

copy 制御ブロック

copy 制御ブロック構造には、Sterling Connect:Direct における copy ステートメント Process ファイルの処理方法を制御するフィールドが含まれます。

出ログ・ファイル

ユーザー出口プログラムが initparm.cfg で指定された場合、Sterling Connect:Direct は出ログを作成します。出ログ・ファイルは、ユーザー出口プログラム専用に提供され、メッセージのデバッグおよびトレースの目的に使用されます。ユーザー出口プログラムは、STDOUT および STDERR で既に開いているログ・ファイルから開始されます。出ログ・ファイルは、以下の通りです。

- ◆ stat_exit.log
- ◆ file_exit.log

◆ security_exit.log

注: 通常の **printf()** 関数および **fprintf (stderr,...)** 関数でログ・ファイルにアクセス可能です。

ログ・ファイルはインストール作業ディレクトリー (d_dir) に置かれています。

```
.../d_dir/work/cd_node
```


A

アプリケーション・プログラミング・インターフェース (Application Program Interface: API)

エンド・ユーザー・アプリケーション(End User Application: EUA) が Sterling Connect:Direct ソフトウェアと通信できるようにするために呼び出される関数のライブラリー。

C

クライアント (Client)

Sterling Connect:Direct サーバーへの要求を作成し、サーバーによる応答を受け入れるプログラム。

コマンドライン・インターフェース (Command Line Interface: CLI)

コマンドライン環境から Sterling Connect:Direct の Process およびコマンドを実行依頼することが可能なプログラム。

コマンド・マネージャー (CMGR)

API から送信されるコマンドを実行し、結果を API に返すプログラム。CMGR は API と連動して、Sterling Connect:Direct の認証手順を実行します。これにより、システムへのアクセスが承認されているユーザー名およびパスワードかどうかを判断します。CMGR は、コマンドの実行の際に要求があった場合には PMGR と通信します。

D

デーモン (daemon)

クライアントにサービスを提供する長期実行プロセス。PMGR は、Sterling Connect:Direct for UNIX のデーモンです。

診断コマンド (Diagnostic Commands)

Sterling Connect:Direct ソフトウェアにおける問題の診断を支援する Sterling Connect:Direct コマンド。

E

エンド・ユーザー・アプリケーション (End User Application: EUA)

特定のタスクを遂行するために、エンド・ユーザーによって開発されるアプリケーション・プログラム。

実行キュー (Execution Queue)

TCQ (伝送制御キュー) 内の論理キュー。実行キュー内の Process は、Sterling Connect:Direct ノードから、あるいは Sterling Connect:Direct ノードへのデータ転送を行っているか、またはタスク実行が可能となる前に、リモート Sterling Connect:Direct ノードへの接続を待機しています。

H

保留キュー (Hold Queue)

TCQ (伝送制御キュー) 内の論理キュー。保留キュー内の Process は、オペレーターによる介入を待った後、スケジューリングのため待機キューに移動します。

I

IBM Sterling Connect:Direct

Sterling Connect:Direct を参照。

IBM Sterling Connect:Direct Browser User Interface

Sterling Connect:Direct Browser User Interface を参照。

IBM Sterling Connect:Direct for UNIX

Sterling Connect:Direct for UNIX を参照。

IBM Sterling Connect:Direct Node

Sterling Connect:Direct ノードを参照。

IBM Sterling Connect:Direct Process

Sterling Connect:Direct Process を参照。

IBM Sterling Control Center

Sterling Control Center を参照。

M

監視コマンド (Monitoring Commands)

統計ファイルおよび TCQ (伝送制御キュー) からの Sterling Connect:Direct Process 実行結果に関する情報の表示を可能とする Sterling Connect:Direct コマンド。

O

操作制御コマンド (Operational Control Commands)

Process の実行依頼、TCQ (伝送制御キュー) 内の特定の Process の特性の変更、TCQ からの実行中の Process や非実行中の Process の排除、および Sterling Connect:Direct の停止を可能とする、Sterling Connect:Direct コマンド。

P

Process マネージャー (Process Manager: PMGR)

Sterling Connect:Direct ソフトウェアの初期化、Sterling Connect:Direct API およびリモート Sterling Connect:Direct ノードからの接続要求の受け入れ、コマンド・マネージャーおよびセッション・マネージャーの作成、一元化された Sterling Connect:Direct 機能が必要な場合のコマンド・マネージャーおよびセッション・マネージャーからの要求の受け入れ、および Sterling Connection:Direct ソフトウェアの実行の停止を行う、長期実行型の Sterling Connect:Direct サーバー。

1 次ノード (Primary Node: PNODE)

Process が実行される Sterling Connect:Direct ノード。1 次ノードは、制御ノードまたはソース・ノードとも呼ばれますが、PNODE は受信側にもなり得るため、必ずしも送信ノードとはなりません。各 Process において、1 つの PNODE と 1 つの SNODE が存在します。Process の実行依頼者は、常に PNODE になります。PNODE 名は、1 から 16 文字の長さで指定できます。

S

セッション (Session)

2つのSterling Connect:Direct ノード間における接続。

セッション・マネージャー (Session Manager: SMGR)

リモート Sterling Connect:Direct ノードとの接続を作成および終了し、必要な Sterling Connect:Direct 作業を実行するサーバー・コンポーネント。

2次ノード (Secondary Node: SNODE)

Process 実行時に1次ノード (PNODE) と通信する Sterling Connect:Direct ノード。2次ノード (SNODE) は、参加ノード、ターゲット・ノード、宛先ノードと称される場合もあります。各 Process には、1つの PNODE と1つの SNODE が存在します。2次ノードは、別のノード (PNODE) によって開始された Process の実行に参加するノードです。SNODE 名は、1から16文字の長さで指定できます。

Sterling Connect:Direct

複数のデータ・センター全体にわたって情報の配信および実働アクティビティの管理を行う、データ転送ソフトウェア製品ファミリー。

Sterling Connect:Direct Browser User Interface

コマンドライン・インターフェースを通じた Sterling Connect:Direct コマンドの実行依頼に代わって、Sterling Connect:Direct Browser User Interface を使用することで、Microsoft Internet Explorer などのインターネット・ブラウザから Process を作成、実行依頼、監視することができます。適切な Sterling Connect:Direct 権限を持っている場合には、Sterling Connect:Direct Browser User Interface を使用して、ネットワーク・マップや初期化パラメーターの表示や変更といった、Sterling Connect:Direct システム管理タスクを実行することもできます。

Sterling Connect:Direct File Agent

Sterling Connect:Direct のアプリケーション・プログラムおよびコンポーネント。指定されたディレクトリーをスキャンして、ファイルの存在を検索します。監視対象ディレクトリーにファイルが見つかると、Sterling Connect:Direct は、Process を実行依頼するか、またはそのファイルに対するルールに指定されているアクションを実行します。

Sterling Connect:Direct for UNIX

Sterling Connect:Direct 製品の UNIX での実装。

Sterling Connect:Direct ノード (Sterling Connect:Direct Node)

Sterling Connect:Direct を稼働するすべてのコンピューター/ワークステーション。

Sterling Connect:Direct Process

事前定義してディレクトリー内に保存することができ、API を通じて実行依頼することで Sterling Connect:Direct for UNIX アクティビティを開始することが可能な一連のステートメント。Process 機能の例には、ファイルのコピ

ーやジョブの実行などがあります。

Sterling Control Center

Sterling Connect:Direct z/OS、UNIX、Microsoft Windows サーバーにおいて、オペレーション担当者に対して継続的かつ全社的なビジネス・アクティビティーの監視機能を提供する一元化された管理システム。複数の Sterling Connect:Direct サーバーを管理して Process の中断、解放、および削除を行い、Sterling Connect:Direct サーバーの停止や、実行中の Process または完了した Process に関する詳細な統計情報の表示を行います。

また、ネットワーク内の Sterling Connect:Direct for z/OS、UNIX、Microsoft Windows サーバー全体における Sterling Connect:Direct 処理を確認し、アクティブな Process および完了した Process に関する情報を検索する際のサービス・レベルを監視します。スケジュール通りに発生した、または発生しないデータ配信イベントの通知を受け取り、処理基準に基づいて、アラートの生成、Eメール通知の送信、エンタープライズ管理システム (EMS) に対する Simple Network Management Protocol (SNMP) トラップの生成、またはシステム・コマンドの実行を行うことができるルールを定義します。サーバー障害または Process が時間どおりに開始しないといった、アラートを監視します。

T

TCQ ステータス値 (TCQ Status Value)

Process が TCQ (伝送制御キュー) に置かれた際に、Sterling Connect:Direct によって Process に割り当てられる 2 文字のコード。Process のステータスは、**select process** コマンドを使用して調査することができます。

伝送制御キュー (Transmission Control Queue: TCQ)

Sterling Connect:Direct for UNIX に実行依頼されたすべての Process を保持するキュー。TCQ には、以下の 4 つの論理キューが含まれます

- ◆ 実行
- ◆ 待機
- ◆ タイマー
- ◆ 保留

タイマー・キュー (Timer Queue)

TCQ (伝送制御キュー) 内の論理キュー。タイマー・キュー上の Process は、開始時刻を待った後、スケジュールリングのため待機キューに移動します。

W

待機キュー (Wait Queue)

TCQ (伝送制御キュー) 内の論理キュー。待機キュー上の Process は、リモート Sterling Connect:Direct ノードから、またはリモート Sterling Connect:Direct ノードへの接続を待機します。

A

API 関数呼び出し

Sterling Connect:Direct サーバーへの接続 85
exit_child_init() 100
ndmapi_connect() 86
ndmapi_disconnect() 87
ndmapi_recvresp() 87
ndmapi_sendcmd() 91
概要 82
recv_exit_msg() 101
send_exit_file() 102
send_exit_msg() 103

C

C プログラム、コンパイル・コマンド 83

C++ プログラム、コンパイル・コマンド 84

cdcustrpt ユーティリティ 78

cdsacomp 70

cfgCheck ユーティリティ 77
引数 76

change process コマンド

class パラメーター 34
説明 22、32
フォーマット 33
hold パラメーター 34
newsnode パラメーター 34
概要 32
pname パラメーター 33、37、40、43、47、48
pnumber パラメーター 33、37、41、43、48
prty パラメーター 35
release パラメーター 35
snode パラメーター 33、38、41、44、51
submitter パラメーター 34、38、42、45、53

class パラメーター

change process コマンド 34
submit コマンド 25

cmd_id パラメーター 92

cmd_name パラメーター 92

cmd_text パラメーター 91

cmgr パラメーター 54

CMGR、説明 8

comm パラメーター 55

コマンド

TCQ 58
change process 22、32
規則 24
delete process 22、35
fg 21
flush process 22、37
フォアグラウンド 21
操作制御 22
select process 40、43
select statistics 47
stop 22、39
CLI の停止 22
submit 25
構文 22
trace 54

コマンドライン・インターフェース (CLI)

説明 18
起動 18
停止 22
使用 18

コマンド・マネージャー、概要 8

コマンド

ndmmsg 69
ndmxmlt 65

C++ プログラムのコンパイル・コマンド AIX

83、84
HP 83、84
Linux 83、84
LinuxS390 84
Sun 83、84

圧縮ユーティリティ 70
 構成検査ユーティリティ
 引数 76
 構成レポート・ユーティリティ、
 基本インストール 78
 cdcustrpt 78
 Sterling Connect:Direct Secure Plus 80
 SwiftNet 80
 サーバーへの接続、API 関数呼び出しを使用 85
 変換テーブルの作成 65

D

データ応答、API コマンド用 85
 delete process コマンド 22、35
 説明 35
 pname パラメーター 36
 pnumber パラメーター 37
 snode パラメーター 38
 submitter パラメーター 37
 説明
 CMGR 8
 PMGR 7
 SMGR 8
 ユーザー認証 9
 destfile パラメーター、select statistics コマンド 47
 direct コマンド
 パラメーター、-e nn 20
 パラメーター、-h 21
 パラメーター、-n name 20
 パラメーター、-P 19
 パラメーター、-p nnnnn 20
 パラメーター、-r 21
 パラメーター、-s 19
 パラメーター、-t nn 20
 パラメーター、-x 20
 パラメーター、-z 21

E

-e nn パラメーター、direct コマンド 20
 エラー番号、ndmxmlt によって生成 68
 error パラメーター
 ndmapi_connect() 関数 86
 ndmapi_recvresp() 関数 88
 ndmapi_sendcmd() 関数 91
 エラー応答 85
 ERROR_RC 戻りコード
 exit_child_init() 関数 101

recv_exit_msg() 関数 102、103
 サーバー接続の確立、API 関数を使用 85
 EX ステータス値 61
 実行キュー 60
 出口ログファイル 107
 file_exit.log 107
 stat_exit.log 107
 exit_child_init() 関数 100
 説明 100
 パラメーター、logfile 101
 戻りコード、ERROR_RC 101
 戻りコード、GOOD_RC 101
 exit_flag パラメーター
 recv_exit_msg() 関数 101
 exit_flag パラメーター、send_exit_file() 関数 102
 exit_msg パラメーター
 recv_exit_msg() 関数 103

F

-f パラメーター、ndmmsg コマンド用 70
 -ffiller パラメーター 66
 ファイル・オープン出口
 説明 99
 メッセージ・タイプ 104
 file_exit.log 107
 filename パラメーター 25
 ファイル、メッセージ 68
 flush process コマンド 22、37
 説明 37
 フォアグラウンド、CLI Process の切り替え 21
 関数呼び出し
 exit_child_init() 100
 ndmapi_connect() 86
 ndmapi_disconnect() 87
 ndmapi_recvresp() 87
 ndmapi_sendcmd() 91
 ndmapi-sendcmd() 91
 概要 82
 recv_exit_msg() 101
 send_exit_file() 102
 send_exit_msg() 103

G

汎用パラメーター 値 23

GOOD_RC 戻りコード

exit_child_init() 関数 101

recv_exit_msg() 関数 102、103

H

-h パラメーター、direct 用 21

HC ステータス値 63

HE ステータス値 63

HI ステータス値 63

HO ステータス値 63

hold パラメーター

change process コマンド 34

submit コマンド 25

保留キュー 62

HR ステータス値 62、63

HS ステータス値 63

I

immediate パラメーター 40

情報応答、API 85

L

-l パラメーター、ndmmsg コマンド用 70

リスト、パラメーター値 25

logfile パラメーター exit_child_init() 関数 101

M

-m パラメーター 66

Process の管理 58

TCQ 内での Process の操作 58

max.age、パラメーター 11

maxdelay パラメーター 26

メッセージ

ファイル、概要 68

メッセージ ID フォーマット 68

メッセージ・レコード・フォーマット 69

メッセージ・ユーティリティ

メッセージ ID フォーマット 68

メッセージ・レコード・フォーマット 69

概要 68

変換テーブルの修正 65

CLI Process の切り替え

フォアグラウンド 21

msg_type パラメーター、recv_exit_msg() 関数 101

msg_type パラメーター、send_exit_msg() 関数 103

N

-n name パラメーター、direct コマンド 20

ndm_hostname パラメーター 86

NDM_NO_ERROR 戻りコード

ndmapi_connect() 関数 87、90、92

ndm_portname パラメーター 87

ndmapi_connect() 関数

説明 86

error パラメーター 86、88、91

フォーマット 86

ndm_hostname パラメーター 86

NDM_NO_ERROR 87、90、92

ndm_portname パラメーター 87

ndmapi_disconnect() 関数、属性 87

ndmapi_recvresp() 関数

説明 87

例 91

resp_buffer パラメーター 88、89、90

resp_length パラメーター 88

resp_moreflag パラメーター 90

TRUNCATED 戻りコード 90

ndmapi_sendcmd() 関数 91

cmd_id パラメーター 92

cmd_name パラメーター 92

cmd_text パラメーター 91

説明 91

resp_moreflag パラメーター 91

ret_data パラメーター 92

sendcmd_data パラメーター 92

ndmmsg

説明 69

メッセージ ID フォーマット 68

メッセージ id フォーマット 68

メッセージ・レコード・フォーマット 69

概要 68

パラメーター、-l 70

パラメーター、-s 70
 パラメーター、-f 70
 メッセージ・テキストの表示 68
 ndmxlt
 変換テーブルの作成 65
 エラー生成 68
 変換テーブルの修正 65
 ndmxlt コマンド・パラメーター
 -ffiller 66
 -m 66
 -oooutputfile 65
 -rradix 66
 -ssourcefile 65
 ndmxlt ユーティリティ、copy ステートメント 67
 newname パラメーター、submit コマンド 26
 newsnode パラメーター、change process コマンド 34

O

-oooutputfile パラメーター、ndmxlt 65
 操作制御コマンド 22

P

-p nnnnn パラメーター、direct コマンド 20
 pacct パラメーター 26
 パラメーター、TCQ 用スケジューリング 59
 PE ステータス値 61
 pmgr パラメーター、trace コマンド 56
 PMGR、説明 7
 pname パラメーター
 change process コマンド 33、37、40、43、48
 delete process コマンド 36
 pnodeid パラメーター、submit コマンド 26
 pnumber パラメーター
 change process コマンド 33、37、41、43、48
 delete process コマンド 37
 Process の再始動、概要 9
 Process、サンプル 11
 prty パラメーター
 change process コマンド 35
 submit コマンド 26

S

-s パラメーター
 direct コマンド 19

Q

queue パラメーター 41、43
 キュー
 実行 60
 保留 62
 タイマー 62
 待機 61
 quiesce パラメーター 40

R

-r パラメーター、direct コマンド 21
 reccat パラメーター 48
 recids パラメーター 49
 recv_buf_len パラメーター 102
 recv_exit_msg() 関数
 説明 101
 ERROR_RC 戻りコード 102、103
 exit_flag パラメーター 101、102、103
 GOOD_RC 戻りコード 102、103
 msg_type パラメーター 101、103
 recv_buf_len パラメーター 102
 send_buf パラメーター 102
 release パラメーター 35
 リモート・ノード情報レコード
 作成 13
 resp_buffer パラメーター 88、89、90
 resp_length パラメーター 88
 resp_moreflag パラメーター
 ndmapi_recvresp() 関数 90
 ndmapi_sendcmd() 関数 91
 ret_data パラメーター 92
 retain パラメーター 27
 戻りコード
 ERROR_RC、exit_child_init() 関数 101
 ERROR_RC、recv_exit_msg() 関数 102、103
 GOOD_RC、exit_child_init() 関数 101
 GOOD_RC、recv_exit_msg() 関数 102、103
 NDM_NO_ERROR、ndmapi_connect() 関数 87、90、92
 TRUNCATED、ndmapi_recvresp() 関数 90
 -rradix パラメーター、ndmxlt ユーティリティ用 66
 ndmmsg コマンド 70
 sacct パラメーター 27
 サンプル

索引

- Process 11
 - シェル・スクリプト 12
- Scheduling Sterling Connect:Direct アクティビティ 59
 - retain パラメーター 59
 - startt パラメーター 59
- セキュリティー出口
 - 説明 99
 - メッセージ・タイプ 106
- select process コマンド 40、43
 - 説明 40、43
 - queue パラメーター 41、43
 - snode パラメーター 42、45
- select statistics コマンド 47
 - 説明 47
 - フォーマット 46
 - reccat パラメーター 48
 - recids パラメーター 49
 - required パラメーター 47
 - startt パラメーター 52
 - stopt パラメーター 52
- send_buf パラメーター
 - recv_exit_msg() 関数 102
 - send_exit_msg() 関数 103
- send_buf_len パラメーター 103
- send_exit_file() 関数
 - 説明 102
 - フォーマット 102
- send_exit_msg() 関数 103
 - 説明 103
 - フォーマット 103
 - send_buf パラメーター 103
 - send_buf_len パラメーター 103
- sendcmd_data パラメーター 92
- セッション・マネージャー、概要 8
- シェル・スクリプト、サンプル 12
- SMGR
 - trace コマンド 56
- SMGR、説明 8
- snode パラメーター
 - change process コマンド 33、38、41、44、51
 - delete process コマンド 37
 - select process コマンド 42、45
 - submit コマンド 28
- snodeid パラメーター 29
- srcfile パラメーター
 - select statistics コマンド 52
- sourcedfile パラメーター 65
- スタンドアロン型パッチ圧縮 70
- startt パラメーター
 - select statistics コマンド 52
 - submit コマンド 30
- stat_exit.log 107
- 統計出口
 - 説明 100
 - メッセージ・タイプ 104
- ステータス値
 - 保留キュー 63
 - 概要 59
 - 待機キュー 61
- step パラメーター 40
- Sterling Connect:Direct
 - API 関数呼び出し 82
 - コマンド 18
 - 関数 calls 82
- stop コマンド 22、39
 - 説明 39
 - immediate パラメーター 40
 - quiesce パラメーター 40
 - step パラメーター 40
- CLI の停止 22
 - stopt パラメーター 52
- submit コマンド
 - &symbolic name パラメーター 30
 - class 25
 - 説明 25
 - filename パラメーター 25
 - hold 25
 - newname パラメーター 26
 - パラメーター、maxdelay 26
 - パラメーター、prty 26
 - パラメーター、pacct 26
 - pnodeid パラメーター 26
 - retain パラメーター 27

sacct パラメーター 27
 snode パラメーター 28
 snodeid パラメーター 29
 startt パラメーター 30

submitter パラメーター
 change process コマンド 34、38、42、45、53
 delete process コマンド 37

Process の実行依頼、定義 22

T

-t nn パラメーター、direct コマンド 20

TCQ

コマンド 58
 hold パラメーター 59
 概要 58
 Process の進捗 60
 retain パラメーター 59
 startt パラメーター 59

TCQ ステータス値

保留キュー 62
 待機キュー 61

タイマー・キュー 62

trace コマンド 54

cmgr パラメーター 55
 comm パラメーター 55
 説明 54
 フォーマット 55
 pmgr パラメーター 56
 smgr パラメーター 56

変換テーブル・ユーティリティ

copy ステートメント 67
 変換テーブルの作成 65
 エラー番号 68
 変換テーブルの修正 65

伝送制御キューとそのコマンド 58

hold パラメーター 59
 概要 58
 保留キューのステータス値 63

伝送制御キュー、Process の進捗 60

TRUNCATED 戻りコード 90

U

ユーザー認証情報ファイル
 説明 9

ユーザー出口

ファイル・オープン出口 99
 ファイル・オープン出口メッセージ・タイプ 104
 セキュリティー出口 99
 セキュリティー出口タイプ 106
 統計出口 100
 統計出口メッセージ 104

ユーティリティ

ndmxt および copy ステートメント 67
 ndmxt、変換テーブルの作成 65
 ndmxt、変換テーブルの修正 65
 変換テーブル 65
 変換テーブル、copy ステートメント 67
 変換テーブル、変換テーブルの修正 65

V

構成ファイルの検証 77

view process、コマンド 22、40

W

WA ステータス値 61

待機キュー 61

WC ステータス値 61、62

ワイルドカード機能 23

WR ステータス値

タイマー・キュー 62
 待機キュー 61

WS ステータス値 61、62

X

-x パラメーター、direct コマンド 20

XXLT001I エラー番号、変換テーブル 68

XXLT002I エラー番号、変換テーブル 68

XXLT003I エラー番号、変換テーブル 68

XXLT004I エラー番号、変換テーブル 68

XXLT005I エラー番号、変換テーブル 68

索引

XXLT006I エラー番号、変換テーブル 68

XXLT007I エラー番号、変換テーブル 68

XXLT008I エラー番号、変換テーブル 68

Z

-z パラメーター、direct 21

特記事項

本書は米国 IBM が提供する製品およびサービスについて作成したものです。

本書に記載の製品、サービス、または機能が日本においては提供されていない場合があります。日本で利用可能な製品、サービス、および機能については、日本 IBM の営業担当員にお尋ねください。本書で IBM 製品、プログラム、またはサービスに言及していても、その IBM 製品、プログラム、またはサービスのみが使用可能であることを意味するものではありません。これらに代えて、IBM の知的所有権を侵害することのない、機能的に同等の製品、プログラム、またはサービスを使用することができます。ただし、IBM 以外の製品とプログラムの操作またはサービスの評価および検証は、お客様の責任で行っていただきます。

IBM は、本書に記載されている内容に関して特許権 (特許出願中のものを含む) を保有している場合があります。本書の提供は、お客様にこれらの特許権について実施権を許諾することを意味するものではありません。実施権についてのお問い合わせは、書面にて下記宛先にお送りください。

〒103-8510

東京都中央区日本橋箱崎町 19 番 21 号

日本アイ・ビー・エム株式会社、法務・知的財産、知的財産権ライセンス渉外

以下の保証は、国または地域の法律に沿わない場合は、適用されません。

IBM およびその直接または間接の子会社は、本書を特定物として現存するままの状態で提供し、商品性の保証、特定目的適合性の保証および法律上の瑕疵担保責任を含むすべての明示もしくは黙示の保証責任を負わないものとします。

国または地域によっては、法律の強行規定により、保証責任の制限が禁じられる場合、強行規定の制限を受けるものとします。

この情報には、技術的に不適切な記述や誤植を含む場合があります。本書は定期的に見直され、必要な変更は本書の次版に組み込まれます。IBM は予告なしに、随時、この文書に記載されている製品またはプログラムに対して、改良または変更を行うことがあります。

本書において IBM 以外の Web サイトに言及している場合がありますが、便宜のため記載しただけであり、決してそれらの Web サイトを推奨するものではありません。それらの Web サイトにある資料は、この IBM 製品の資料の一部ではありません。そ

これらの Web サイトは、お客様の責任でご使用ください。

IBM は、お客様が提供するいかなる情報も、お客様に対してなんら義務も負うことのない、自ら適切と信ずる方法で、使用もしくは配布することができるものとします。

本プログラムのライセンス保持者で、(i) 独自に作成したプログラムとその他のプログラム (本プログラムを含む) との間での情報交換、および (ii) 交換された情報の相互利用を可能にすることを目的として、本プログラムに関する情報を必要とする方は、下記に連絡してください。

IBM Corporation

Dept. 600A

Mail Drop 1329

Somers, NY 10589

U.S.A.

本プログラムに関する上記の情報は、適切な使用条件の下で使用することができますが、有償の場合もあります。

本書で説明されているライセンス・プログラムまたはその他のライセンス資料は、IBM 所定のプログラム契約の契約条項、IBM プログラムのご使用条件、またはそれと同等の条項に基づいて、IBM より提供されます。

この文書に含まれるいかなるパフォーマンス・データも、管理環境下で決定されたものです。そのため、他の操作環境で得られた結果は、異なる可能性があります。一部の測定が、開発レベルのシステムで行われた可能性があります。その測定値が、一般に利用可能なシステムのものと同じである保証はありません。さらに、一部の測定値が、推定値である可能性があります。実際の結果は、異なる可能性があります。お客様は、お客様の特定の環境に適したデータを確かめる必要があります。

IBM 以外の製品に関する情報は、その製品の供給者、出版物、もしくはその他の公に利用可能なソースから入手したものです。

IBM は、それらの製品のテストは行っておりません。したがって、他社製品に関する実行性、互換性、またはその他の要求については確認できません。

IBM 以外の製品の性能に関する質問は、それらの製品の供給者にお問い合わせください。

IBM の将来の方向または意向に関する記述については、予告なしに変更または撤回される場合があります。単に目標を示しているものです。

本書はプランニング目的としてのみ記述されています。記述内容は製品が使用可能になる前に変更になる場合があります。本書には、日常の業務処理で用いられるデータや報告書の例が含まれています。より具体性を与えるために、それらの例には、個人、企業、ブランド、あるいは製品などの名前が含まれている場合があります。これらの名称はすべて架空のものであり、名称や住所が類似する企業が実在しているとしても、それは偶然にすぎません。

著作権使用許諾:

本書には、様々なオペレーティング・プラットフォームでのプログラミング手法を例示するサンプル・アプリケーション・プログラムがソース言語で掲載されています。お客様は、サンプル・プログラムが書かれているオペレーティング・プラットフォームのアプリケーション・プログラミング・インターフェースに準拠したアプリケーション・プログラムの開発、使用、販売、配布を目的として、いかなる形式においても、IBM に対価を支払うことなくこれを複製し、改変し、配布することができます。このサンプル・プログラムは、あらゆる条件下における完全なテストを経ていません。従って IBM は、これらのサンプル・プログラムについて信頼性、利便性もしくは機能性があることをほのめかしたり、保証することはできません。これらのサンプル・プログラムは特定物として現存するままの状態を提供されるものであり、いかなる保証も提供されません。IBM は、お客様の当該サンプル・プログラムの使用から生ずるいかなる損害に対しても一切の責任を負いません。

それぞれの複製物、サンプル・プログラムのいかなる部分、またはすべての派生的創作物にも、次のように、著作権表示を入れていただく必要があります。

© IBM 2011. このコードの一部は、IBM Corp. のサンプル・プログラムから取られています。© Copyright IBM Corp. 2011.

この情報をソフトコピーでご覧になっている場合は、写真やカラーの図表は表示されない場合があります。

商標

IBM、IBM ロゴおよび ibm.com® は、世界の多くの国で登録された International Business Machines Corp. の商標です。他の製品名およびサービス名等は、それぞれ IBM または各社の商標である場合があります。現時点での IBM の商標リストについては、

<http://www.ibm.com/legal/copytrade.shtml> の『[Copyright and trademark information](#)』を参照してください。

Adobe、Adobe ロゴ、PostScript、PostScript ロゴは、Adobe Systems Incorporated 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。

IT Infrastructure Library は英国 Office of Government Commerce の一部である the Central Computer and Telecommunications Agency の登録商標です。

インテル、Intel、Intel ロゴ、Intel Inside、Intel Inside ロゴ、Centrino、Intel Centrino ロゴ、Celeron、Xeon、Intel SpeedStep、Itanium、および Pentium は、Intel Corporation または子会社の米国およびその他の国における商標または登録商標です。

Linux は、Linus Torvalds の米国およびその他の国における商標です。

Microsoft、Windows、Windows NT および Windows ロゴは、Microsoft Corporation の米国およびその他の国における商標です。

ITIL は英国 Office of Government Commerce の登録商標および共同体登録商標であって、米国特許商標庁にて登録されています。

UNIX は The Open Group の米国およびその他の国における登録商標です。

Java およびすべての Java 関連の商標およびロゴは Oracle やその関連会社の米国およびその他の国における商標または登録商標です。

Cell Broadband Engine, Cell/B.E は、米国およびその他の国における Sony Computer Entertainment, Inc. の商標であり、同社の許諾を受けて使用しています。

Linear Tape-Open、LTO、LTO ロゴ、Ultrium、および Ultrium ロゴは、米国およびその他の国における HP、IBM Corp.、および Quantum の商標です。

Connect Control Center[®]、Connect:Direct[®]、Connect:Enterprise[®]、Gentran[®]、Gentran[®]:Basic[®]、Gentran:Control[®]、Gentran:Director[®]、Gentran:Plus[®]、Gentran:Realtime[®]、Gentran:Server[®]、Gentran:Viewpoint[®]、Sterling Commerce[™]、Sterling Information Broker[®]、および Sterling Integrator[®] は、Sterling Commerce[™], Inc. および IBM Company の商標または登録商標です。

他の会社名、製品名およびサービス名等はそれぞれ各社の商標です。