

CA Client Automation

DTSCLI コマンド リファレンス ガイド

12.9



このドキュメント（組み込みヘルプシステムおよび電子的に配布される資料を含む、以下「本ドキュメント」）は、お客様への情報提供のみを目的としたもので、日本 CA 株式会社（以下「CA」）により随時、変更または撤回されることがあります。本ドキュメントは、CA が知的財産権を有する機密情報であり、CA の事前の書面による承諾を受けずに本書の全部または一部を複写、譲渡、変更、開示、修正、複製することはできません。

本ドキュメントで言及されている CA ソフトウェア製品のライセンスを受けたユーザは、社内でユーザおよび従業員が使用する場合に限り、当該ソフトウェアに関連する本ドキュメントのコピーを妥当な部数だけ作成できます。ただし、CA のすべての著作権表示およびその説明を当該複製に添付することを条件とします。

本ドキュメントを印刷するまたはコピーを作成する上記の権利は、当該ソフトウェアのライセンスが完全に有効となっている期間内に限定されます。いかなる理由であれ、上記のライセンスが終了した場合には、お客様は本ドキュメントの全部または一部と、それらを複製したコピーのすべてを破棄したことを、CA に文書で証明する責任を負います。

準拠法により認められる限り、CA は本ドキュメントを現状有姿のまま提供し、商品性、特定の使用目的に対する適合性、他者の権利に対して侵害のないことについて、黙示の保証も含めいかなる保証もしません。また、本ドキュメントの使用に起因して、逸失利益、投資損失、業務の中断、営業権の喪失、情報の喪失等、いかなる損害（直接損害か間接損害かを問いません）が発生しても、CA はお客様または第三者に対し責任を負いません。CA がかかる損害の発生の可能性について事前に明示に通告されていた場合も同様とします。

本ドキュメントで参照されているすべてのソフトウェア製品の使用には、該当するライセンス契約が適用され、当該ライセンス契約はこの通知の条件によっていかなる変更も行われません。

本書の制作者は CA および CA Inc. です。

「制限された権利」のもとでの提供：アメリカ合衆国政府が使用、複製、開示する場合は、FAR Sections 12.212、52.227-14 及び 52.227-19(c)(1)及び(2)、ならびに DFARS Section 252.227-7014(b)(3) または、これらの後継の条項に規定される該当する制限に従うものとします。

Copyright © 2014 CA. All rights reserved. 本書に記載されたすべての商標、商号、サービス・マークおよびロゴは、それぞれの各社に帰属します。

CA Technologies 製品リファレンス

このマニュアルセットで参照されている CA 製品は、以下のとおりです。

- CA Advantage® Data Transport® (CA Data Transport)
- CA Asset Intelligence
- CA APM (CA Asset Portfolio Management)
- CA Business Intelligence
- CA Common Services™
- CA DMM (CA Desktop Migration Manager)
- CA Embedded Entitlements Manager (CA EEM)
- CA MDM (CA Mobile Device Management)
- CA NSM (CA Network and Systems Management)
- CA Patch Manager
- CA Process Automation
- CA Service Desk Manager
- CA WorldView™

CA への連絡先

テクニカルサポートの詳細については、弊社テクニカルサポートの Web サイト (<http://www.ca.com/jp/support/>) をご覧ください。

目次

第 1 章 : Data Transport Service コマンドライン インターフェース 7

データ転送.....	8
管理対象転送.....	9
エージェント間転送.....	9
転送ステータス.....	10
ディスクリット モード.....	11
エージェントの DiscreetMode および転送優先順位の設定.....	12
転送優先順位およびエージェントの DiscreetMode 設定の連携.....	13

第 2 章 : dtscli コマンド - データ転送の管理 15

dtscli -tos パラメータ グループ - TOS の場所を指定.....	19
dtscli -sos パラメータ グループ - SOS の場所を指定.....	21
dtscli -transfer パラメータ グループ - 管理対象転送を実行.....	22
dtscli -job パラメータ グループ - 転送ジョブの管理.....	40
dtscli -schedule パラメータ グループ - スケジュールの管理.....	49
dtscli -log パラメータ グループ - ログ レベルおよびログ モードの指定.....	56
dtscli -agent パラメータ グループ - エージェント間転送の実行.....	58

第 3 章 : 管理対象転送 (TOS) 79

dtscli コマンド -- 転送の作成.....	80
dtscli コマンド - 複数の出力ファイルの送信 (管理対象).....	83
dtscli コマンド -- 転送ジョブの管理.....	84
dtscli コマンド - スキップ ロジックの使用.....	86
連続する転送ジョブでのスキップ ロジックの使用.....	87
dtscli コマンド - フィルタの適用.....	88
dtscli コマンド -- パーセル フィルタの追加.....	89
dtscli コマンド -- ファイル フィルタの追加.....	91
定義済みフィルタ.....	93
dtscli コマンド -- 転送エイリアスの使用.....	96
dtscli コマンド -- 転送ステータスの表示.....	97
HTTP プロトコルのパラメータ.....	97

第 4 章：エージェント間転送

99

dtsccli コマンド -- 簡単なポイントツーポイント転送の作成	100
dtsccli コマンド -- 複数のポイントツーポイント転送の作成	100
dtsccli コマンド - ホップ転送の設定	102
dtsccli コマンド - 複数の出力ファイルの送信（エージェント間）	104
dtsccli コマンド -- 方向の変更	105
TCP プロトコルのパラメータ	106
HTTP プロトコルのパラメータ	106
PPP（Point-to-Point Protocol）パラメータ	108
UDP プロトコルのパラメータ	110
CPI-C プロトコルのパラメータ	111
CPI-C を使用する複数の出力ファイルの送信	111
SPX プロトコルのパラメータ	112

第 1 章 : Data Transport Service コマンド ライン インターフェース

DTSCli を使用すると、データ転送のスケジュール設定、開始、監視、および制御を実行できます。 コマンドラインを使用して、以下のタスクを実行できます。

- 転送、転送ジョブ、およびスケジュールの作成
- 転送、転送ジョブ、およびスケジュールの削除
- 転送、転送ジョブ、およびスケジュールのステータスの取得
- 転送ジョブに対する転送の追加、挿入、および削除
- 転送ジョブの起動、中断、再開、中止、およびリセット
- 転送ジョブの優先度の調整
- スケジュールに対する転送ジョブの追加、挿入、および削除
- スケジュールの有効化および無効化
- エージェント間転送の実行
- 管理対象転送の実行

注 : Client Automation では、`dtaccli` の代わりに `dtsccli` が使用されます。`dtaccli` は現在も提供されていますが、`dtsccli` への移行をお勧めします。

このセクションには、以下のトピックが含まれています。

[データ転送](#) (P. 8)

[転送ステータス](#) (P. 10)

[ディスクリートモード](#) (P. 11)

データ転送

DTSLI コマンドを使用するデータ転送は、管理対象転送またはエージェント間転送のいずれかである必要があります。単一の転送の場合はいずれか一方にする必要があります、両方にすることはできません。

転送操作のルール

- 同じコマンドに複数の **-agent** オペランドを指定することはできません。
- 同じ **-transfer** オペランド内に複数の **rpath** を指定することはできません。
- 同じコマンドで **-agent** と **-transfer** オペランドを同時に指定することはできません。
- DTSLI コマンドのオペランドまたオペランドパラメータ式に埋め込みスペースが含まれている場合は、式全体を引用符で囲みます。入力パス オペランドと出力パス オペランド、フィルタ、プロトコルパラメータ、およびメソッドパラメータは、常に引用符で囲みます。また、この要件はほかのオペランドおよびパラメータにも適用されます。
- Data Transport Service r2 または r3 以上の場合にのみ、データ転送で *macros* をオプションで使用すると、*dtaccli* または *dtsccli* コマンドを使用するときに、入力ファイル（送信時）および出力ファイル（受信時）の名前を体系的に変更できます。Data Transport Service において、マクロとは、1つまたは複数の変数またはアクションを表す一連の文字列です。出力ファイル名には、送信コンピュータの名前、転送の日時、またはほかのカスタマイズされたテキストを反映させることができます。

重要： DTS は、IPv4 ブロードキャストおよび IPv4/IPv6 マルチキャストのアドレス指定をサポートします。BCAST 一対多プロトコルは、IPv4 アドレスでのみ使用できます。ブロードキャストタイプの転送を IPv6 ネットワーク経由で実行する場合、関連する IPv6 マルチキャストアドレスと共に MCAST プロトコルを使用します。

管理対象転送

管理対象転送では、転送の実行に転送オブジェクトサーバが使用され、任意でスケジュールオブジェクトサーバが使用されます。管理対象転送を実行するには、`dtsccli` コマンドの `-transfer (-t)` 転送パラメータオペランドを指定します。特定の条件では、転送オブジェクトサーバ (`-tos` オペランド) を指定します。転送に関わる DTS エージェントのセキュリティモードが「失敗」に設定されている場合は、`iuser` および `ruser` のパラメータが必要になります。また、転送対象のファイル/ディレクトリの入出力パスを、`ipath` および `rpath` パラメータを使用して指定します。管理対象転送のほかのすべてのオペランドは任意で指定します。

関連項目：

[管理対象転送 \(TOS\)](#) (P. 79)

[dtsccli コマンド - 複数の出力ファイルの送信 \(管理対象\)](#) (P. 83)

エージェント間転送

エージェント間転送では、あるエージェントからほかのエージェントへの転送が処理されます。転送オブジェクトサーバまたはスケジュールオブジェクトサーバは転送に関与しません。エージェント対エージェントの転送については、`dtsccli` コマンドの `-agent` パラメータオペランドを指定します。転送に関与する DTS エージェントのセキュリティモードが「失敗」に設定されている場合は、`iuser` および `ruser` のパラメータが必要になります。また、転送対象のファイル/ディレクトリの入出力パスを、`ipath` および `rpath` パラメータを使用して指定します。エージェント間転送のほかのすべてのオペランドは任意で指定します。

関連項目：

[エージェント間転送](#) (P. 99)

[dtsccli コマンド - 複数の出力ファイルの送信 \(エージェント間\)](#) (P. 104)

転送ステータス

`dtsccli` コマンドに対して **Data Transport Service** が返す値を示します。

Value	Description
INIT	転送ジョブは定義されましたが、起動されていません。
QUEUED	転送ジョブは起動されましたが、開始側によって現在はキューに格納されています。
READ_FILTERING	開始読み取りフィルタが転送ジョブに適用されています。
CONNECTING	転送ジョブは起動されましたが、データはまだ伝送されていません。
RETRYING	ネットワーク接続が失敗した場合、転送ジョブがセッション再試行を実行しています。
IN_PROGRESS	転送ジョブはアクティブであり、進行中です。
INTERRUPTED	転送ジョブは中断されています。
WRITE_FILTERING	応答書き込みフィルタが適用されています。
COMPLETE	転送ジョブは正常に完了しました。
ABORTED	転送ジョブは完了する前に終了されました。
ABORTING	転送ジョブが中止されています。
FAILED	転送ジョブ中にエラーが発生しました。チェックポイントの再起動が有効な場合、問題を修正した後で転送ジョブを再開することができます。
SUSPENDED	転送ジョブのデータ伝送は中断され、後で再開できます。
RESETTING	転送ジョブはリセットされています。
RESUMING	転送ジョブは再開されましたが、開始した DTS エージェントからステータスが受信されませんでした。
CALCULATING	転送オブジェクト サーバがネットワーク オブジェクト サーバからの開始側および応答側間の経路検索を要求し、応答を待機しています。
WAITING_TO_CONNECT	転送ジョブは、 DTS エージェントおよび転送ジョブ オブジェクト サーバ間の接続が確立されるのを待機しています。

関連項目：

[dtsccli -transfer パラメータ グループ - 管理対象転送を実行 \(P. 22\)](#)

[dtsccli -job パラメータ グループ - 転送ジョブの管理 \(P. 40\)](#)

ディスクリート モード

ディスクリート モードでの転送は、ディスクリート転送と呼ばれます。送受信するコンピュータの負荷が大きい場合は、ディスクリート転送はバックグラウンドで送受信されます。DTS エージェントでは、ディスクリート転送によるコンピュータのパフォーマンスとユーザの生産性への影響を最小限に抑えることができるよう、負荷に基づいて、転送を送信するタイミングを決定し、最適な転送速度を計算します。

ディスクリート モードでは、転送に関与するコンピュータの負荷が最小になり、パフォーマンスが最適化されます。そのため、ディスクリート転送は通常モードよりも遅いレートで実行される場合が多くなります。ディスクリート転送は、通常転送よりも優先度が低い転送に推奨されます。

ディスクリート転送の進捗の割合は、DTS エージェントによって自動的および動的に決定されます。DTS エージェントは、コンピュータの送受信の負荷（CPU 使用率およびマシンのネットワーク使用率の両方）を監視します。コンピュータの負荷が大きい場合、DTS エージェントは利用可能な予備のリソースを使用します。ただし、負荷が増加すると、エージェントによってリソースは解放され、コンピュータで実行されているほかのタスクが優先されます。このようにして、ディスクリート モードでは、転送を実行しながらコンピュータのパフォーマンスが最適化されます。

制限

ディスクリートモードは **Data Transport Service r1、r1.1**、およびサービスパック **1** ではサポートされません。すべての転送について、開始側または応答側の **DTS エージェント** がディスクリートモードをサポートしていない古いバージョンの **Data Transport Service** を実行している場合、ディスクリートモードの指定は無視されます。この場合の転送は正常ですが、通常モードで送受信されます。

ディスクリートモードは、ポイントツーポイント転送にのみ適用されます。ブロードキャスト、マルチキャスト、またはファンアウトには適用されません。

関連項目：

[dtscli -job パラメータグループ - 転送ジョブの管理](#) (P. 40)

エージェントの DiscreetMode および転送優先順位の設定

任意の **DTS エージェント** について、転送の開始、転送への応答、またはその両方にディスクリートモードを使用することを指定できます。また、任意の転送に関して、開始コンピュータ、応答コンピュータ、またはその両方にディスクリートモードを指定できます。エージェントの **DiscreetMode** 設定および転送優先順位設定の両方により、どのモード（通常またはディスクリート）が転送の送信に使用され、およびどのモードが転送の受信に使用されるかが決定します。

注： **DTS エージェントの DiscreetMode 設定 (Y または N)** を指定するには、*「Data エクスプローラ ヘルプ」* の「設定ポリシー」にある「データ転送エージェントプラグインポリシーグループ」のトピックを参照してください。

関連項目：

[dtscli -transfer パラメータグループ - 管理対象転送を実行](#) (P. 22)

[dtscli -agent パラメータグループ - エージェント間転送の実行](#) (P. 58)

転送優先順位およびエージェントの DiscreetMode 設定の連携

以下の表に、転送優先順位の設定およびエージェントのディスクリートモード設定がどのように連携して、転送の送信に使用されるモードが決定され、転送の受信に使用されるモードが決定されるのかを示します。

Initiator_Priority (ipriority), Responder_Priority (rpriority)	N から N へ	N から Y へ	Y から N へ	Y から Y へ
default、default	通常転送	ディスクリート 受信	ディスクリート 送信	ディスクリート 転送
default、discreet	ディスクリート 受信	ディスクリート 受信	ディスクリート 転送	ディスクリート 転送
discreet、default	ディスクリート 送信	ディスクリート 転送	ディスクリート 送信	ディスクリート 転送
discreet、discreet	ディスクリート 転送	ディスクリート 転送	ディスクリート 転送	ディスクリート 転送
default、urgent	通常転送	通常転送	通常転送	通常転送
discreet、urgent	通常転送	通常転送	通常転送	通常転送
urgent、default	通常転送	通常転送	通常転送	通常転送
urgent、discreet	通常転送	通常転送	通常転送	通常転送
urgent、urgent	通常転送	通常転送	通常転送	通常転送

この表は、デフォルトおよび緊急の転送の両方が同じ種類の転送、つまり通常転送を意味することを示しています。同様に、デフォルトまたは緊急の送信は通常送信を意味し、デフォルトまたは緊急の受信は通常受信を意味します。

任意の転送、任意の受信、または任意の送信に関して、緊急設定はディスクリート設定を常に上書きして通常にしますが、ディスクリート設定は通常設定を常に上書きしてディスクリートにします。

注：イニシエーターは送信コンピュータであり、応答側は受信コンピュータであると想定されています。ただし、開始側が受信コンピュータの場合、上記のテキスト中の「送信」などの用語は、「受信」などに置き換えてください。

第 2 章 : dtscli コマンド - データ転送の管理

コマンドライン引数は数が多いため、DTSCLI により、関連する引数がグループにまとめられます。引数はコマンドラインで左から右へ指定し、最近指定された引数のグループから順に解釈されます。

コマンドの形式

```
dtscli [-tos parameters]  
        [-sos parameters]  
        [-transfer parameters]  
        [-job parameters]  
        [-schedule parameters]  
        [-mode mode]  
        [-log parameters]  
        [-c comments-string]  
        [@filename]  
dtscli [-agent parameters]  
        [-log parameters]  
        [-c comments-string]  
        [@filename]  
dtscli -help [group [argument]]  
dtscli -version
```

注：このガイドでは、例の中の **-a|t** 規則は、管理対象転送とエージェント間転送のいずれかとして機能することを示しています。**-a|t** をそのまま使用せず、**-a**（または **-agent**）および **-t**（または **-transfer**）を適宜使用してください。

-mode

(オプション) 「オブジェクト存続期間」モードを指定します。これにより、DTSLI によって作成された転送が、作成後に起動されるかどうか、また起動される場合、転送完了後に転送が削除されるかどうかを制御します。

注：-mode オペランドは、-m と省略できます。

有効な値は以下のとおりです。

defer

転送ジョブを作成しますが、転送は実行しません。転送ジョブは遅延プロセスのスケジュールに追加されます。

例：以下の例では、オブジェクト存続期間の **defer** モードでエージェント間転送を作成しています。**defer** モードでは、転送および転送ジョブが作成されますが、起動はされません。転送ジョブは後で起動するためのスケジュールに追加されます。

```
dtscli -tos host=SomewhereElse user=Administrator::password
-t "ipath=branch2002::c:¥daily_receipts¥aug1000.xls"
"rpath=hq050::c:¥daily_receipts¥aug1000.xls"
-m defer
```

delete

転送ジョブを実行し、完了に成功または失敗した後、動的に作成された転送および転送ジョブを削除します。

deleteonsuccess

転送ジョブを実行し、正常に完了した後、動的に作成された転送および転送ジョブを削除します。

keep

転送ジョブを実行し、転送ジョブを再度起動することができるように、動的に作成された転送オブジェクトを保持します。動的に作成された転送ジョブを使用するか、またはコピーして新しい転送ジョブを作成する場合は、このオプションを使用します。

例：以下の例では、オブジェクト存続期間の **keep** モードで管理対象転送を作成しています。たとえば、毎日更新され、毎晩支社で処理するために送信されるファイルの転送ジョブを再使用する場合は、**keep** モードを指定します。

```
dtsccli -tos host=SomewhereElse user=Administrator::password
        -t "ipath=branch2002::c:¥daily_receipts¥aug1000.xls"
        rpath=hq050::c:¥daily_receipts¥aug1000.xls"
        -m keep
```

デフォルト： delete

-c

(オプション) コメント文字列を指定します。

デフォルト： なし

@filename

(オプション) 入力として読み取られ使用されるコマンドファイルのフルパス名を指定します。ファイル名と追加オペランド (-a、-mode など) の両方を同じ **dtsccli** コマンドで指定できます。

デフォルト： なし

例：以下の例では、エージェント間転送に対する入力として **dist_sales_rpt.txt** ファイルに指定したコマンドファイルを実行しています。

```
dtsccli -a @dist_sales_rpt.txt
```

-help

`dtsccli` コマンド パラメータおよびそのオプションごとに、階層構造のヘルプが表示されます。

注：-help オペランドは、-h と省略できます。

例：

- 上位レベルのヘルプの表示

```
dtsccli -h
```

- 転送パラメータのリストの表示

```
dtsccli -h transfer
```

- `delivery` という名前の転送パラメータのヘルプの表示

```
dtsccli -h transfer delivery
```

-version

`dtsccli` コマンド ライン インターフェースのバージョン情報を表示します。

注：-version オペランドは、-v と省略できます。

例：

```
dtsccli -v
```

dtsccli -tos パラメータ グループ - TOS の場所を指定

(オプション) **-tos** パラメータ グループによって、転送オブジェクト サーバ (TOS) の場所および必要なクレデンシャルが指定されます。

構文形式：

```
dtsccli -tos [host=hostname] [user=username::password] [timeout=timeout]  
[port=port] [trustuser=username::password] [trustdomain=domain]
```

host=hostname

(オプション) この転送に使用するホスト サーバの名前を指定します。サーバがローカル コンピュータに存在しない場合は、このパラメータを使用して明示的に指定する必要があります。

デフォルト： *localmachine*

user=username::password

(オプション) サーバに接続するためのユーザ名とパスワードを指定します。

デフォルト： なし

timeout=timeout

(オプション) 転送がサーバに対して接続を要求するとき、何秒でタイムアウトするか、値を秒単位で指定します。タイムアウト設定は、DTSCLI と TOS との間のすべての通信に使用されます。

port=port

(オプション) TOS への接続時に使用するレガシー通信ポートを指定します。

デフォルト： 8198

trustuser=username::password

(オプション) トラストドセキュリティ ドメインのユーザ名およびパスワードを指定します。

注： トラストドセキュリティ ドメインへのサインオンおよび信頼された転送の実行には、**trustuser=** および **trustdomain=** オペランドを同時に使用します。

trustdomain=domain

(オプション) トラストドセキュリティ ドメインの名前を指定します。トラストドセキュリティ ドメインに正常にサインオンすると、そのドメインで信頼された転送を実行できます。トラストドセキュリティ ドメインでは、**Data Transport Service** 管理者は、信頼されたユーザおよび信頼されたアクセス先の **DTS** を指定します。信頼されたアクセスとは、エージェントのセキュリティ モードが「fail」に設定された場合でも、転送時に **Data Transport** エージェントのユーザ名およびパスワードを指定せずに、これらのユーザが正常に転送を起動できることを意味します。

Data Transport Service の管理者がユーザ ID および **DTS** エージェントのコンピュータをトラストドセキュリティ ドメインに割り当て、ほかの要件を完了している場合は、信頼された転送を実行できます。信頼された転送の詳細について、および信頼された転送を実行できるかどうかを判断するには、**Data Transport Service** の管理者にお問い合わせください。

例 :

以下の例では、管理者が、信頼できるユーザ **redlead1** を信頼されたドメイン **redblue1** に追加しています。これで、転送にセキュリティ クレデンシャル (**iuser** および **ruser** パラメータ) を指定せずに、マシン **yellow** および **green** 間で転送が実行されます。

```
dtcli -tos host=syd500 user=sesede::silles timeout=30 port=8898
trustuser=redlead1::blue1 trustdomain=redblue1
-t "ipath=yellow::input_path" "rpath=green::output_path"
```

dtsccli -sos パラメータ グループ - SOS の場所を指定

(オプション) **-sos** パラメータ グループによって、スケジュール オブジェクト サーバ (SOS) の場所および必要なクレデンシャルが指定されます。

構文形式：

```
dtsccli -sos [host=hostname] [user=username::password] [timeout=timeout] [port=port]
```

host=hostname

(オプション) この転送に使用するホスト サーバの名前を指定します。サーバがローカル コンピュータに存在しない場合は、このパラメータを使用して明示的に指定する必要があります。

デフォルト： *localmachine*

user=username::password

(オプション) サーバに接続するためのユーザ名とパスワードを指定します。

注： **-default** 設定を指定すると、この設定でデフォルト設定が上書きされます。

timeout=timeout

(オプション) 転送がサーバに対して接続を要求するとき、何秒でタイムアウトするか、値を秒単位で指定します。タイムアウト設定は、DTSCli と SOS との間のすべての通信に使用されます。

port=port

(オプション) SOS への接続時に使用するレガシー通信ポートを指定します。

デフォルト： 8200

dtsccli -transfer パラメータ グループ - 管理対象転送を実行

-transfer パラメータ グループによって、管理対象転送時に転送オブジェクトのステータスが作成、削除、および取得されます。

構文形式：

```
dtsccli -transfer [checkpoint=y|n] [commit=y|n] [cskip=nn|prefixtransfer]
[fskip=nn|prefixtransfer] [data1=userdata1] [data2=userdata2]
[delimiter_conversion=y|n] [delivery=discreet_delivery_period]
[description=description] [direction=s|r] [expiry=discreet_expiry_period]
[id=object_id] "ipath=host::path" [ipriority=urgent|default|discreet]
[iuser=username::password] [iuuid=uuid] [ruuid=uuid] ["f_filters=filters"]
["p_filters=filters"] [label=label] [method=create|delete|status]
["method_parameters=method_parameters"] [output_mode=a|c|r|s|w]
[retry_interval=session_retry_interval] [retry_limit=session_retry_limit]
"rpath=host::path" [rpriority=urgent|default|discreet]
[ruser=username::password]
```

注：-transfer オペランドは、-t と省略できます。

checkpoint=y|n

(オプション) チェックポイントの再開機能 (Data Transport Service で使用されるエラー回復メカニズム) をオンにするかどうかを指定します。有効な値は、y (yes) および n (no) です。チェックポイントの再開が有効な場合は、転送中に障害が発生すると、障害が発生した時点から転送の再開が試みられます。チェックポイントの再開が無効な場合は、転送中に障害が発生すると、転送は最初から再開されます。

デフォルト：なし

commit=y|n

(オプション) ディスクへのコミット機能をオンにするかどうかを指定します。有効な値は、y (yes) および n (no) です。有効な場合、すべてのデータのパーセルを受信した後、DTS エージェントによってファイル出力バッファがフラッシュされハードディスクに書き込まれます。ディスクへのコミットを指定しない (n) 場合、DTS エージェントでは、オペレーティング システムのファイル I/O に依存して、必要時にバッファのフラッシュが実行されます。

デフォルト：なし

cskip=nn|prefixtransfer

fskip=nn|prefixtransfer

注：Data Transport Service r2、r3、および Client Automation のみに適用されます。

(オプション) スキップ ロジック設定を指定します。転送が完了 (cskip=) または失敗 (fskip=) した場合にスキップする転送数を指定します。スキップ ロジックは、順次転送ジョブの転送のみに適用されます。連続する転送ジョブでの各転送に対してスキップ ロジックを設定すると、次の *n* 回の転送をスキップしたり、この転送に失敗すると別の転送にスキップできます。

nn

転送をスキップする回数 (1、2、3、または負の数字など) を指定します。負の数字を指定すると、ジョブの中の転送がそれ以上起動されないように、残りのすべての転送がスキップされます。

prefixtransfer

以下のように、プレフィクス (I、L、または N) の後に、次に起動する転送の ID、ラベル、または名前を指定します。

■ **Itransfer-ID**

次に起動する転送のオブジェクト ID を指定します。たとえば、I657 は、オブジェクト ID が 657 の転送を指定します。

■ **Ltransfer-label**

次に起動する転送のラベルを指定します。たとえば、Lrecpt_mon では、ラベルが recpt_mon の転送が指定されます。

■ **Ntransfer-name**

次に起動する転送の名前を指定します。たとえば、Ntot251200 は、名前が tot251200 の転送を指定します。

デフォルト：なし (空)

注：ロジックのチェックは実行されません。cskip=nn 式および fskip=nn 式で nn に指定した値はすべてスキップ ロジック値に使用されます。

data1=user_data1

(オプション) ユーザ定義データを転送と関連付けることができます。

デフォルト：なし

data2=user_data2

(オプション) ユーザ定義データを転送と関連付けることができます。

デフォルト: なし

delimiter_conversion=y|n

(オプション) 区切り文字の変換をオンにするかどうかを指定します。有効な値は、**y** (yes) および **n** (no) です。受信コンピュータが送信コンピュータと異なるプラットフォームで実行されている場合でも、入力ファイル名と同じ形式で出力ファイル名を指定するときのインスタンスに対してのみ、区切り文字の変換 (**y**) を使用します。このようなプラットフォーム間転送では、区切り文字の変換によって、受信プラットフォームに転送されたデータの出力パス名が正しく区切られます。

関連するプラットフォームで同様の (完全に同じではない) 区切り文字およびファイル命名規則を使用している場合は、区切り文字の変換がプラットフォーム間転送で使用されます。最も一般的な用途には、**Windows** と **UNIX/Linux** コンピュータとの間の 1 対多の転送があります。これ以外のプラットフォーム間転送もサポートされています。

転送の作成または更新中に、転送の入力パス名および出力パス名を完全かつ正確に指定した場合は、区切り文字の変換を使用しないでください。プラットフォーム間転送の入力ファイル名として、同じ形式の出力ファイル名を指定する場合にのみ、区切り文字の変換を使用します。

注: 区切り文字の変換は注意して使用してください。受信プラットフォームのファイル命名規則または区切り文字規則に対するエラーチェックまたは検証メカニズムとしては機能しません。

デフォルト: なし

例: **Windows** および **UNIX** システムへのファンアウト転送の例を次に示します。

```
dtsccli -t "ipath=c:%input.txt"
      "rpath=unixserver1::c:%tmp%output.txt"
      iuser=administrator::adminpass
      ruser=root::rootpass
      delimiter_conversion=y
      -t "ipath=c:%input.txt"
      "rpath=winserver1::c:%tmp%output.txt"
      iuser=administrator::adminpass
      ruser=root::rootpass
      delimiter_conversion=y
```

delivery=discreet_delivery_period

(オプション) ディスクリット配信期間を分単位で指定します。転送の送信 (ipriority 使用) または転送の受信 (rpriority 使用) のどちらか、または両方に対して転送優先順位「discreet」を指定した場合にのみ、このオプションを適用できます。転送の送信および受信の両方にディスクリットモードを指定すると、同じディスクリット配信期間が両方に適用されます。

ディスクリット配信期間には、DTS エージェントの開始または応答によって、ディスクリットモードでの転送の送信または受信を試みる期間の合計が指定されます。ディスクリット送信またはディスクリット受信がこの期間内に完了しない場合、転送の優先順位は自動的に **discreet** から **urgent** に変更されます。(転送の優先順位の設定については、ipriority および rpriority オペランドの記述で説明されています)。

転送が起動されたとき (作成されたときではない) に時間の計測が開始されます。

ディスクリット配信期間 (delivery) は、通常、ディスクリット有効期限 (expiry) と共に使用されます。ディスクリット配信期間は、ディスクリット有効期限以下にする必要があります。これらのオプション (delivery または expiry) の片方のみを指定した場合、指定しないオプションにはデフォルト値が使用されます。

デフォルト: -1 (無制限)

例: ディスクリット配信期間が 360 分 (6 時間)、およびディスクリット有効期限 (expiry) が 720 分 (12 時間) でディスクリット転送を指定する場合は、以下を入力します。

```
dtscli -a|t "ipath=ukwin4q::c:¥winnt¥tngdts.ini"
            "rpath=philauk::c:¥config¥ukwinp4q¥tngdts.ini"
            ipriority=discreet rpriority=discreet
            delivery=360 expiry=720
```

"description=description"

(オプション) 転送の説明を指定します。説明に 1 つ以上のスペースが含まれている場合は、引用符で式全体を囲みます。

限度: 1 から 255 文字

direction=s|r

(オプション) 転送時に開始コンピュータがデータを受信または送信するかどうかを指定します。有効な値は以下のとおりです。

s

開始側から応答側にデータを送信します。

r

応答側から開始側にデータを送信します。

注： **direction=r** オプションを使用する場合、ソース ファイルは受信コンピュータ上にあり、宛先ファイルは開始コンピュータ上にあります (通常、その反対もあります)。

デフォルト： s

例： 以下の管理対象転送の例では、開始コンピュータ (NYC_HQ) は応答側 (STORE888) からデータを受信します。

```
dtsccli -a|t "ipath=NYC_HQ::c:¥receipts¥store888¥010398.rcp"  
"rpath=STORE888::PDEV.RRARI09.RECEIPTS" direction=r
```

expiry=discreet_expiry_period

(オプション) ディスクリット有効期限を分単位で指定します。転送の送信 (ipriority 使用) または転送の受信 (rpriority 使用) のどちらか、または両方に対して転送優先順位「discreet」を指定した場合にのみ、このオプションを適用できます。転送の送信および受信の両方にディスクリットモードを指定すると、同じディスクリット有効期限が両方に適用されます。

転送の期限が切れると、転送は削除され、送信の再試行は発生しません。有効期限 (expiry) は、ディスクリット配信期間 (delivery) 以上にする必要があります。これらのオプション (expiry または delivery) の片方のみを指定した場合、指定しないオプションにはデフォルト値が使用されます。

転送が起動されたとき (作成されたときではない) に時間の計測が開始されます。

デフォルト: -1 (無制限)

以下に例を示します。この例では、ディスクリット配信期間 (delivery) が 180 分 (3 時間)、およびディスクリット有効期限が 360 分 (6 時間) のディスクリット転送を指定しています。

```
dtscli -a|t "ipath=ukwin4q::c:%winnt%tngdts.ini"
          "rpath=philauk::c:%config%ukwinp4q%tngdts.ini"
          ipriority=discreet rpriority=discreet delivery=180 expiry=360
```

id=object_id

(オプション) 転送メソッド、転送ジョブメソッド、またはスケジュールメソッドを、1 つ、複数、またはすべての指定された転送、転送ジョブ、またはスケジュールに適切に適用します。

注: このオペランドは、既存の転送に対してのみ使用されます。転送を作成するときに ID を指定しないでください。

デフォルト: なし

例：

- メソッド（method=create|delete|status を参照）を単一の転送に適用するには、以下の形式を使用します。

```
dtsccli -t method=method id=id
```

- 追加、挿入、および削除メソッドでは、id= operand オペランドまたは the label=label オペランドのいずれかを使用して、メソッドを複数の転送に適用できます。id= operand では、メソッドを適用するときに複数の転送オブジェクト ID を指定するには、以下に示すように id= expression 全体を引用符で囲みます。

```
dtsccli -t method=method "id=120 130 140"
```

- メソッドをすべての転送オブジェクトに適用するには、以下に示すように id=all を指定します。

```
dtsccli -t method=method id=all
```

"ipath=host::path"

転送の入力ファイルを引用符で囲んで指定します。ホスト名の後ろにはコロンを 2 個と、転送のフルパス名を続けます。以下に例を示します。

- Windows

```
"ipath=mmmww185::c:¥receipts¥010398.rcp"
```

- UNIX/Linux

```
"ipath=machj17::/home/receipts/010398.rcp"
```

- HTTP 転送（インターネット ダウンロード）

```
"ipath=DTA_host_name::destination_path"
```

注： ipath= operand が必須です。

host

コンピュータ名、IP アドレス、または IPX アドレスを指定します。IP アドレスのセパレータにはピリオド (.) を使用します。たとえば、172.24.255.255 のようになります。IPX アドレスのセパレータにはピリオドまたはコロン (:) を使用します。たとえば、00000005.0000000000001 または 00000005:0000000000001 のようになります。

開始側として動作する DTS エージェントは、このコマンドラインユーティリティからリクエストを受信し、ターゲットの応答側または応答側グループへの接続を確立し、データを送信します。開始側は、転送を開始するコンピュータです。ホスト名を省略すると、ローカル ホストがデフォルトで使用されます。

path

入力データのフルパス名を指定します。

例：

- canoes という Windows コンピュータから tipper という別の Windows コンピュータに管理対象転送を送信する場合は、以下のコマンドを入力します。

```
dtscli -a|t "ipath=canoes::C:¥My Documents¥Monthly Reports¥apr2008.doc"
"rpath=tipper::C:¥Monthly Reports¥David¥apr2006.doc"
```

- CHICAGO12 と NEWYORK20 という 2 台の UNIX/Linux コンピュータ間で管理対象転送を送信するには、UNIX/Linux プロンプトで以下のコマンドを入力します。

```
dtscli -a|t "ipath=CHICAGO12::/home/ptfiles/ptcr1000.txt"
"rpath=NEWYORK20::/home/ptfiles/ptcr1000.txt"
```

- Miami という Windows コンピュータから NEWYORK20 という UNIX コンピュータに管理対象転送を送信する場合は、以下のコマンドを入力します。

```
dtscli -a|t "ipath=Miami::C:¥outgoing¥file.txt"
"rpath=NEWYORK20::/home/incoming/file.txt"
```

注：UNIX/Linux 上で dtscli コマンドを実行する場合、円記号は、「エスケープ」文字になることに注意してください。Windows パスを指定するには、円記号を 2 つ続ける必要があります。以下に例を示します。

```
dtscli -a|t "ipath=Miami::C:¥¥outgoing¥¥file.txt"
"rpath=NEWYORK20::/home/incoming/file.txt"
```

ipriority=urgent|default|discreet

(オプション) 転送の開始コンピュータに転送の優先順位を指定します。優先順位は、以下で説明されているように **urgent**、**default**、または **discreet** です。開始コンピュータに対してここで指定する転送の優先順位は、受信コンピュータに転送の優先順位 (**rpriority**) を設定し、転送の開始および受信コンピュータにエージェントの **DiscreetMode** (通常またはディスクリート) を設定した状態で機能します。(エージェントの **DiscreetMode** の設定は、**Data Transport** エージェントのパラメータの 1 つであり、**DSM** エクスプローラを使用して設定することもできます。)

注： 詳細については、「*DSM* エクスプローラ ヘルプ」の「設定ポリシー」にある「データ転送エージェント プラグイン ポリシー グループ」のトピックを参照してください。

以下の転送優先順位の設定およびエージェントの **DiscreetMode** 設定の両方により、どのモード (通常またはディスクリート) が転送の送信に使用され、およびどのモードが転送の受信に使用されるかが決定されます。

urgent

開始コンピュータのエージェントの **DiscreetMode** の設定が **Y** (はい) の場合でも、通常モードで転送が送信されます。

default

開始コンピュータのエージェントの **DiscreetMode** の設定が **N** (いいえ) の場合に通常モードで転送が送信されます。ただし、開始コンピュータのエージェントの **DiscreetMode** 設定が **Y** (はい) の場合、転送はディスクリート モードで送信されます。

discreet

開始コンピュータのエージェントの **DiscreetMode** の設定が **N** (いいえ) の場合でも、ディスクリート モードで転送が送信されます。

ディスクリート転送はバックグラウンドで送受信され、また送受信するコンピュータの負荷が大きくない場合は、最小限の帯域幅を使用します。**DTS** エージェントでは、ディスクリート転送によるコンピュータのパフォーマンスとユーザの生産性への影響を最小限に抑えることができるよう、負荷に基づいて、転送を送信するタイミングを決定し、最適な転送速度を計算します。

デフォルト： default

以下に例を示します。ディスクリット配信期間 (delivery) およびディスクリット有効期限 (expiry) にデフォルト値を使用して、ディスクリットモードで転送を送信および受信 (rpriority) するには、以下のコマンドを入力します。

```
dtscli -a|t "ipath=ukwin4q::c:¥winnt¥tngdts.ini"
            "rpath=philauk::c:¥config¥ukwinp4q¥tngdts.ini"
            ipriority=discreet rpriority=discreet
```

iuser=username::password

(オプション) 開始コンピュータのユーザ名およびパスワードを指定します。

注： 転送に関連する DTS エージェントのセキュリティ モードが「fail」に設定されている場合、このパラメータは必須です。

デフォルト： なし

以下に例を示します。この例では、開始コンピュータ (ipath iuser) と受信元コンピュータ (rpath ruser) の両方に対してユーザ名およびパスワードを指定しています。

```
dtscli -a|t "ipath=NYC_HQ::c:¥receipts¥store888¥010398.rcp"
iuser=Frank::furter
            "rpath=STORE888::PDEV.RRARI09.RECEIPTS" ruser=Molly::pitcher
```

iuuid=uuid

ruuid=uuid

(オプション) uuid= および ruuid= パラメータを指定します。これらのパラメータは、受信した転送リクエスト (転送開始のリクエスト) が実際に (ほかのコンピュータ宛てではなく) このコンピュータ宛てに送信されるように、転送の受信コンピュータの DTS エージェントで使用されます。Data Transport Service では、CA MDB の各 DTS エージェントに保存された情報に対して、これらのパラメータ値をチェックすることによって、Dynamic Host Configuration Protocol (DHCP) がより効果的に使用されています。

デフォルト： なし

重要： これらのパラメータは、管理データベース (MDB) の管理者または CA のテクニカル サポートの指示の下でのみ使用してください。

"f_filters=filters"

"p_filters=filters"

(オプション) 転送のフィルタのタイプを指定します。ファイルフィルタ (f_filters) では、一度に 1 ファイルずつ転送データが処理されます。パーセルフィルタ (p_filters) では、一度に 1 パーセルずつ転送データが処理されます (パーセルは、ファイル内のより小さいデータ単位です)。

通常、フィルタ パラメータを指定するのはエージェント間転送のみです。管理対象転送では指定しません。ただし、暗号化フィルタだけは例外です。パラメータが必要なほかのすべてのフィルタの場合は、管理対象転送によって転送オブジェクト サーバで指定された値が自動的に取得されます。

ファイルを指定するには、以下の形式を使用します。

```
dtsccli -a|t "ipath=ipath" "rpath=rpath" ...  
"f_filters=file_filters"
```

パーセルを指定するには、以下の形式を使用します。

```
dtsccli -a|t "ipath=ipath" "rpath=rpath" ...  
"p_filters=parcel_filters"
```

単一の読み取りフィルタや書き込みフィルタ、または読み取りフィルタと書き込みフィルタが一致するペアを指定できます。

"f_filters=file_filters" または "p_filters=parcel_filters" 式全体を引用符で囲みます。複数のフィルタをスペースで区切ります。

フィルタ指定の各タイプには、以下で示す位置にコロン (:) が必要です。

- 読み取りフィルタおよび一致する書き込みフィルタを指定するには、以下を入力します。

```
"f|p_filters=read_filter_name[(read_filter_params)]  
:write_filter_name[(write_filter_params)]"
```

- 一致する書き込みフィルタがない読み取りフィルタを指定するには、以下を入力します。

```
"f|p_filters=read_filter_name[(read_filter_params)]:"
```

- 一致する読み取りフィルタがない書き込みフィルタを指定するには、以下を入力します。

```
"f|p_filters=:write_filter_name[(write_filter_params)]"
```

フィルタは以下の順序で適用されます。

1. ファイル読み取りフィルタ
2. パーセル読み取りフィルタ
3. パーセル書き込みフィルタ
4. ファイル書き込みフィルタ

同じタイプの読み取りフィルタ（ファイルまたはパーセル）は、コマンドラインで指定する場合と同じ順序で適用されます。同じタイプの書き込みフィルタは、コマンドラインで指定する場合と反対の順序で適用されます。

デフォルト： バイナリ読み取りおよび書き込みフィルタ

以下に例を示します。次のコマンドでは、以下の順序で指定されたフィルタが適用されます。

- バイナリ読み取りパーセル フィルタ
- Huffman 圧縮パーセル フィルタ
- Huffman 解凍パーセル フィルタ
- バイナリ書き込みパーセル フィルタ

```
dtscli -a "ipath=ukwip3n::c:%temp%getstart.pdf"
        "rpath=ukwip3n::c:%temp%getstart2.pdf"
        "p_filters=BINARY_READ:BINARY_WRITE
        HUFFMAN_COMPRESS:HUFFMAN_UNCOMPRESS"
```

この例では、どのフィルタにもパラメータがありません。

注： 各転送には、テキスト フィルタとバイナリ フィルタのいずれかを追加する必要があります。フィルタを指定する場合、バイナリ フィルタはデフォルトで適用されません。したがって、同じコマンドでテキスト フィルタとバイナリ フィルタのいずれかを明示的に指定する必要があります。

label=label

(オプション) 作成する転送にラベルを指定します。ラベルを割り当てるには、**label=label** オペランドを使用して、数値以外の値を使用し、できるだけ論理的で簡単な名前を指定します。オブジェクト作成時にのみ、オブジェクトを作成する **dtsccli** コマンドの入力時にラベルを指定できます。

ラベルを使用すると、エイリアス (1 つ以上のオブジェクトを識別するためのオブジェクト ID 以外の代替手段) が提供されます。必要に応じて、複数のオブジェクト ID 値の代わりに単一のラベル値を指定することによって、複数のオブジェクトに同じラベルを付与でき、単一コマンドでこれらのオブジェクトすべてに同じメソッドを適用することができます。

この方法でラベルを使用するには、2 ステップのプロセスが必要です。

- 各オブジェクトに同じラベルを指定するには、**label=label** オペランドを使用して、各オブジェクトを作成します (その結果、各オブジェクトには一意のオブジェクト ID の同じラベルが付与されます)。
- 必要なメソッドをすべてのオブジェクトに適用させるには、**"id=id1 id2 id3..."** 式の代わりに **id=label** を指定します。

デフォルト: なし

例:

- 以下の例では、同じラベル (ステータス) を持つ 3 つの転送が作成されます。

```
dtsccli -t "ipath=nnwipg4h::c:%pics%large1.bmp"
"rpath=sswipg55::c:%pics%large1.bmp" label=stats
-t "ipath=nnwipg4h::c:%pics%large2.bmp"
"rpath=sswipg55::c:%pics%large2.bmp" label=stats
-t "ipath=nnwipg4h::c:%pics%large3.bmp"
"rpath=sswipg55::c:%pics%large3.bmp"
label=stats -mode defer
```

- 以下の例では、3 つの転送がすべて削除されます。

```
dtsccli -t id=stats method=delete
```

注: オブジェクトのグループに同一のラベルを指定し、また同一のメソッドを適用するこの手法は、転送、転送ジョブ、およびスケジュールにも適用できます。

method=create|delete|status

(オプション) 転送に適用するメソッドを指定します。有効な値は次のとおりです。

create

転送を作成します。

delete

1 つ以上の既存の転送を削除します。引用符で囲んだスペース区切りのリストで、**id=object_id** オペランドを使用して削除する転送を指定します。以下のフォーマットを使用します。

```
method=delete "id=id1 id2 id3 ... "
```

status

1 つ以上の転送のステータスが表示されます。引用符で囲んだスペース区切りのリストで、**id=object_id** オペランドを使用してチェックする転送を指定します。以下のフォーマットを使用します。

```
method=status "id=id1 id2 id3 ... "
```

指定した各転送のステータスまたは転送ステータスがコマンドの実行後に返されます。

デフォルト: create

例:

- そのオブジェクト ID が 442、445 および 554 である 3 つの転送を削除します。

```
dtscli -t method=delete "id=442 445 554"
```

- 同じ 3 つの転送のステータスを表示します。

```
dtscli -t method=status "id=442 445 554"
```

"method_parameters=method_parameters"

(オプション) 現在は使用されていません。この転送メソッドパラメータは、将来の使用のために予約されています。

デフォルト: なし

output_mode=a|c|r|s|w

(オプション) 出力モードを指定します。有効な値は以下のとおりです。

a

既存ファイルにデータを追加します。ファイルがない場合は、転送に失敗します。

c

出力ファイルを新規作成します。すでにファイルがある場合は、転送に失敗します。

r

既存ファイルを置き換えます。すでにファイルがない場合は、転送に失敗します。

s

出力ファイルを新規作成します。すでにファイルがある場合は、既存ファイルにデータが追加されます。

w

出力ファイルを新規作成します。すでにファイルがある場合は、置き換えられます。

デフォルト : w

retry_interval=session_retry_interval

(オプション) セッション再試行の間隔 (コンピュータへの接続中に転送の失敗が発生した場合の再試行の分数) を指定します。

デフォルト : 60

retry_limit=session_retry_limit

(オプション) セッション再試行の制限 (コンピュータへの接続中に転送の失敗が発生した場合に実行する再試行の回数) を指定します。

デフォルト : 20

"rpath=host::path"

引用符で囲まれた転送の出力ファイルを指定します。ホスト名の後ろにはコロンを 2 個と、転送のフルパス名を続けます。以下に例を示します。

■ Windows

```
"rpath=mmmww188::c:¥receipts¥store888¥010398.rcp"
```

■ UNIX/Linux

```
"rpath=uxmnuu18::/home/receipts/store888/010398.rcp"
```

■ HTTP 転送（インターネット ダウンロード）

```
"rpath=http_host_name::http_source_file_path_name"
```

注： rpath= operand が必須です。

host

コンピュータ名、IP アドレス、または IPX アドレスを指定します。IP アドレスのセパレータにはピリオド (.) を使用します。たとえば、172.24.255.255 のようになります。IPX アドレスのセパレータにはピリオドまたはコロン (:) を使用します。たとえば、00000005.0000000000001 または 00000005:0000000000001 のようになります。

応答側として動作する DTS エージェントでは、受信接続をリスニングおよび応答します。これらの受信接続（受信データ転送）は、DTS エージェントによって開始されます。

path

受信 DTS エージェントで使用する出力データの完全パス名を指定します。

注： UNIX/Linux から Windows へのデータ転送に dtscli コマンドを使用するときに、出力パスを引用符で囲まない場合は、各ディレクトリおよびパスを区切るために円記号を 2 つ入力する必要があります。

例：

- この UNIX/Linux から Windows の例では、`rpath` が引用符で囲まれているため、必要な円記号の区切り文字は 1 つのみです。

```
dtsccli -a|t "ipath=uxmnjj17::/home/receipts/010398.rcp"  
"rpath=mmmw188::c:¥receipts¥store888¥010398.rcp"
```

- この例では、引用符を使用して各入力および出力パスの式を囲っていないため、円記号が 2 つ必要です。

```
dtsccli -a|t ipath=iuxmnjj17::/home/receipts/010398.rcp  
rpath=mmmw188::c:¥¥receipts¥¥store888¥¥010398.rcp
```

注：各ディレクトリやパスを区切るための 2 つの円記号は、UNIX/Linux から Windows への転送の出力パスでのみ必要です。

`rpriority=urgent|default|discreet`

（オプション）転送の受信コンピュータに転送の優先順位を指定します。優先順位は、以下で説明されているように `urgent`、`default`、または `discreet` です。ここで指定する転送優先順位は、開始コンピュータの転送優先順位の設定（`ipriority`）および転送の開始コンピュータおよび応答コンピュータのエージェントの `DiscreetMode` 設定（通常またはディスクリート）と連携します。（エージェントの `DiscreetMode` の設定は、`Data Transport` エージェントのパラメータの 1 つであり、DSM エクスプローラを使用して設定することもできます。）

注：詳細については、「*DSM エクスプローラ ヘルプ*」の「設定ポリシー」にある「データ転送エージェントプラグインポリシー グループ」のトピックを参照してください。

以下の転送優先順位の設定およびエージェントの `DiscreetMode` 設定の両方により、どのモード（通常またはディスクリート）が転送の送信に使用され、およびどのモードが転送の受信に使用されるかが決定されます。

urgent

応答コンピュータのエージェントの `DiscreetMode` の設定が Y（はい）の場合でも、通常モードで転送が受信されます。

default

応答コンピュータのエージェントの **DiscreetMode** の設定が **N**（いいえ）の場合に通常モードで転送が受信されます。ただし、応答コンピュータのエージェントの **DiscreetMode** 設定が **Y** の場合、転送はディスクリート モードで受信されます。

discreet

応答コンピュータのエージェントの **DiscreetMode** の設定が **N** の場合でも、ディスクリート モードで転送が受信されます。

ディスクリート転送はバックグラウンドで送受信され、また送受信するコンピュータの負荷が大きい場合は、最小限の帯域幅を使用します。DTS エージェントでは、ディスクリート転送によるコンピュータのパフォーマンスとユーザの生産性への影響を最小限に抑えることができるよう、負荷に基づいて、転送を送信するタイミングを決定し、最適な転送速度を計算します。

デフォルト : default

以下に例を示します。ディスクリート配信期間 (**delivery**) およびディスクリート有効期限 (**expiry**) にデフォルト値を使用して、ディスクリート モードで転送を送信 (**ipriority**) および受信するには、以下のコマンドを入力します。

```
dtscli -a|t "ipath=ukwin4q::c:¥winnt¥tngdts.ini"
            "rpath=philauk::c:¥config¥ukwinp4q¥tngdts.ini"
            ipriority=discreet rpriority=discreet
```

ruser=username::password

（オプション）応答コンピュータのユーザ名およびパスワードを指定します。

注：転送に関連する DTS エージェントのセキュリティ モードが「失敗」に設定されている場合、このパラメータは必須です。

デフォルト : なし

例：以下の例では、開始コンピュータ (**ipath**) と受信側コンピュータ (**rpath**) の両方に対してユーザ名およびパスワードが指定されています。

```
dtaccli -a|t "ipath=NYC_HQ::c:¥receipts¥store888¥010398.rcp"
            iuser=polly::nomial ¥
            "rpath=STORE888::PDEV.RRARI09.RECEIPTS" ruser=redbird::bdfeder ... ¥
```

関連項目：

[dtsccli コマンド - フィルタの適用](#) (P. 88)

[dtsccli コマンド -- 転送の作成](#) (P. 80)

[dtsccli コマンド - スキップ ロジックの使用](#) (P. 86)

[転送ステータス](#) (P. 10)

[エージェントの DiscreetMode および転送優先順位の設定](#) (P. 12)

dtsccli -job パラメータ グループ - 転送ジョブの管理

(オプション) **-job** パラメータ グループによって、転送ジョブ オブジェクトでほかのメソッドの作成、削除、および呼び出しが行われます。

注：以下のオペランド以外に、転送 (-t) スキップ ロジック オペランド **cskip|fskip=** を使用する場合があります。このオペランドは、順次転送ジョブでの転送にのみ適用されるオプションのパラメータです。連続する転送ジョブでの各転送に対してスキップ ロジックを設定すると、次の *n* 回の転送をスキップしたり、この転送が完了または失敗すると別の転送にスキップできます。

構文形式：

```
dtsccli -job [id=object_id] ["description=description"] [label=label]
        method=method "method_parameters=method_parameters" [sequential=y|n]
        [transfers=transfer_object_ids] [data1=userdata1] [data2=userdata2]
```

注：-job オペランドは、-j と省略できます。

id=object_id

(オプション) 転送メソッド、転送ジョブ メソッド、またはスケジュール メソッドを、1 つ、複数、またはすべての指定された転送、転送ジョブ、またはスケジュールに適切に適用します。

注：既存の転送ジョブの場合のみ必須です。転送ジョブを作成するときにこのオペランドを指定しないでください (method=create)。

デフォルト：なし

例：

- 単一の転送ジョブにメソッド (method=method を参照) を適用します。

```
dtscli -j method=method id=id
```

- 複数の転送ジョブを指定します。id= オペランドまたは label=label オペランドを使用します。id= オペランドの場合、メソッドを適用するときに 2 つ以上の転送ジョブ ID を指定する場合は、id= 式全体を引用符で囲みます。

```
dtscli -j method=method "id=120 130 140"
```

- 2 つの転送 (123 および 124) を転送ジョブ (789) に追加します。

```
dtscli -j id="789" method=add method_parameters="123 124"
```

- すべての転送ジョブにメソッドを適用します。id=all を指定します。

```
dtscli -j method=method id=all
```

"description=description"

(オプション) 転送ジョブの説明を指定します。説明に 1 つ以上のスペースが含まれている場合は、引用符で式全体を囲みます。

限度：1 から 255 文字

デフォルト：なし

label=label

(オプション) 作成する転送ジョブにラベルを指定します。ラベルを割り当てるには、**label=label** オペランドを使用して、数値以外の値を使用してできるだけ論理的で簡単な名前を指定します。オブジェクト作成時にのみ、オブジェクトを作成する **dtscli** コマンドの入力時にラベルを指定できます。

ラベルを使用すると、エイリアス (1 つ以上のオブジェクトを識別するためのオブジェクト ID 以外の代替手段) が提供されます。必要に応じて、複数のオブジェクト ID 値の代わりに単一のラベル値を指定することによって、複数のオブジェクトに同じラベルを付与でき、単一コマンドでこれらのオブジェクトすべてに同じメソッドを適用することができます。

この方法でラベルを使用するには、2 ステップのプロセスが必要です。

- 各オブジェクトに同じラベルを指定するには、**label=label** オペランドを使用して、各オブジェクトを作成します (その結果、各オブジェクトには一意のオブジェクト ID の同じラベルが付与されます)。
- 必要なメソッドをすべてのオブジェクトに適用させるには、**"id=id1 id2 id3..."** 式の代わりに **id=label** を指定します。

例：ラベルが「red」である 3 つのジョブを作成します。

```
dtscli -j label=red -t "ipath=ipath" "rpath=rpath" -mode defer
dtscli -j label=red -t "ipath=ipath" "rpath=rpath" -t "ipath=ipath"
"rpath=rpath" -mode defer
dtscli -j label=red -t "ipath=ipath" "rpath=rpath" -t "ipath=ipath"
"rpath=rpath" -mode defer
```

- 3 つをすべてアクティブにします。

```
dtscli -j id=red method=activate
```

- ラベルを持つすべての転送ジョブに同じメソッドを適用します。

```
dtscli -j method=method label=all
```

注：オブジェクトのグループに同一のラベルを指定し、また同一のメソッドを適用することで、転送、ジョブの転送、およびスケジューリングを行います。

method=method

転送ジョブまたは特定の転送ジョブ内の転送に適用するメソッドを指定します。転送ジョブのメソッドは、以下のとおりです。

注： create 以外のメソッドを呼び出す場合、有効な引数は id= (必須)、method= (必須) です。また、メソッドがパラメータを使用する場合、有効な引数は method_parameters のみです。

abort

実行中の転送ジョブを停止します。

activate

転送ジョブ内のすべての転送を実行します。

追加

指定された転送を転送ジョブに追加します。ジョブ内の転送のリストの最後に追加します。このメソッドはパラメータを使用します。詳細については、「method_parameters=method_parameters」を参照してください。

create

新しい転送ジョブを作成します。

delete

指定された転送ジョブを削除します。

insert

指定された転送を転送ジョブに追加します。ジョブ内の転送のリストの指定した位置に追加します。このメソッドはパラメータを使用します。詳細については、「method_parameters=method_parameters」を参照してください。

削除

指定された転送を転送ジョブから削除します。このメソッドはパラメータを使用します。詳細については、「method_parameters=method_parameters」を参照してください。

reset

転送ジョブ内のすべての転送をリセットします。このアクションを使用すると、**Data Transport Service** を使用するアプリケーションで、定義済みの転送を再利用し、再度起動できます。

再開

suspend メソッドによって中断されたか失敗したものの再起動可能な、転送ジョブ内の転送を再起動します。

suspend

転送ジョブ内の指定された転送を停止します。ただし、転送は後で再開できる状態に保持されます。

status

1 つ以上の転送ジョブのステータス、つまり **転送ステータス** をレポートします。

urgent

ディスクリットおよびデフォルトの転送ジョブにのみ適用されます。指定されたディスクリットまたはデフォルトの転送を緊急の転送にします。このメソッドによって、任意およびすべての転送優先順位の設定、およびディスクリットモードを指定する **DTS** エージェントの **discreet_mode** 設定は上書きされます。その結果、転送は通常転送として送受信されます。

discreet

ジョブ内の指定した任意の転送に適用されます。転送がディスクリット転送として送受信されるようにします。このメソッドによって、任意およびすべての転送優先順位の設定、および通常モードまたは緊急モードを指定する **DTS** エージェントの **discreet_mode** 設定は上書きされます。

転送は、その時点で、転送の存続期間中に **Data Transport Service** によって決定されるディスクリット転送の転送率を使用して処理されます。ディスクリット転送の存続期間は、**delivery** および **expiry** 転送パラメータによって決定されます。

default

該当の転送優先順位の設定をデフォルトにします。その結果、転送にかかわるコンピュータの DTS エージェントの `discreet_mode` 設定によって、どのモード（通常またはディスクリート）が転送の送信に使用され、どのモードが転送の受信に使用されるかが決定されます。

注：urgent、discreet、および default の各メソッドは、アクティブな転送ジョブにのみ適用できます。そのため、転送ジョブを起動して urgent、default、または discreet にするには、以下のコマンドを入力します。

```
dtscli -j id=nnn method=activate
```

```
dtscli -j id=nnn method=urgent|discreet|default
```

"method_parameters=method_parameters"

転送ジョブ メソッドのパラメータを指定します。method_parameters= パラメータは、add、insert、および remove 転送ジョブ メソッドにのみ適用されます。

例に示すように、パラメータ値はスペースで区切り、method_parameters 式全体を引用符で囲みます。

追加

追加する転送オブジェクト ID のリストを指定します。

例：転送ジョブ 220 へ転送 123-126 を追加します。

```
dtscli -j id=220 method=add "method_parameters=123 124 125 126"
```

削除

削除する転送オブジェクト ID のリストを指定します。

以下に例を示します。この例では、転送 123 - 126 を転送ジョブ 220 から削除します。

```
dtscli -j id=220 method=remove "method_parameters=123 124 125 126"
```

insert

新しい転送ジョブを既存の転送ジョブに挿入します。**position** パラメータと共に使用されます。**position** パラメータは、最初に指定される **method_parameter** である必要があります。残りの **method_parameter** は、挿入される転送のオブジェクト ID です。

転送が順次に実行されない場合は、転送ジョブ内での転送の位置は重要ではありません（「**sequential=y|n**」を参照してください）。デフォルトでは、1つの転送ジョブに含まれる転送は、順次に実行されるのではなく同時に実行されます。

例：

- 転送 123 - 126 が転送ジョブ 388 の位置 1（既存のすべての転送の前）に挿入されます。

```
dtsccli -j id=388 method=insert "method_parameters=1 123 124 125 126"
```

- 転送 123 - 126 が転送ジョブ 388 の位置 0 に挿入されます。位置 0 は、これらのオブジェクトが既存のすべての転送の後に、リストに追加されることを意味します。

```
dtsccli -j id=388 method=insert "method_parameters=0 123 124 125 126"
```

position

順次転送ジョブの転送のリストに転送を追加する位置を指定します。位置の番号によって、いつ転送が実行されるかが指定されます。たとえば、順次転送のリストに追加する転送に **position=3** を指定すると、新しい転送は転送ジョブが実行されるときに 3 番目に実行されます。転送のリストの最後に転送を追加するには、**position=0** を指定します。

注： **position** パラメータは、**method_parameters** 引数の最初のパラメータとして指定します。このパラメータは、**insert** メソッドにのみ適用されます。

前述のとおり、転送が順次に実行されない場合は、転送ジョブ内での転送の位置は重要ではありません（「**sequential=y|n**」を参照してください）。デフォルトでは、1つの転送ジョブに含まれる転送は、順次に実行されるのではなく同時に実行されます。

status

転送ジョブのステータス、つまり転送ステータスをレポートします。複数の転送ジョブのステータスを表示するには、以下の形式を使用します。

```
dtscli -j "id=id1 id2 id3 ..." method=status
```

文字列 "id=id1 id2 id3 ..." では、ステータスを表示する転送ジョブのオブジェクト ID を指定します。

例：

- 特定のステータスの転送ジョブのみを表示します。

```
dtscli -j id="id1 id2 id3 ..." method=status  
method_parameters=transfer_state
```

- ステータスが **COMPLETE** の転送ジョブをすべて表示します。

```
dtscli -j id=all method=status "method_parameters=COMPLETE"
```

結果の出力には、**status=COMPLETE** のジョブのみがレポートされます。

注：method_parameters=transfer_state 式では、大文字小文字が区別されます。転送ステータスはすべて大文字で指定します。

デフォルト：なし

sequential=y|n

(オプション) 転送ジョブ内のすべての転送を順次起動して実行するかどうかを指定します。転送ジョブ内のすべての転送を順次（指定した順序で1つずつ）起動して実行する場合は、sequential=y を指定します。「method_parameters=method_parameters」を参照してください。

転送ジョブ内のすべての転送を同時に起動して実行する場合は、sequential=n を指定します。

デフォルト：n

transfers=transfer_object_ids

(オプション) 転送ジョブ内の転送のリストを指定します。転送ジョブが起動されると、各転送が起動されて実行されます。転送はあらかじめ作成されている必要があり、各転送のオブジェクト ID がわかっている必要があります。転送をスペースで区切ってリストにし、引用符で囲みます。

注：method=add、insert、または remove パラメータの場合のみ必須です。

例：転送 110、114、および 118 を含む単一の転送ジョブを作成します。

```
dtsccli -j transfers="110 114 118"
```

注：デフォルトの転送ジョブメソッドは create です。

data1=user_data1

(オプション) ユーザ定義データを転送ジョブと関連付けることができます。

デフォルト：なし

data2=user_data2

(オプション) ユーザ定義データを転送ジョブと関連付けることができます。

デフォルト：なし

関連項目：

[dtsccli コマンド -- 転送ジョブの管理](#) (P. 84)

[dtsccli コマンド - スキップ ロジックの使用](#) (P. 86)

[ディスクリットモード](#) (P. 11)

[転送ステータス](#) (P. 10)

dtscli -schedule パラメータ グループ - スケジュールの管理

(オプション) `-schedule` パラメータ グループによって、スケジュール オブジェクトでほかのメソッドの作成、削除、および呼び出しが行われます。

構文形式：

```
dtscli -schedule [id=object_id] [calendar=y|n] ["calendars=calendars"]  
["calendar_check_times=calendar_check_times"] [description=description]  
[enable=y|n] [label=label] method=method  
["method_parameters=method_parameters"] schedule_time=schedule_time  
["jobs=transfer_job_object_ids"] [valid_from=valid_from] [valid_to=valid_to]  
[data1=user_data1] [data2=user_data2]
```

注： `-schedule` オペランドは、`-s` と省略できます。

id=object_id

(オプション) 転送メソッド、転送ジョブ メソッド、またはスケジュール メソッドを、1 つ、複数、またはすべての指定された転送、転送ジョブ、またはスケジュールに適切に適用します。

注： 既存のスケジュールの場合のみ必須です。 スケジュールを作成するときにこのオペランドを指定しないでください (`method=create`)。

デフォルト： なし

例：

- 単一のスケジュールにメソッド (`method=add|insert|remove` を参照) を適用します。

```
dtscli -s method=method id=id
```

- 複数の ID を指定するには、`id=` オペランドまたは `label=label` オペランドを使用します。 `id=` オペランドの場合、2 つ以上のオブジェクト ID を指定する場合は、`id=` 式全体を引用符で囲みます。

```
dtscli -s method=method "id=520 530 540"
```

- 同じ転送ジョブ (330) を 2 つの異なる転送スケジュール (520 および 530) に追加します。

```
dtscli -s id="520 530" method=add method_parameters="330"
```

- すべてのスケジュールにメソッドを適用します。 `id=all` を指定します。

```
dtscli -s method=method id=all
```

calendar=y|n

(オプション) スケジュールがカレンダー定義に基づくか、1 回限りのスケジュールの起動であるかを指定します。値が **y** の場合、スケジュールはカレンダー定義に基づきます。したがって、関連するカレンダープロパティも設定する必要があります (**calendars**、**valid_from**、**valid_to**、および **calendar_check_times**)。値が **n** の場合、スケジュールは 1 回限りの起動に基づきます。したがって、**schedule_time** パラメータも設定する必要があります。

デフォルト : n

"calendars=calendars"

(オプション) 複雑なスケジュールを定義します。カレンダーをスケジュールに関連付けると、四半期ごと、月次、週次、日次、月の最初の金曜日ごとなど、繰り返しの複雑な起動日時を指定することができます。

有効なカレンダー名をスペースで区切ってリストにし、式全体を引用符で囲みます。

calendars パラメータは、**valid_from** および **valid_to** パラメータと共に使用する必要があります。

デフォルト : なし

"calendar_check_times=calendar_check_times"

(オプション) スケジュールに関連付けられたカレンダーをチェックする時間を指定します。

このプロパティには、時間を早いものから順にスペースで区切り、リストにして指定する必要があります。フォーマットは **HHMMSS** で、式全体を引用符で囲みます。値は、スケジュールに関連付けられたカレンダーをチェックする午前零時以降の時間を示します。

注 : このプロパティが空の場合、スケジュールに関連付けられたカレンダーはチェックされません。

デフォルト : なし

"description=description"

(オプション) スケジュールの説明を指定します。説明に 1 つ以上のスペースが含まれている場合は、引用符で式全体を囲みます。

限度 : 1 から 255 文字

enable=y|n

(オプション) 作成時にスケジュールをオンにするかどうかを指定します。作成時にスケジュールを有効にするには、**enable=y** を指定します。作成時にスケジュールを有効にしない場合は、**enable=n** を指定します。

スケジュールが作成されると、**enable** および **disable** スケジュール メソッドを使用して、スケジュールのステータスを変更できます。

デフォルト : y

label=label

(オプション) 作成するスケジュールにラベルを指定します。ラベルを割り当てるには、**label=label** オペランドを使用して、数値以外の値を使用してできるだけ論理的で簡単な名前を指定します。オブジェクト作成時にのみ、つまり、オブジェクトを作成する **dtscli** コマンドの入力時にラベルを指定できます。

ラベルを使用すると、エイリアス (1 つ以上のオブジェクトを識別するためのオブジェクト ID 以外の代替手段) が提供されます。必要に応じて、複数のオブジェクト ID 値の代わりに単一のラベル値を指定することによって、複数のオブジェクトに同じラベルを付与でき、単一コマンドでこれらのオブジェクトすべてに同じメソッドを適用することができます。

この方法でラベルを使用するには、2 ステップのプロセスが必要です。

- 各オブジェクトに同じラベルを指定するには、**label=label** オペランドを使用して、各オブジェクトを作成します (その結果、各オブジェクトには一意のオブジェクト ID の同じラベルが付与されます)。
- 必要なメソッドをすべてのオブジェクトに適用させるには、"**id=id1 id2 id3...**" 式の代わりに **id=label** を指定します。

デフォルト : なし

例：

- 同じラベル (june) を持つ 2 つのスケジュールを作成します。

```
dtcli -s "jobs=110 120" label=june method=create
```

```
dtcli -s "jobs=130 140" label=june method=create
```

- 同じメソッド (enable) を両方のスケジュールに適用します。

```
dtcli -s id=june method=enable
```

注：オブジェクトのグループに同一のラベルを指定し、また同一のメソッドを適用することで、転送、ジョブの転送、およびスケジューリングを行います。

method=method

スケジュールに適用するメソッドを指定します。

注：create 以外のメソッドを呼び出す場合、有効な引数は id= (必須)、method= (必須) です。また、メソッドがパラメータを使用する場合、有効な引数は method_parameters のみです。

追加

指定されたカレンダーをスケジュールに追加します。スケジュール内のカレンダーのリストの最後に追加します。このメソッドはパラメータを使用します。「method_parameters」を参照してください。

create

新しいスケジュールを作成します。

delete

1 つ以上のスケジュールを削除します。

disable

1 つ以上のスケジュールを無効にします。

enable

1 つ以上のスケジュールを有効にします。

insert

指定されたカレンダーをスケジュールに追加します。スケジュール内のカレンダーのリストの指定した位置に追加します。このメソッドはパラメータを使用します。「method_parameters」を参照してください。

削除

指定されたカレンダーをスケジュールから削除します。このメソッドはパラメータを使用します。「method_parameters」を参照してください。

status

1 つ以上のスケジュールのステータスをレポートします。このメソッドによって示されるのは、スケジュールのステータスであり、スケジュールに関連する転送ジョブのステータスではありません。システムの応答は、初期化済み、成功、または失敗です。このメソッドはパラメータを使用します。「method_parameters」を参照してください。

スケジュールが有効または無効になると、このプロパティは初期化済みにリセットされます。

デフォルト：なし

"method_parameters=method_parameters"

(オプション) スケジュールに適用されるメソッドパラメータを指定します。パラメータはスペースで区切り、method_parameters 式全体を引用符で囲みます。

追加

既存のスケジュールに追加する転送ジョブ ID のリストを指定します。

例：転送ジョブ 123 - 126 をスケジュール 220 に追加します。

```
dtscli -s id=220 method=add "method_parameters=123 124 125 126"
```

削除

既存のスケジュールから削除する転送ジョブ ID のリストを指定します。

例：転送ジョブ 123 - 126 をスケジュール 220 から削除します。

```
dtscli -s id=220 method=remove "method_parameters=123 124 125 126"
```

insert

新しい転送ジョブを既存のスケジュールに挿入します。 **position** パラメータと共に使用します。

例：

- 転送ジョブ **123 - 126** がスケジュール **388** の位置 **1**（既存のすべての転送ジョブの前）に挿入されます。

```
dtcli -s id=388 method=insert "method_parameters=1 123 124 125 126"
```

- 転送ジョブ **123 - 126** がスケジュール **388** の位置 **0** に挿入されます。位置 **0** は、これらのオブジェクトが既存のすべての転送ジョブの後に、リストに追加されることを意味します。

```
dtcli -s id=388 method=insert "method_parameters=0 123 124 125 126"
```

position

1 つ以上の新しい転送ジョブが挿入される、既存のスケジュール内の位置を指定します。 **position** パラメータは、最初に指定される **method_parameter** である必要があります。残りの **method_parameter** は、挿入される転送ジョブのオブジェクト ID です。 **position** パラメータは、**insert** メソッドにのみ適用されます。

status

スケジュールのステータスをレポートします。 **method=status** 式のみを使用して、スケジュールのステータスをレポートできます。また、特定のステータスのみをレポートするために、 **method_parameters=I|F|S** 式を追加できます。有効な値は、**I**= 初期化済み、**S**= 成功、および **F**= 失敗です。以下のフォーマットを使用します。

```
dtcli -s id=schedule_id method=status "method_parameters=I|F|S"
```

指定したメソッドパラメータ（ステータス）と一致した場合、**Data Transport Service** によってスケジュールのステータスが表示されます。

デフォルト：なし

schedule_time=schedule_time

スケジュールを起動する日時を指定します。schedule_time パラメータは、スケジュールを 1 回のみ起動する必要がある場合に使用されます。schedule_time が設定された場合、calendar パラメータを N に設定する必要があります。スケジュールを定期的に起動する必要がある場合、calendars パラメータを使用する必要があります、calendar パラメータを Y に設定する必要があります。

デフォルト：なし

"jobs=transfer_job_object_ids"

(オプション) スケジュール オブジェクト内の転送ジョブ オブジェクトを指定します。スケジュールに関連付けられた calendars または schedule_time によって指定された日時にスケジュールが起動されると、これらの転送ジョブが起動されます。

転送ジョブ オブジェクト ID をスペースで区切ってリストにし、式全体を引用符で囲みます。以下の形式を使用します。

```
dtscli ... "jobs=132 154 185"
```

注：転送ジョブをスケジュールに追加する前に、転送ジョブを作成し、それらのオブジェクト ID に注意します。

デフォルト：なし

valid_from=valid_from

(オプション) スケジュール オブジェクトのカレンダ指定が検証される期間の開始を指定します。

valid_from パラメータは、valid_to カレンダおよび calendars パラメータと共に使用する必要があります。valid_from パラメータは、valid_to パラメータよりも前の時間を指定する必要があります。

デフォルト：なし

valid_to=valid_to

(オプション) スケジュール オブジェクトのカレンダ指定が検証される期間の終了を指定します。

valid_to パラメータは、valid_from カレンダおよび calendars パラメータと共に使用する必要があります。valid_to パラメータは、valid_from パラメータよりも後の時間を指定する必要があります。

デフォルト：なし

`data1=user_data1`

(オプション) ユーザ定義データをスケジュールと関連付けることができます。

デフォルト: なし

`data2=user_data2`

(オプション) ユーザ定義データをスケジュールと関連付けることができます。

デフォルト: なし

dtsccli -log パラメータ グループ - ログ レベルおよびログ モードの指定

-log パラメータ グループによって、転送が同期して起動されるか非同期で起動されるか、および必要なログのレベルを指定します。

構文形式:

```
dtsccli -log interval=interval level=level mode=append|replace monitor=y|n path=path
```

注: -log オペランドは、-l と省略できます。

`interval=interval`

転送を監視しているときに進捗情報メッセージがログされるパーセント間隔を指定します。

デフォルト: 10

例: 転送が 10 パーセント完了するたびにコンソールにメッセージを書き込むように指定します。

```
dtsccli ... -l interval=10
```

転送が起動された後、累積されたシステム応答は、転送が 30 パーセント完了した後では以下のように表示されます。

```
転送 457 が起動されました ...
    10% 完了 ...
    20% 完了 ...
    30% 完了 ...
```

level=*level*

ログ レベルを指定します。有効な値は以下のとおりです。

0 = サイレント

1 = 最小

指定した場合、オブジェクトの作成および削除メッセージは、このレベルでは記録されません。

2 = 中

3 = 最大

デフォルト : 1

mode=append|replace

ログ モードを指定します。有効な値は以下のとおりです。

append

ログ メッセージを既存のファイルに追加します。

replace

既存のログ ファイルを置換します。

デフォルト : append

monitor=*y*|*n*

転送の進捗状況を監視するかどうかを指定します。有効な値は、*y* (yes) および *n* (no) です。*y* の場合、転送は監視され、転送の進捗状況がコンソールに書き込まれます。転送が完了または失敗するまで、コマンドラインでコマンドを入力することができません。

デフォルト : *y*

path=*path*

ログ メッセージが書き込まれるファイルの場所を指定します。パス名を入力する場合は、引用符で囲む必要があります。たとえば、以下のよう指定します。

```
dtsccli -l path="c:\logs\080100.log"
```

標準出力のパス (stdout) を入力する場合、シェルで別の場所を指定していない限り、出力はコンソールに表示されます。

デフォルト : stdout

dtcli -agent パラメータ グループ - エージェント間転送の実行

(オプション) `-agent` パラメータ グループでは、エージェント間転送を実行します。

構文形式：

```
dtcli -agent [broadcast_address=broadcast_address] [checkpoint=y|n] [commit=y|n]
[delimiter_conversion=y|n] [delivery=discreet_delivery_period]
[direction=s|r] [expiry=discreet_expiry_period] ["f_filters=filters"]
["p_filters=filters"] [hop_hosts=hop_hosts] "ipath=host::path"
[ipriority=urgent|default|discreet] [iuser=username::password]
[multicast_address=multicast_address] [output_mode=a|c|r|s|w]
[parcel_size=parcel_size] [protocol=protocol]
[protocol_parameters=protocol_parameters]
[retry_interval=session_retry_interval] [retry_limit=session_retry_limit]
"rpath=host::path" [rpriority=urgent|default|discreet]
[ruser=username::password] [throttle=throttle] [transfer_mode=transfer_mode]
```

注： `-agent` オペランドは、`-a` と省略できます。

重要： DTS は、IPv4 ブロードキャストおよび IPv4/IPv6 マルチキャストのアドレス指定をサポートします。BCAST 一対多プロトコルは、IPv4 アドレスでのみ使用できます。ブロードキャストタイプの転送を IPv6 ネットワーク経由で実行する場合、関連する IPv6 マルチキャストアドレスと共に MCAST プロトコルを使用します。

broadcast_address=broadcast_address

(オプション) IPv4 ブロードキャスト (bcast) アドレスを指定します。ブロードキャスト転送にのみ有効です。このパラメータを指定する場合は、`transfer_mode=bcast` および `protocol=BCAST` も指定します。

注： 実行される転送のタイプは転送モードに依存します。転送モードが `bcast` または `mccast` の場合、UDP ブロードキャストまたはマルチキャスト転送です。

注： ブロードキャストアドレスおよびマルチキャストアドレスを計算する手順については、「*Data Transport Service 管理ガイド*」を参照してください。

デフォルト： なし

例：ブロードキャストを IP アドレス 172.24.255.255 に送信するには、ブロードキャスト（bcast）の転送モードを指定します。

```
dtscli -a "ipath=steve::c:¥steve¥memo2.doc"
        "rpath=katie::c:¥katie¥memo2.doc"
        broadcast_address=172.24.255.255
        transfer_mode=bcast protocol=bcast
```

ブロードキャストでは、ブロードキャスト アドレスへの 1 つのデータ送信のみが必要です。パケットは、ブロードキャスト アドレス（172.24.255.255）によって識別されるサブネット（172.24.0.0）のすべてのコンピュータにブロードキャストされます。ブロードキャストをリスニングしているそれぞれのコンピュータが、パケットを受信します。

checkpoint=y|n

（オプション）チェックポイントの再開機能（Data Transport Service で使用されるエラー回復メカニズム）をオンにするかどうかを指定します。有効な値は、y（yes）および n（no）です。チェックポイントの再開が有効な場合は、転送中に障害が発生すると、障害が発生した時点から転送の再開が試みられます。チェックポイントの再開が無効な場合は、転送中に障害が発生すると、転送は最初から再開されます。

デフォルト：なし

commit=y|n

（オプション）ディスクへのコミット機能をオンにするかどうかを指定します。有効な値は、y（yes）および n（no）です。有効な場合、すべてのデータのパーセルを受信した後、DTS エージェントによってファイル出力バッファがフラッシュされハードディスクに書き込まれます。ディスクへのコミットを指定しない（n）場合、DTS エージェントでは、オペレーティング システムのファイル I/O に依存して、必要時にバッファのフラッシュが実行されます。

デフォルト：なし

delimiter_conversion=y|n

(オプション) 区切り文字の変換をオンにするかどうかを指定します。有効な値は、**y** (yes) および **n** (no) です。受信コンピュータが送信コンピュータと異なるプラットフォームで実行されている場合でも、入力ファイル名と同じ形式で出力ファイル名を指定するときのインスタンスに対してのみ、区切り文字の変換 (**y**) を使用します。このようなプラットフォーム間転送では、区切り文字の変換によって、受信プラットフォームに転送されたデータの出力パス名が正しく区切られます。

関連するプラットフォームで同様の（完全に同じではない）区切り文字およびファイル命名規則を使用している場合は、区切り文字の変換がプラットフォーム間転送で使用されます。最も一般的な用途には、**Windows** と **UNIX/Linux** コンピュータとの間の 1 対多の転送があります。これ以外のプラットフォーム間転送もサポートされています。

転送の作成または更新中に、転送の入力パス名および出力パス名を完全かつ正確に指定した場合は、区切り文字の変換を使用しないでください。プラットフォーム間転送の入力ファイル名として、同じ形式の出力ファイル名を指定する場合にのみ、区切り文字の変換を使用します。

注： 区切り文字の変換は注意して使用してください。受信プラットフォームのファイル命名規則または区切り文字規則に対するエラーチェックまたは検証メカニズムとしては機能しません。

デフォルト： なし

例： Windows および UNIX システムへのファンアウト転送

```
dtsccli -a "ipath=c:¥input.txt"
        "rpath=unixserver1::c:¥tmp¥output.txt"
        "rpath=winserver1::c:¥tmp¥output.txt"
delimiter_conversion=y
iuser=administrator::adminpass
ruser=root::rootpass
transfer_mode=fanout
```

delivery=discreet_delivery_period

(オプション) ディスクリート配信期間を分単位で指定します。転送の送信 (ipriority 使用) または転送の受信 (rpriority 使用) のどちらか、または両方に対して転送優先順位「discreet」を指定した場合にのみ、このオプションを適用できます。転送の送信および受信の両方にディスクリートモードを指定すると、同じディスクリート配信期間が両方に適用されます。

ディスクリート配信期間には、DTS エージェントの開始または応答によって、ディスクリートモードでの転送の送信または受信を試みる期間の合計が指定されます。ディスクリート送信またはディスクリート受信がこの期間内に完了しない場合、転送の優先順位は自動的に **discreet** から **urgent** に変更されます。(転送の優先順位の設定については、ipriority および rpriority オペランドの記述で説明されています)。

転送が起動されたとき (作成されたときではない) に時間の計測が開始されます。

ディスクリート配信期間 (delivery) は、通常、ディスクリート有効期限 (expiry) と共に使用されます。ディスクリート配信期間は、ディスクリート有効期限以下にする必要があります。これらのオプション (delivery または expiry) の片方のみを指定した場合、指定しないオプションにはデフォルト値が使用されます。

デフォルト: -1 (無制限)

例: ディスクリート配信期間が 360 分 (6 時間)、およびディスクリート有効期限 (expiry) が 720 分 (12 時間) でディスクリート転送を指定する場合は、以下を入力します。

```
dtscli -a|t "ipath=ukwin4q::c:¥winnt¥tngdts.ini"
            "rpath=philauk::c:¥config¥ukwinp4q¥tngdts.ini"
            ipriority=discreet rpriority=discreet
            delivery=360 expiry=720
```

direction=s|r

(オプション) 転送時に開始コンピュータがデータを受信または送信するかどうかを指定します。有効な値は以下のとおりです。

s

開始側から応答側にデータを送信します。

r

応答側から開始側にデータを送信します。

注： **direction=r** オプションを使用する場合、ソース ファイルは受信コンピュータ上にあり、宛先ファイルは開始コンピュータ上にあります (通常、その反対もあります)。

デフォルト： s

例： 以下の管理対象転送の例では、開始コンピュータ (NYC_HQ) は応答側 (STORE888) からデータを受信します。

```
dtsccli -a|t "ipath=NYC_HQ::c:¥receipts¥store888¥010398.rcp"  
"rpath=STORE888::PDEV.RRARI09.RECEIPTS" direction=r
```

expiry=discreet_expiry_period

(オプション) ディスクリット有効期限を分単位で指定します。転送の送信 (ipriority 使用) または転送の受信 (rpriority 使用) のどちらか、または両方に対して転送優先順位「discreet」を指定した場合にのみ、このオプションを適用できます。転送の送信および受信の両方にディスクリットモードを指定すると、同じディスクリット有効期限が両方に適用されます。

転送の期限が切れると、転送は削除され、送信の再試行は発生しません。有効期限 (expiry) は、ディスクリット配信期間 (delivery) 以上にする必要があります。これらのオプション (expiry または delivery) の片方のみを指定した場合、指定しないオプションにはデフォルト値が使用されます。

転送が起動されたとき (作成されたときではない) に時間の計測が開始されます。

デフォルト: -1 (無制限)

以下に例を示します。この例では、ディスクリット配信期間 (delivery) が 180 分 (3 時間)、およびディスクリット有効期限が 360 分 (6 時間) のディスクリット転送を指定しています。

```
dtscli -a|t "ipath=ukwin4q::c:%winnt%tngdts.ini"
            "rpath=philauk::c:%config%ukwinp4q%tngdts.ini"
            ipriority=discreet rpriority=discreet delivery=180 expiry=360
```

"f_filters=filters"

"p_filters=filters"

(オプション) 転送のフィルタのタイプを指定します。ファイルフィルタ (f_filters) では、一度に 1 ファイルずつ転送データが処理されます。パーセルフィルタ (p_filters) では、一度に 1 パーセルずつ転送データが処理されます (パーセルは、ファイル内のより小さいデータ単位です)。

通常、フィルタ パラメータを指定するのはエージェント間転送のみです。管理対象転送では指定しません。ただし、暗号化フィルタだけは例外です。パラメータが必要なほかのすべてのフィルタの場合は、管理対象転送によって転送オブジェクト サーバで指定された値が自動的に取得されます。

ファイルを指定するには、以下の形式を使用します。

```
dtcli -a|t "ipath=ipath" "rpath=rpath" ...  
      "f_filters=file_filters"
```

パーセルを指定するには、以下の形式を使用します。

```
dtcli -a|t "ipath=ipath" "rpath=rpath" ...  
      "p_filters=parcel_filters"
```

単一の読み取りフィルタや書き込みフィルタ、または読み取りフィルタと書き込みフィルタが一致するペアを指定できます。

"f_filters=file_filters" または "p_filters=parcel_filters" 式全体を引用符で囲みます。複数のフィルタをスペースで区切ります。

フィルタ指定の各タイプには、以下で示す位置にコロン (:) が必要です。

- 読み取りフィルタおよび一致する書き込みフィルタを指定するには、以下を入力します。

```
"f|p_filters=read_filter_name[(read_filter_params)]  
      :write_filter_name[(write_filter_params)]"
```

- 一致する書き込みフィルタがない読み取りフィルタを指定するには、以下を入力します。

```
"f|p_filters=read_filter_name[(read_filter_params)]:"
```

- 一致する読み取りフィルタがない書き込みフィルタを指定するには、以下を入力します。

```
"f|p_filters=:write_filter_name[(write_filter_params)]"
```

フィルタは以下の順序で適用されます。

1. ファイル読み取りフィルタ
2. パーセル読み取りフィルタ
3. パーセル書き込みフィルタ
4. ファイル書き込みフィルタ

同じタイプの読み取りフィルタ（ファイルまたはパーセル）は、コマンドラインで指定する場合と同じ順序で適用されます。同じタイプの書き込みフィルタは、コマンドラインで指定する場合と反対の順序で適用されます。

デフォルト： バイナリ読み取りおよび書き込みフィルタ

以下に例を示します。次のコマンドでは、以下の順序で指定されたフィルタが適用されます。

- バイナリ読み取りパーセル フィルタ
- Huffman 圧縮パーセル フィルタ
- Huffman 解凍パーセル フィルタ
- バイナリ書き込みパーセル フィルタ

```
dtscli -a "ipath=ukwign3n::c:%temp%getstart.pdf"
          "rpath=ukwign3n::c:%temp%getstart2.pdf"
          "p_filters=BINARY_READ:BINARY_WRITE
          HUFFMAN_COMPRESS:HUFFMAN_UNCOMPRESS"
```

この例では、どのフィルタにもパラメータがありません。

注： 各転送には、テキスト フィルタとバイナリ フィルタのいずれかを追加する必要があります。フィルタを指定する場合、バイナリ フィルタはデフォルトで適用されません。したがって、同じコマンドでテキスト フィルタとバイナリ フィルタのいずれかを明示的に指定する必要があります。

hop_hosts=hop_hosts

（オプション）すべての出力ファイルについて、ホスト コンピュータを経由するホップを実行します。たとえば、ターゲットの受信コンピュータ（エンド ノード）が開始側に直接はアクセスできないものの、ホップ コンピュータを経由してアクセスできる場合には、ホップを指定します。

1 つ以上の追加のホップ コンピュータを指定する必要がある場合は、各コンピュータ名を入力します。コンピュータ名は空白で区切ります。各ホップ コンピュータに、転送を受け付ける十分なディスク領域が必要です。不十分な場合、転送は失敗します。

実行されると、転送はコマンドラインで指定した順序でホップ コンピュータを経由します。1 つの転送に最大 **100** の出力ファイルを指定することができ、転送ごとに最大 **25** のホップ コンピュータを指定することができます。

注：（開始側が実行されているプラットフォームと）異なるプラットフォームへのホップを指定し、指定した転送プロトコルがホップ コンピュータが実行されているプラットフォームでサポートされていない場合、転送は失敗します。すべてのプラットフォームでサポートされているプロトコルの詳細については、「protocol=」を参照してください。

例：transfer_mode にポイント ツー ポイント (p2p) を指定し、2 つの出力ファイルについてコンピュータ 172.24.213.47 を経由するホップを指定します。

```
dtsccli -a "ipath=ddilgp78::c:¥simpex.txt" hop_hosts=172.24.213.47
          "rpath=GGilyphy::c:¥qa¥in¥simpex.txt"
          "rpath=GGilyphx::c:¥qa¥in¥simpex.txt"
          transfer_mode=p2p
```

"ipath=host::path"

転送の入力ファイルを引用符で囲んで指定します。ホスト名の後ろにはコロンを 2 個と、転送のフルパス名を続けます。以下に例を示します。

- Windows

```
"ipath=mmmww185::c:¥receipts¥010398.rcp"
```

- UNIX/Linux

```
"ipath=machj17::/home/receipts/010398.rcp"
```

- HTTP 転送 (インターネット ダウンロード)

```
"ipath=DTA_host_name::destination_path"
```

注：ipath= operand が必須です。

host

コンピュータ名、IP アドレス、または IPX アドレスを指定します。IP アドレスのセパレータにはピリオド (.) を使用します。たとえば、172.24.255.255 のようになります。IPX アドレスのセパレータにはピリオドまたはコロン (:) を使用します。たとえば、00000005.0000000000001 または 00000005:0000000000001 のようになります。

開始側として動作する DTS エージェントは、このコマンドラインユーティリティからリクエストを受信し、ターゲットの応答側または応答側グループへの接続を確立し、データを送信します。開始側は、転送を開始するコンピュータです。ホスト名を省略すると、ローカル ホストがデフォルトで使用されます。

path

入力データのフルパス名を指定します。

例：

- canoes という Windows コンピュータから tipper という別の Windows コンピュータに管理対象転送を送信する場合は、以下のコマンドを入力します。

```
dtscli -a|t "ipath=canoes::C:¥My Documents¥Monthly Reports¥apr2008.doc"
"rpath=tipper::C:¥Monthly Reports¥David¥apr2006.doc"
```

- CHICAGO12 と NEWYORK20 という 2 台の UNIX/Linux コンピュータ間で管理対象転送を送信するには、UNIX/Linux プロンプトで以下のコマンドを入力します。

```
dtscli -a|t "ipath=CHICAGO12::/home/ptfiles/ptcr1000.txt"
"rpath=NEWYORK20::/home/ptfiles/ptcr1000.txt"
```

- Miami という Windows コンピュータから NEWYORK20 という UNIX コンピュータに管理対象転送を送信する場合は、以下のコマンドを入力します。

```
dtscli -a|t "ipath=Miami::C:¥outgoing¥file.txt"
"rpath=NEWYORK20::/home/incoming/file.txt"
```

注：UNIX/Linux 上で dtscli コマンドを実行する場合、円記号は、「エスケープ」文字になることに注意してください。Windows パスを指定するには、円記号を 2 つ続ける必要があります。以下に例を示します。

```
dtscli -a|t "ipath=Miami::C:¥¥outgoing¥¥file.txt"
"rpath=NEWYORK20::/home/incoming/file.txt"
```

ipriority=urgent|default|discreet

(オプション) 転送の開始コンピュータに転送の優先順位を指定します。優先順位は、以下で説明されているように **urgent**、**default**、または **discreet** です。開始コンピュータに対してここで指定する転送の優先順位は、受信コンピュータに転送の優先順位 (**rpriority**) を設定し、転送の開始および受信コンピュータにエージェントの **DiscreetMode** (通常またはディスクリート) を設定した状態で機能します。(エージェントの **DiscreetMode** の設定は、**Data Transport** エージェントのパラメータの 1 つであり、**DSM** エクスプローラを使用して設定することもできます。)

注：詳細については、「**DSM** エクスプローラ ヘルプ」の「設定ポリシー」にある「データ転送エージェント プラグイン ポリシー グループ」のトピックを参照してください。

以下の転送優先順位の設定およびエージェントの **DiscreetMode** 設定の両方により、どのモード（通常またはディスクリート）が転送の送信に使用され、およびどのモードが転送の受信に使用されるかが決定されます。

urgent

開始コンピュータのエージェントの **DiscreetMode** の設定が **Y**（はい）の場合でも、通常モードで転送が送信されます。

default

開始コンピュータのエージェントの **DiscreetMode** の設定が **N**（いいえ）の場合に通常モードで転送が送信されます。ただし、開始コンピュータのエージェントの **DiscreetMode** 設定が **Y**（はい）の場合、転送はディスクリートモードで送信されます。

discreet

開始コンピュータのエージェントの **DiscreetMode** の設定が **N**（いいえ）の場合でも、ディスクリートモードで転送が送信されます。

ディスクリート転送はバックグラウンドで送受信され、また送受信するコンピュータの負荷が大きい場合は、最小限の帯域幅を使用します。DTS エージェントでは、ディスクリート転送によるコンピュータのパフォーマンスとユーザの生産性への影響を最小限に抑えることができるよう、負荷に基づいて、転送を送信するタイミングを決定し、最適な転送速度を計算します。

デフォルト : default

以下に例を示します。ディスクリート配信期間 (**delivery**) およびディスクリート有効期限 (**expiry**) にデフォルト値を使用して、ディスクリートモードで転送を送信および受信 (**rpriority**) するには、以下のコマンドを入力します。

```
dtsccli -a|t "ipath=ukwin4q::c:¥winnt¥tngdts.ini"
            "rpath=philauk::c:¥config¥ukwinp4q¥tngdts.ini"
            ipriority=discreet rpriority=discreet
```

iuser=username::password

(オプション) 開始コンピュータのユーザ名およびパスワードを指定します。

注：転送に関連する DTS エージェントのセキュリティ モードが「fail」に設定されている場合、このパラメータは必須です。

デフォルト：なし

以下に例を示します。この例では、開始コンピュータ (ipath iuser) と受信元コンピュータ (rpath ruser) の両方に対してユーザ名およびパスワードを指定しています。

```
dtscli -a|t "ipath=NYC_HQ::c:%receipts%store888¥010398.rcp"
iuser=Frank::furter
"rpath=STORE888::PDEV.RRARI09.RECEIPTS" ruser=Molly::pitcher
```

multicast_address=multicast_address

(オプション) IPv4 または IPv6 マルチキャスト (mcast) アドレスを指定します。Data Transport Service で使用する有効なマルチキャストアドレスの範囲は、224.0.1.0 - 238.255.255.255 (「グローバル スコープ アドレス」) です。UDP マルチキャスト転送にのみ有効です。このパラメータを指定する場合は、transfer_mode=mcast および protocol=MCAST も指定します。

注：ブロードキャスト アドレスおよびマルチキャスト アドレスを計算する手順については、「*Data Transport Service 管理ガイド*」を参照してください。

デフォルト：なし

例：

- マルチキャストを IP アドレス 224.1.2.3 に送信し、mcast の転送モードを指定します

```
dtscli -a "ipath=steve::c:%steve¥memo2.doc"
"rpath=katie::c:%katie¥memo2.doc"
multicast_address=224.1.2.3
transfer_mode=mcast protocol=mcast
```

- ブロードキャスト タイプの転送を IPv6 ネットワーク経由で FF15:0:1:1:1::45 に送信し、mcast の転送モードを指定します。

```
dtscli -a "ipath=steve::c:%steve¥memo2.doc"
"rpath=katie::c:%katie¥memo2.doc"
multicast_address=FF15:0:1:1:1::45
transfer_mode=mcast protocol=mcast
```

output_mode=a|c|r|s|w

(オプション) 出力モードを指定します。有効な値は以下のとおりです。

a

既存ファイルにデータを追加します。ファイルがない場合は、転送に失敗します。

c

出力ファイルを新規作成します。すでにファイルがある場合は、転送に失敗します。

r

既存ファイルを置き換えます。すでにファイルがない場合は、転送に失敗します。

s

出力ファイルを新規作成します。すでにファイルがある場合は、既存ファイルにデータが追加されます。

w

出力ファイルを新規作成します。すでにファイルがある場合は、置き換えられます。

デフォルト : w

parcel_size=parcel_size

(オプション) この転送で DTS エージェントを開始することによって送信できる最大パーセルサイズ(バイト単位)を指定します。最小パーセルサイズは 128 バイトであり、理論的には、最大パーセルサイズは使用可能な仮想メモリの大きさによってのみ制限されます。

デフォルト : 524,288 バイト

以下に例を示します。この例では、benjamin という名前の Windows コンピュータから joseph という Windows コンピュータに、最大パーセルサイズ 50,000 バイトを使用してポイント ツー ポイント転送を送信します。

```
dtsccli -a "ipath=benjamin::c:¥nhmh.c" parcel_size=50000  
"rpath=joseph::c:¥nhmh.c"
```

`protocol=protocol`

(オプション) エージェント間転送のプロトコルを指定します。プロトコルには、パラメータが必要な場合があります。詳細については、`protocol_parameters` オペランドを参照してください。

有効な値は以下のとおりです。

TCP

[Transmission Control Protocol \(伝送制御プロトコル\)](#) (P. 106)を指定します。

UDP

[User Datagram Protocol \(ユーザデータグラムプロトコル\)](#) (P. 110)を指定します。

SPX

[Sequenced Packet Exchange](#) (P. 112) を指定します。

CPIC

[Common Programming Interface for Communications](#) (P. 111) を指定します。

PPP

[Point to Point Protocol \(ポイントツーポイントプロトコル\)](#) (P. 108)を指定します。一般電話回線、ISDN、その他の高速回線上でのポイントツーポイント転送にのみ有効です。

HTTP

[Hypertext Transfer Protocol \(ハイパーテキスト転送プロトコル\)](#) (P. 97)を指定します。HTTP 転送 (インターネットダウンロード) には、特定の構文形式が必要です。

BCAST

ブロードキャストを指定します。このプロトコルを使用する場合は、`broadcast_address=broadcast_address` および `transfer_mode=bcast` も指定する必要があります。(IPv4 でのみ使用できます。)

MCAST

マルチキャストを指定します。このプロトコルを使用する場合は、`multicast_address=multicast_address` および `transfer_mode=mcast` も指定する必要があります。(IPv4 および IPv6 で使用できます。)

デフォルト : TCP

注：必要に応じて、dtcli コマンドの `protocol=` および `protocol_parameters` オペランドを使用して、特定の転送のみに使用されるプロトコルおよびプロトコルパラメータを設定できます。ただし、特定のコンピュータに関するすべての転送のデフォルトのプロトコルおよびデフォルトのプロトコルパラメータを設定するには、WorldView または DSM エクスプローラのいずれかを使用して、コンピュータのプロトコル関連の設定を変更する必要があります。WorldView を使用してこれらの設定を変更する手順については、「*Data Transport Service WorldView Administration Client* ヘルプ」を参照してください。DSM エクスプローラを使用した手順については、「*DSM エクスプローラ ヘルプ*」の「設定ポリシー」を参照してください。

`protocol_parameters=protocol_parameters`

(オプション) プロトコル固有のパラメータを指定します。通常、以下の要件および制限が適用されます。

- アドレス関連のパラメータが必要です。
- Data Transport Service によって使用されるデフォルトのポート番号をチェックし、すでに使用されているポート番号と競合しないことを確認する必要があります。競合がある場合は、Data Transport Service のポート番号を使用されていないポート番号に変更します。
- CPI-C プロトコルを使用することによってのみコンピュータがほかのコンピュータに接続できる場合、そのコンピュータは dtcli コマンドを使用したデータ転送において応答側としてのみ機能します。

必要なパラメータなどの詳細については、必要なプロトコルを選択してそのパラメータを表示してください。

[HTTP](#) (P. 97)

[PPP \(Windows\)](#) (P. 108)

[PPP \(Solaris Solstice\)](#) (P. 108)

[CPIC](#) (P. 111)

[TCP](#) (P. 106)

[UDP](#) (P. 110)

[SPX](#) (P. 112)

`retry_interval=session_retry_interval`

(オプション) セッション再試行の間隔 (コンピュータへの接続中に転送の失敗が発生した場合の再試行の分数) を指定します。

デフォルト : 60

`retry_limit=session_retry_limit`

(オプション) セッション再試行の制限 (コンピュータへの接続中に転送の失敗が発生した場合に実行する再試行の回数) を指定します。

デフォルト : 20

`"rpath=host::path"`

引用符で囲まれた転送の出力ファイルを指定します。ホスト名の後ろにはコロンを 2 個と、転送のフルパス名を続けます。以下に例を示します。

- Windows

`"rpath=mmmww188::c:¥receipts¥store888¥010398.rcp"`

- UNIX/Linux

`"rpath=uxmnuu18::/home/receipts/store888/010398.rcp"`

- HTTP 転送 (インターネット ダウンロード)

`"rpath=http_host_name::http_source_file_path_name"`

注 : `rpath= operand` が必須です。

host

コンピュータ名、IP アドレス、または IPX アドレスを指定します。IP アドレスのセパレータにはピリオド (.) を使用します。たとえば、172.24.255.255 のようになります。IPX アドレスのセパレータにはピリオドまたはコロン (:) を使用します。たとえば、00000005.000000000001 または 00000005:000000000001 のようになります。

応答側として動作する DTS エージェントでは、受信接続をリスニングおよび応答します。これらの受信接続 (受信データ転送) は、DTS エージェントによって開始されます。

path

受信 DTS エージェントで使用する出力データの完全パス名を指定します。

注： UNIX/Linux から Windows へのデータ転送に dtsccli コマンドを使用するときに、出力パスを引用符で囲まない場合は、各ディレクトリおよびパスを区切るために円記号を 2 つ入力する必要があります。

例：

- この UNIX/Linux から Windows の例では、rpath が引用符で囲まれているため、必要な円記号の区切り文字は 1 つのみです。

```
dtsccli -a|t "ipath=uxmnjj17::/home/receipts/010398.rcp"  
"rpath=mmmww188::c:¥receipts¥store888¥010398.rcp"
```

- この例では、引用符を使用して各入力および出力パスの式を囲っていないため、円記号が 2 つ必要です。

```
dtsccli -a|t ipath=iuxmnjj17::/home/receipts/010398.rcp  
rpath=mmmww188::c:¥¥receipts¥¥store888¥¥010398.rcp
```

注： 各ディレクトリやパスを区切るための 2 つの円記号は、UNIX/Linux から Windows への転送の出力パスでのみ必要です。

rpriority=urgent|default|discreet

(オプション) 転送の受信コンピュータに転送の優先順位を指定します。優先順位は、以下で説明されているように **urgent**、**default**、または **discreet** です。ここで指定する転送優先順位は、開始コンピュータの転送優先順位の設定 (**ipriority**) および転送の開始コンピュータおよび応答コンピュータのエージェントの **DiscreetMode** 設定 (通常またはディスクリート) と連携します。(エージェントの **DiscreetMode** の設定は、**Data Transport** エージェントのパラメータの 1 つであり、DSM エクスプローラを使用して設定することもできます。)

注：詳細については、「*DSM エクスプローラ ヘルプ*」の「設定ポリシー」にある「データ転送エージェント プラグイン ポリシー グループ」のトピックを参照してください。

以下の転送優先順位の設定およびエージェントの **DiscreetMode** 設定の両方により、どのモード (通常またはディスクリート) が転送の送信に使用され、およびどのモードが転送の受信に使用されるかが決定されます。

urgent

応答コンピュータのエージェントの **DiscreetMode** の設定が **Y** (はい) の場合でも、通常モードで転送が受信されます。

default

応答コンピュータのエージェントの **DiscreetMode** の設定が **N** (いいえ) の場合に通常モードで転送が受信されます。ただし、応答コンピュータのエージェントの **DiscreetMode** 設定が **Y** の場合、転送はディスクリート モードで受信されます。

discreet

応答コンピュータのエージェントの **DiscreetMode** の設定が **N** の場合でも、ディスクリート モードで転送が受信されます。

ディスクリート転送はバックグラウンドで送受信され、また送受信するコンピュータの負荷が大きい場合は、最小限の帯域幅を使用します。**DTS** エージェントでは、ディスクリート転送によるコンピュータのパフォーマンスとユーザの生産性への影響を最小限に抑えることができるよう、負荷に基づいて、転送を送信するタイミングを決定し、最適な転送速度を計算します。

デフォルト : default

以下に例を示します。ディスクリット配信期間 (delivery) およびディスクリット有効期限 (expiry) にデフォルト値を使用して、ディスクリットモードで転送を送信 (ipriority) および受信するには、以下のコマンドを入力します。

```
dtsccli -a|t "ipath=ukwin4q::c:¥winnt¥tngdts.ini"
            "rpath=philauk::c:¥config¥ukwinp4q¥tngdts.ini"
            ipriority=discreet rpriority=discreet
```

ruser=username::password

(オプション) 応答コンピュータのユーザ名およびパスワードを指定します。

注: 転送に関連する DTS エージェントのセキュリティ モードが「失敗」に設定されている場合、このパラメータは必須です。

デフォルト: なし

例: 以下の例では、開始コンピュータ (ipath) と受信側コンピュータ (rpath) の両方に対してユーザ名およびパスワードが指定されています。

```
dtaccli -a|t "ipath=NYC_HQ::c:¥receipts¥store888¥010398.rcp"
iuser=polly::nomial ¥
            "rpath=STORE888::PDEV.RRARI09.RECEIPTS" ruser=redbird::bdfeeder ... ¥
```

throttle=throttle

(オプション) スロットル係数を指定します。このコンピュータからデータ転送を開始したときに使用されるネットワーク帯域幅の量です。

有効な値は、0 (スロットリングなし、つまり、パーセル送信間に遅延の強制はありません) - 100 です。1 増やすごとに、パーセル送信操作間の遅延間隔が 50 ミリ秒ずつ増加します。たとえば、20 を指定した場合、パーセル送信間に 1 秒の遅延が発生します。0 を指定した場合、パーセル送信間に遅延の強制は発生しません。

デフォルト: 0

例: 1 秒半のスロットル係数を指定するには、以下のように入力します。

```
dtsccli -a "ipath=temo33::c:¥jpegs¥cows.jpeg" protocol=spx
            throttle=10 "rpath=kyto202::c:¥jpegs¥cows.jpeg"
```

`transfer_mode=transfer_mode`

(オプション) 転送モードを指定します。転送モードが **bcast** または **mcast** の場合、UDP ブロードキャストまたはマルチキャスト転送です。有効な値は以下のとおりです。

p2p

ポイントツーポイントモードを指定します。

PPP

PPP (Point-to-Point Protocol) モードを指定します。一般電話回線、ISDN、その他の高速回線上でのポイントツーポイント転送にのみ有効です。

fanout

ファンアウトモードを指定します。

bcast

ブロードキャストモードを指定します。IPv4 ブロードキャスト転送にのみ有効です。この転送モードを使用する場合は、`broadcast_address=broadcast_address` および `protocol=bcast` も指定します。

mcast

マルチキャストモードを指定します。IPv4 および IPv6 マルチキャスト転送に有効です。この転送モードを使用する場合は、`multicast_address=multicast_address` および `protocol=mcast` も指定します。

デフォルト : p2p

例 : ファンアウトを LondonHQ という名前のコンピュータから Local100 という名前のサテライト コンピュータに Local110 を経由して送信します。

```
dtscli -a "ipath=LondonHQ::c:¥mandates¥memo5005.doc"
"rpath=Local100::c:¥mandates¥memo5005.doc"
"rpath=Local101::c:¥mandates¥memo5005.doc"
"rpath=Local102::c:¥mandates¥memo5005.doc"
.
.
.
"rpath=Local110::c:¥mandates¥memo5005.doc"
transfer_mode=fanout
```

関連項目：

[dtcli コマンド - フィルタの適用](#) (P. 88)

[エージェントの DiscreetMode および転送優先順位の設定](#) (P. 12)

第 3 章 : 管理対象転送 (TOS)

データ転送は、転送オブジェクト サーバ (TOS) で `-transfer` (または `-t`) パラメータを使用して作成および管理できます。

例 :

この例は、2 台の Windows コンピュータでの簡単な管理対象転送です。`-transfer` パラメータを使用する必要があります。

```
dtscli -tos host=hostname user=username::password
-t "ipath=wstabp3j::C:¥My Documents¥Monthly Reports¥apr2008.doc"
  "rpath=wstabp3n::C:¥Monthly Reports¥David¥apr2008.doc"
```

例 :

複数の管理対象転送を実行するには、`-transfer` パラメータを複数指定します。この例では、同じコンピュータから同じ受信者に 2 つのファイルを送信します。また、同じファイルを別々の受信者に複数回送信したり、別々のコンピュータから別々のファイルを送信できます。

```
dtscli -tos host=hostname user=username::password
-t "ipath=jones201::c:¥mandates¥memo5005.doc"
  "rpath=smith110::c:¥mandates¥memo5005.doc"
-t "ipath=jones201::c:¥mandates¥memo5010.doc"
  "rpath=smith110::c:¥mandates¥memo5010.doc"
```

このセクションには、以下のトピックが含まれています。

[dtscli コマンド -- 転送の作成](#) (P. 80)

[dtscli コマンド - 複数の出力ファイルの送信 \(管理対象\)](#) (P. 83)

[dtscli コマンド -- 転送ジョブの管理](#) (P. 84)

[dtscli コマンド - スキップ ロジックの使用](#) (P. 86)

[dtscli コマンド - フィルタの適用](#) (P. 88)

[dtscli コマンド -- 転送エイリアスの使用](#) (P. 96)

[dtscli コマンド -- 転送ステータスの表示](#) (P. 97)

[HTTP プロトコルのパラメータ](#) (P. 97)

dtsccli コマンド -- 転送の作成

`dtsccli -transfer`（または `-t`）コマンドによって、転送が作成されます。

単純な転送には、以下の構文が使用されます。

```
dtsccli -transfer "ipath=InitiatorMachine::SourceFile"
                "rpath=ResponderMachine::DestinationFile"
                iuser=InitiatorUser::Password
                ruser=ResponderUser::Password
リリース 12.5.00
```

例：

この例では、最初に転送（ID 270）を作成し、転送ジョブ（ID 271）内に配置し、転送ジョブを起動します。

```
dtsccli -transfer "ipath=Gold::c:\%dtstemp%\file.src"
                "rpath=Silver::c:\%dtstemp%\file.dest"
                iuser=dts::dts
                ruser=dts::dts
```

サンプル出力メッセージ

```
転送 270 が作成されました。
転送ジョブ 271 が作成されました。
転送ジョブ 271 が起動されました。
転送ジョブ 271 のステータス： STARTING
転送ジョブ 271 のステータス： IN_PROGRESS
```

注：転送 ID およびジョブ ID は、実際の処理ではこの例とは異なる場合があります。

例：

dtscli -transfer のデフォルトの動作では、転送はただちに起動され、正常に完了すると削除されます。しかし、この動作を変更したい場合もあるかもしれません。

-mode パラメータ グループを使用すると、コントロールのモードを指定できます。具体的には、転送がただちに実行されるかどうか、および転送が実行される場合は、転送が完了したときに基礎となるオブジェクトが削除されるかどうかを指定できます。

この例では、起動された後も転送は保持されます。

```
dtscli -transfer "ipath=Gold::c:%dtstemp%file.src"
               "rpath=Silver::c:%dtstemp%file.dest"
               iuser=dts::dts
               ruser=dts::dts
               -mode keep
```

サンプル出力メッセージ

```
転送 292 が作成されました。
転送ジョブ 293 が作成されました。    <- 転送ジョブ ID
転送ジョブ 293 が起動されました。
転送ジョブ 293 のステータス： STARTING
転送ジョブ 293 のステータス： IN_PROGRESS
```

出力には、転送および転送ジョブの ID が示されます。

転送が作成され、起動されますが、削除されません。ADT Transfer Client がインストールされている場合は、作成された転送および転送ジョブを表示できます。

例：

この例では、新しい転送が作成されますが、起動されないか、転送がジョブに追加されません。

```
dtsccli -transfer "ipath=Gold::c:%dtstemp%file.src"
               "rpath=Silver::c:%dtstemp%file.dest"
               iuser=dts::dts
               ruser=dts::dts
               -mode defer
```

サンプル出力メッセージ

転送 291 が作成されました。

出力には、作成された転送の ID が示されます。

転送が作成されるものの、起動は延期されます。この場合も、転送クライアントを使用して、`[unassigned transfers]` で作成された転送を表示します。dtsccli を使用して転送を表示することもできます。

```
dtsccli -transfer method=status id=all
```

関連項目：

[dtsccli -transfer パラメータ グループ - 管理対象転送を実行](#) (P. 22)

dtscli コマンド - 複数の出力ファイルの送信 (管理対象)

`dtscli -transfer` (または `-t`) コマンドによって、管理対象転送を使用して複数の出力ファイルを送信できます。入力および出力ファイルごとに `-transfer` オペランドをスペースで区切って入力します。1 つの `dtscli` コマンドを使用して、最大 16 の転送を指定できます

例 :

この例では、出力ファイルを 1 つのみ指定しています。

```
dtscli -t "ipath=steve::c:¥misc¥secrets.doc"
        "rpath=katie::c:¥misc¥secrets.doc"
```

重要 : 複数の管理対象転送を作成する場合は、作成する転送の数にかかわらず、`-tos`、`-sos`、`-job`、`-schedule` の各オペランドを 1 回のみ指定します。ただし、`-transfer` オペランドおよびそのすべてのパラメータ (`ipath=`、`rpath=`、`direction=` など) は、転送 (`-transfer` 式) ごとに 1 回指定します。

例 : CPI-C を除くすべてのサポートされるプロトコルの管理対象転送

新しい転送を作成した後、`dtscli` による前の転送からの値の引継ぎは行われません。以下の例では、3 つの転送が指定されています。最初の転送では区切り文字の変換 (`delimiter_conversion=y`) を使用していますが、ほかの 2 つの転送ではこのパラメータを指定していないため、デフォルト

(`delimiter_conversion=n`) が使用されます。この例は Windows で有効で、同じ入力ファイルが 3 つの異なるコンピュータに送信されます。リモート転送オブジェクト サーバはすべての転送に使用されます。

```
dtscli -tos host=wsiwig
        -t "ipath=steve::c:¥misc¥secrets.doc" "rpath=margaret::c:¥misc¥secrets.doc"
        delimiter_conversion=y
        -t "ipath=steve::c:¥misc¥secrets.doc" "rpath=katie::c:¥misc¥secrets.doc"
        -t "ipath=steve::c:¥misc¥secrets.doc" "rpath=roshni::c:¥misc¥secrets.doc" ...
```

上記の例では、(同じパス名の) 同じ入力ファイルが各受信コンピュータに送信されます。必要に応じて、2 つ以上の異なる入力ファイルの送信、出力ファイルのパス名の変更、またはその両方を行うことができます。

この例では、3つの異なる入力ファイルが送信され、出力ファイルごとに新しいファイル名が指定されています。

```
dtsccli -t "ipath=steve::c:¥receipts¥monday.doc"
"rpath=margaret::c:¥monday¥mon_fs.doc"
-t "ipath=steve::c:¥receipts¥tuesday.doc"
"rpath=margaret::c:¥tuesday¥tues_fs.doc"
-t "ipath=steve::c:¥receipts¥wednesday.doc"
"rpath=margaret::c:¥wednesday¥wed_fs.doc" ...
```

注：転送において任意のコンピュータのセキュリティモードが「fail」に設定されている場合は、すべての受信コンピュータについて適切なユーザー名およびパスワードが指定されていることを確認します。または、有効なユーザーIDおよびパスワードを指定していない受信コンピュータで、セキュリティモードがQUIETまたはWARNになっていることを確認します。これ以外の場合、受信側コンピュータのDTSエージェントのセキュリティ機能によって、そのコンピュータへの転送が失敗します。

関連項目：

[管理対象転送](#) (P. 9)

dtsccli コマンド -- 転送ジョブの管理

dtsccli -job（または-j）コマンドでは転送ジョブを管理します。転送ジョブに適用できるさまざまなメソッドがあります。使用可能なメソッドのリストを表示するには、以下のいずれかのヘルプコマンドを使用します。②

```
dtsccli -help job
dtsccli -help job method
```

例：転送ジョブの起動

この例では、以前作成および保持した転送ジョブを再起動します。

```
dtsccli -job method=activate id=transfer_job_id
```

例：転送ジョブのリセット

この例では、`reset` メソッドを使用して転送ジョブをリセットします。

```
dtscli -job method=reset id=transfer_job_id
```

例：転送ジョブの中止

```
dtscli -job method=abort id=transfer_job_id
```

または、`Ctrl + C` キーを押して現在の操作を中止できます。

サンプル出力メッセージ

```
転送ジョブ 293 が起動されました。  
転送ジョブ 293 のステータス： STARTING  
転送ジョブ 293 のステータス： IN_PROGRESS  
転送ジョブ 293 のステータス： ABORTED
```

例：新しい転送ジョブの作成

```
dtscli -job method=create
```

サンプル出力メッセージ

```
転送ジョブ 305 が作成されました。
```

ジョブ転送 ID をメモする代わりに、名前のエイリアスまたはラベルを使用できます。このジョブのラベルを設定することにより、ジョブをラベルで参照できます。

例：転送ジョブへの転送の追加

この例では、作成されたが起動されていない転送ジョブを転送に追加します。前の例での転送 ID が必要です。

```
dtscli -job id=transfer_job_id method=add method_parameters=your_transfer_id
```

関連項目：

[dtscli -job パラメータ グループ - 転送ジョブの管理](#) (P. 40)

dtscli コマンド - スキップ ロジックの使用

その他のどの転送をアクティブにすべきかスキップすべきか 1 つ以上の特定の転送の成功または失敗が決める場合、スキップ ロジックは最も役立ちます。1 つまたは両方のスキップ ロジック設定 (CSKIP= および FSKIP=) を転送に対して指定することができます。

以下の表に、代表的なスキップ ロジック設定およびその結果を示します。

値	転送が完了した場合	転送が失敗した場合
cskip=1 fskip=2	次の転送はスキップされます。	次の 2 つの転送は、スキップされます。
*fskip=0	次の転送は起動されます。	次の転送は起動されます。
*cskip=-1	転送はこれ以上起動されません。	転送はこれ以上起動されません。
fskip=l450	次の転送は起動されます。	次から転送 450 までの転送はスキップされます。転送 450 は起動されます。
fskip= LtransferOne	次の転送は起動されます。	次からラベル「TransferOne」の転送までの転送はスキップされます。この転送は起動されます。
fskip= NtransferTwo	次の転送は起動されます。	次の転送から、「TransferTwo」という名前の転送までは転送がスキップされます。その後この転送は起動されます。

* これらの場合、0 または -1 という値によって、実行されるスキップ ロジックがないことが示されます。

関連項目：

[dtscli -transfer パラメータ グループ - 管理対象転送を実行](#) (P. 22)

[dtscli -job パラメータ グループ - 転送ジョブの管理](#) (P. 40)

連続する転送ジョブでのスキップ ロジックの使用

次に、例を挙げます。転送 A が成功した場合、転送 B - D はスキップされ、転送 E - G が実行されますが、転送 A が失敗した場合、転送 B - D が実行され、転送 E - G はスキップされます。転送 A 以外の転送が失敗した場合は、ジョブ全体を中止します。転送 A 以外の転送が失敗した場合は、ジョブ全体を中止します。

以下の手順に従います。

1. 順次転送ジョブを作成し、転送 A - G をシーケンス順にジョブに追加します。

2. 転送 A

```
dtscli "ipath=ipath" "rpath=rpath" ... -t cskip=3 fskip=0
```

3. 転送 B

```
dtscli "ipath=ipath" "rpath=rpath" ... -t cskip=0 fskip=-1
```

4. 転送 C

```
dtscli "ipath=ipath" "rpath=rpath" ... -t cskip=0 fskip=-1
```

転送 B および C が成功した場合は、次の転送が起動されますが、どちらかの転送が失敗した場合は、この例の元の要件に従って、ジョブ全体が中止されます。

5. 転送 D

```
dtscli "ipath=ipath" "rpath=rpath" ... -t cskip=-1 fskip=-1
```

転送 D では、`cskip` と `fskip` の値の両方に `-1` を指定します。その結果、転送 B - D が実行された場合、転送 D が成功または失敗した後には、転送は実行されません。

6. 転送 E

```
dtscli "ipath=ipath" "rpath=rpath" ... -t cskip=0 fskip=-1
```

7. 転送 F

```
dtscli "ipath=ipath" "rpath=rpath" ... -t cskip=0 fskip=-1
```

転送 E および F が成功した場合は、次の転送が起動されますが、どちらかの転送が失敗した場合は、ジョブ全体が中止されます。

注：転送 G はジョブの最後の転送なので、スキップ ロジック設定は必要ありません。

dtsccli コマンド - フィルタの適用

dtsccli コマンドを使用して、フィルタした転送を送信できます。DTS エージェントでは、ファイル転送時にフィルタを適用できます。暗号化と復号化、圧縮と解凍などの方法で、フィルタによってデータが変換されます。

フィルタの基本的な種類

ファイル フィルタ (`f_filters`)

一度に 1 ファイルずつ転送データが処理されます。ファイル フィルタはファイルが転送される前に、ファイル全体に対して適用されます。

パーセル フィルタ (`p_filters`)

一度に 1 パーセルずつ転送データが処理されます（パーセルは、ファイル内のより小さなデータ単位です）。パーセル フィルタは、ファイルの転送中に適用されます。

フィルタは通常、組み合わされて機能します。読み取りフィルタによって実行されたアクションの逆のアクションが、書き込みフィルタによって実行される必要があります。フィルタは以下の順序で適用されます。

1. ファイル読み取りフィルタ
2. パーセル読み取りフィルタ
3. パーセル書き込みフィルタ
4. ファイル書き込みフィルタ

同じタイプの読み取りフィルタ（ファイルまたはパーセル）は、コマンドラインで指定する場合と同じ順序で適用されます。同じタイプの書き込みフィルタは、コマンドラインで指定する場合と反対の順序で適用されます。

通常、フィルタ パラメータを指定するのはエージェント間転送のみです。管理対象転送では指定しません。ただし、暗号化フィルタだけは例外です。たとえば、**Data Transport Service** では、**PLAIN** ファイルおよびパーセル暗号化フィルタが提供されます。そのどちらでも **ENDEC_KEY** パラメータが必要です。管理対象転送とエージェント間転送の両方について、**PLAIN** 暗号化フィルタを使用する場合は、**ENDEC_KEY** を指定します。パラメータが必要なほかのすべてのフィルタの場合は、管理対象転送によって転送オブジェクト サーバからの値が自動的に取得されます。

関連項目：

[エージェント間転送](#) (P. 99)

[dtscli -transfer パラメータ グループ - 管理対象転送を実行](#) (P. 22)

[dtscli -agent パラメータ グループ - エージェント間転送の実行](#) (P. 58)

[定義済みフィルタ](#) (P. 93)

dtscli コマンド -- パーセル フィルタの追加

有効な DTSLI パーセル フィルタには、次のタイプがあります。

- バイナリ（デフォルト）またはテキスト
- 圧縮
- 暗号化

2 つの主要なパーセル フィルタは、バイナリまたはテキストです。タイプの異なる 2 台のコンピュータ間（Windows と UNIX など）でテキスト ファイルを転送する場合は、テキスト フィルタを使用します。これにより、テキスト形式が保持されます。それ以外の場合は、常にバイナリ フィルタを使用します。

注：バイナリ フィルタとテキスト ファイルは相互排他的です。転送は、テキストかバイナリのいずれかであり、両方はありません。

フィルタをエージェント転送に追加するためのヘルプは、次のコマンドによって表示されます。

```
dtcli -help agent p_filters
```

例：

- 以下の例では、ファイルは **Gold** から **Silver** に、テキスト フィルタおよび短い形式を使用して転送されます。

```
dtcli -agent "ipath=Gold::c:%dtstemp%file.src"
            "rpath=Silver::c:%dtstemp%file.dest"
            "p_filters=text"
```

- 以下の例では同じ操作を実行しますが、長い形式を使用しています。

```
dtcli -agent "ipath=Gold::c:%dtstemp%file.src"
            "rpath=Silver::c:%dtstemp%file.dest"
            "p_filters=TEXT_READ:TEXT_WRITE"
```

注：テキスト ファイルの転送にはテキスト フィルタのみを使用します。

例：

- 以下の例では、ファイルは **Gold** から **Silver** に、圧縮フィルタおよび短い形式を使用して転送されます。

```
dtcli -agent "ipath=Gold::c:%dtstemp%file.src"
            "rpath=Silver::c:%dtstemp%file.dest"
            "p_filters=text compress"
```

- 以下の例では同じ操作を実行しますが、長い形式を使用しています。

```
dtcli -agent "ipath=Gold::c:%dtstemp%file.src"
            "rpath=Silver::c:%dtstemp%file.dest"
            "p_filters=TEXT_READ:TEXT_WRITE HUFFMAN_COMPRESS:HUFFMAN_UNCOMPRESS"
```

例：

- 以下の例では、複数の対のパーセル フィルタを短い形式を使用して追加します。

```
dtscli -agent "ipath=Gold::c:%dtstemp%file.src"
             "rpath=Silver::c:%dtstemp%file.dest"
             "p_filters=binary compress"
```

- 以下の例では同じ操作を実行しますが、長い形式を使用しています。

```
dtscli -agent "ipath=Gold::c:%dtstemp%file.src"
             "rpath=Silver::c:%dtstemp%file.dest"
             "p_filters=BINARY_READ:BINARY_WRITE
             HUFFMAN_COMPRESS:HUFFMAN_UNCOMPRESS"
```

フィルタをリストする順序は重要です。バイナリ フィルタは、圧縮フィルタよりも前にリストする必要があります。たとえば、フィルタを入れ替えて、最初に圧縮フィルタ、次にバイナリ フィルタを呼び出してみると、転送は失敗します。

注： バイナリ フィルタまたはテキスト フィルタをリストの最初に指定してください。

dtscli コマンド -- ファイル フィルタの追加

有効な DTSLI ファイル フィルタのタイプ

- ディレクトリ フィルタ
- 暗号化フィルタ
- ファイル属性フィルタ

ファイル フィルタを追加する場合、パーセル フィルタも必要な場合があります。ディレクトリを転送するには、ファイル フィルタを **dtscli** コマンドに追加する必要があります。ディレクトリ フィルタを追加せずにディレクトリを転送しようとする、以下のエラー メッセージが表示されます。

「入力データ <c:%dtstemp> error=<Permission denied> のオープンが失敗しました。」

ファイル フィルタのヘルプは、次のコマンドによって表示されます。

```
dtscli -help agent f_filters
```

例：ディレクトリの転送

- ディレクトリ フィルタの短い形式を使用してディレクトリを転送します。

```
dtsccli -agent ipath=Gold::c:%dtstemp
        "rpath=Silver::c:%dtstemp"
        "f_filters=dir"
```

- 長い形式を使用して、同じ操作を実行します。

```
dtsccli -agent "ipath=Gold::c:%dtstemp"
        "rpath=Silver::c:%dtstemp"
        "f_filters=DIRTREE_READ:DIRTREE_WRITE"
        "p_filters=binary"
```

注：長い形式を使用する場合は、パーセル フィルタも適用します。

例：ファイルの暗号化

応答者に送信する前にファイルを暗号化します。

```
dtsccli -agent "ipath=Gold::c:%dtstemp"
        "rpath=Silver::c:%dtstemp"
        "f_filters=PLAIN_ENCRYPT_FILE(ENDEC_KEY=key):PLAIN_DECRYPT_FILE()"
        "p_filters=binary"
```

例：転送されるファイルへの元のファイルの属性の適用

ファイル属性フィルタの使用によって、転送されるファイルは元のファイルと同じ属性を持ちます。

- `file.src` に特定の属性（読み取り専用、非表示など）を短い形式を使用して付与します。

```
dtsccli -agent "ipath=Gold::c:%dtstemp%file.src"
        "rpath=Silver::c:%dtstemp%file.dest"
        "f_filters=fattr"
        "p_filters=binary"
```

- 長い形式を使用して、同じ操作を実行します。

```
dtsccli -agent "ipath=Gold::c:%dtstemp%file.src"
        "rpath=Silver::c:%dtstemp%file.dest"
        "f_filters=FILE_ATTRIBUTE:"
        "p_filters=binary"
```

定義済みフィルタ

あらかじめ定義されている DTS ファイル フィルタおよびパーセル フィルタの一覧

フィルタ名	Description	タイプ
BINARY_READ	バイナリの読み取り	read_parcel
BINARY_WRITE	バイナリの書き込み	write_parcel
TEXT_READ	テキストの読み取り	read_parcel
TEXT_WRITE	テキストの書き込み	write_parcel
STAGING	開始側のステージング ファイルの使用	read_file
STAGING	受信側のステージング ファイルの使用	write_file
HUFFMAN_COMPRESS	ハフマン圧縮	read_parcel
RLE_COMPRESS	RLE 圧縮	read_parcel
ZLIB_COMPRESS	ZLIB 圧縮	read_parcel
LZRW_COMPRESS	LZRW 圧縮	read_parcel
HUFFMAN_UNCOMPRESS	ハフマン解凍	write_parcel
RLE_UNCOMPRESS	RLE 解凍	write_parcel
ZLIB_UNCOMPRESS	ZLIB 解凍	write_parcel
LZRW_UNCOMPRESS	LZRW 解凍	write_parcel
PLAIN_ENCRYPT	DTS プレーン暗号化	read_parcel
PLAIN_DECRYPT	DTS プレーン復号化	write_parcel
PLAIN_ENCRYPT_FILE	DTS プレーン暗号化ファイル フィルタ	read_file
PLAIN_DECRYPT_FILE	DTS プレーン復号化ファイル フィルタ	write_file
VIRUS_SCAN	ウイルスの検出	read_file
VIRUS_SCAN	ウイルスの検出	write_file
CREATE_PATH	入力/フィルタのパラメータに基づくパスの作成	read_file
CREATE_PATH	出力/フィルタのパラメータに基づくパスの作成	write_file

フィルタ名	Description	タイプ
DIRTREE_READ	ディレクトリ ツリーのコンテンツを含む アーカイブの作成	read_file
DIRTREE_WRITE	アーカイブのコンテンツに基づくディレク トリ ツリーの作成	write_file
USD_RESOLVE_PATH	将来の使用のために予約	write_file
FILE_ATTRIBUTE	ファイル属性フィルタ	read_file
BINARY_VARREC_READ	MVS 変数レコードの読み取り	read_parcel
BINARY_VARREC_WRITE	MVS 変数レコードの書き込み	write_parcel
PDS_UNLOAD	MVS PDS アンロード	read_file
PDS_UNLOADED_READ	MVS PDS アンロードされたファイルの読み 取り	read_parcel
PDS_UNLOADED_WRITE	MVS PDS アンロードされたファイルの書き 込み	write_parcel
PDS_LOAD	MVS PDS ロード	write_file
VSAM_UNLOAD	MVS VSAM アンロード	read_file
VSAM_UNLOADED_READ	MVS VSAM アンロードされたファイルの読 み取り	read_parcel
VSAM_UNLOADED_WRITE	MVS VSAM アンロードされたファイルの書 き込み	write_parcel
VSAM_LOAD	MVS VSAM ロード	write_file
SAVE_FILE_READ	AS/400 保存ファイルの読み取り	read_file
SAVE_FILE_WRITE	AS/400 保存ファイルの書き込み	write_file
DM_SCRIPT_READ	読み取りステージの Desktop Management スクリプトの実行	read_file
DM_SCRIPT_WRITE	書き込みステージの Desktop Management スクリプトの実行	write_file
AES256_ENCRYPT	CA-DSM AES-256 暗号化	read_parcel
AES256_DECRYPT	CA-DSM AES-256 復号化	write_parcel
AES256_ENCRYPT_FILE	CA-DSM AES-256 暗号化ファイルフィルタ	read_file
AES256_DECRYPT_FILE	CA-DSM AES-256 復号化ファイルフィルタ	write_file

フィルタ名	Description	タイプ
AES192_ENCRYPT	CA-DSM AES-192 暗号化	read_parcel
AES192_DECRYPT	CA-DSM AES-192 復号化	write_parcel
AES192_ENCRYPT_FILE	CA-DSM AES-192 暗号化ファイル フィルタ	read_file
AES192_DECRYPT_FILE	CA-DSM AES-192 復号化ファイル フィルタ	write_file
AES128_ENCRYPT	CA-DSM AES-128 暗号化	read_parcel
AES128_DECRYPT	CA-DSM AES-128 復号化	write_parcel
AES128_ENCRYPT_FILE	CA-DSM AES-128 暗号化ファイル フィルタ	read_file
AES128_DECRYPT_FILE	CA-DSM AES-128 復号化ファイル フィルタ	write_file
3DES_ENCRYPT	CA-DSM 3DES 暗号化	read_parcel
3DES_DECRYPT	CA-DSM 3DES 復号化	write_parcel
3DES_ENCRYPT_FILE	CA-DSM 3DES 暗号化ファイル フィルタ	read_file
3DES_DECRYPT_FILE	CA-DSM 3DES 復号化ファイル フィルタ	write_file
EXT_COMPRESS	外部圧縮	read_file
EXT_UNCOMPRESS	外部解凍	write_file

関連項目：

[dtscli コマンド - フィルタの適用](#) (P. 88)

dtcli コマンド -- 転送エイリアスの使用

転送 ID をすべてトラッキングすることは容易ではありません。DTSLI では名前のエイリアスがサポートされています。番号ではなく転送または転送ジョブ ラベルを使用して参照できます。

例：エイリアスを使用してジョブを作成および起動

転送および転送ジョブの両方にラベルを使用します。

次の手順に従ってください：

1. エイリアスを使用して転送を作成し、起動を延期します。

```
dtcli -transfer "ipath=Gold::c:%dtstemp%file.src"
            "rpath=Silver::c:%dtstemp%file.dest"
            iuser=dts::dts
            ruser=dts::dts
            label=WATER
            -mode defer
```

転送が作成されました。ラベルを確認してください。

2. エイリアスを使用して転送ジョブを作成します。

```
dtcli -job method=create label=FIRE
```

転送ジョブが作成されました。ラベルを確認してください。

3. 転送を転送ジョブに追加します。

```
dtcli -job id=FIRE method=add method_parameters=WATER
```

4. 転送を起動します。

```
dtcli -job method=activate id=FIRE
```

ADT Transfer Client がインストールされている場合は、これを使用して、作成および起動した新しい転送および転送ジョブを表示できます。ただし、ラベルは表示できません。

dtscli コマンド -- 転送ステータスの表示

`status` メソッドを使用して転送のステータスを取得および表示できます。

例：

ラベルが「**WATER**」の転送のステータスを表示します。

```
dtscli -transfer method=status id=WATER
```

例：

特別なキーワード「`id=all`」を使用して、すべての転送のステータスを表示します。

```
dtscli -transfer method=status id=all
```

HTTP プロトコルのパラメータ

Data Transport Service r2 および r3 では、HyperText Transfer Protocol (HTTP) プロトコルがサポートされています。

サーバに DTS エージェントがなくても、転送を作成、管理、実行して HTTP サーバまたはそのプロキシサーバからデータを受信できます。このようにして、DTS によってインターネットからのファイルのダウンロードがサポートされます。DTS を使用すると、HTTP サーバまたはそのプロキシサーバからデータを受信できますが、データをサーバに送信することはできません。

HTTP エージェント間転送を実行する構文は、以下のとおりです。すべてのオペランドが必要です。

```
dtscli -a|t
  protocol=http "ipath=DTA_machine_name::destination_path"
  "rpath=http_host_name::http_source_file_path_name"
  iuser=username::password
  direction=r
```

-a|t

転送のタイプを示します。エージェント間 (-a) または管理対象 (-t) です。

protocol=http

インターネット ダウンロードのプロトコルとして HTTP を指定します。

"ipath=DTA_machine_name::destination_path"

HTTP ダウンロードを受信する DTS エージェント コンピュータの名前を指定します。後ろにコロンを 2 つ付け、このコンピュータに新しくダウンロードされるファイルのパス名を続けます。

"rpath=http_host_name::http_source_file_path_name"

HTTP サーバ (またはそのプロキシサーバ) のホスト名を指定します。後ろにコロンを 2 つ付け、このサーバからダウンロードするファイルのパス名を続けます。

iuser=username::password

開始コンピュータ、すなわち、転送を開始する DTS エージェント コンピュータのユーザ名およびパスワードを指定します。

direction=r

開始コンピュータが転送を受信することを指定します。

第 4 章：エージェント間転送

パラメータ `-agent`（または `-a`）によって、転送がエージェント間転送であることが指定されます。エージェント間転送は、転送オブジェクトサーバ（TOS）が関与しない転送です。

例：簡単なエージェント間転送の実行

以下の例は、2 台の UNIX または Linux コンピュータでの簡単なエージェント間転送です。 `-agent`（`-a`）パラメータを使用する必要があります。

```
dtsccli -a iuser=username::password ruser=username::password
"ipath=LondonHQ::/home/mandates/memo5005.txt"
"rpath=Local100::/home/mandates/memo5005.txt"
```

例：1 台のコンピュータから複数の受領者に 1 つのファイルを送信

エージェント間転送を使用すると、1 台のコンピュータから複数の受領者に 1 つのファイルを送信できます。

```
dtsccli -a "ipath=jones201::c:¥mandates¥memo5005.doc"
"rpath=smith110::c:¥mandates¥memo5005.doc"
"rpath=patel335::c:¥mandates¥memo5005.doc"
```

このセクションには、以下のトピックが含まれています。

[dtsccli コマンド -- 簡単なポイント ツー ポイント転送の作成](#) (P. 100)

[dtsccli コマンド -- 複数のポイント ツー ポイント転送の作成](#) (P. 100)

[dtsccli コマンド - ホップ転送の設定](#) (P. 102)

[dtsccli コマンド - 複数の出力ファイルの送信（エージェント間）](#) (P. 104)

[dtsccli コマンド -- 方向の変更](#) (P. 105)

[TCP プロトコルのパラメータ](#) (P. 106)

[HTTP プロトコルのパラメータ](#) (P. 106)

[PPP（Point-to-Point Protocol）パラメータ](#) (P. 108)

[UDP プロトコルのパラメータ](#) (P. 110)

[CPI-C プロトコルのパラメータ](#) (P. 111)

[SPX プロトコルのパラメータ](#) (P. 112)

dtsccli コマンド -- 簡単なポイント ツー ポイント転送の作成

dtsccli -agent (または **-a**) コマンドを使用すると、簡単なポイント ツー ポイント転送を作成できます。

簡単なポイント ツー ポイント転送の構文

```
dtsccli -agent "ipath=InitiatorMachine::SourceFile"
              "rpath=ResponderMachine::DestinationFile"
              iuser=InitiatorUser::Password
              ruser=ResponderUser::Password
```

例：ファイルの転送

コンピュータ **Gold** からファイル「**c:¥dtstemp¥file.src**」をコンピュータ **Silver** の「**c:¥dtstemp¥file.dest**」に転送します。

```
dtsccli -agent "ipath=Gold::c:¥dtstemp¥file.src"
              "rpath=Silver::c:¥dtstemp¥file.dest"
              iuser=dts::dts
              ruser=dts::dts
```

dtsccli コマンド -- 複数のポイント ツー ポイント転送の作成

dtsccli -agent (または **-a**) コマンドを使用すると、複数のポイント ツー ポイント転送を作成できます。**rpath** および **transfer_mode** パラメータを使用して、同じ入力ファイルを複数の出力場所に転送できます。

transfer_mode パラメータの有効な値は、以下のとおりです。

p2p

ポイント ツー ポイント転送を指定します。

PPP

PPP (Point-to-Point Protocol) モードを指定します。一般電話回線、ISDN、そのほかの高速回線上での ポイント ツー ポイント転送にのみ有効です。

注：DTS は、最新の PPP 通信プロトコルである PPPv6 をサポートしません。

f (または fanout)

ファンアウト転送を指定します。

bcast

IPv4 ブロードキャスト ポイント転送を指定します。

mcast

IPv4 または IPv6 マルチキャスト転送を指定します。

デフォルト : p2p

例 : ポイント ツー ポイント モードでの転送

この例では、2 つのポイント ツー ポイント転送を実行します。最初に、Gold および Silver 間の転送が実行され、この転送が終了した後に、Gold および Bronze 間の転送が実行されます。

```
dtscli -agent "ipath=Gold::c:%dtstemp%file.src"
            "rpath=Silver::c:%dtstemp%file.dest"
            "rpath=Bronze::c:%dtstemp%file.dest"
```

例 : ファンアウト モードでの転送

ファンアウト転送とは、直接ポイント ツー ポイント通信プロトコルを使用して、データが複数の応答コンピュータに送信される転送を意味します。データは 1 回読み取られ、送信はターゲットの応答側ごとに 1 回ずつ、合わせて複数回行われます。

前述のポイント ツー ポイント転送の例の転送メソッドを fanout モードに設定します。

```
dtscli -agent "ipath=Gold::c:%dtstemp%file.src"
            "rpath=Silver::c:%dtstemp%file.dest"
            "rpath=Bronze::c:%dtstemp%file.dest"
            transfer_mode=fanout
```

例：ブロードキャストとしての転送

この例では、転送メソッドをブロードキャスト（bcast）に設定します。ブロードキャストにはブロードキャストアドレスが必要です。

```
dtsccli -agent "ipath=Gold::c:%dtstemp%file.src"
            "rpath=Silver::c:%dtstemp%file.dest"
            "rpath=Bronze::c:%dtstemp%file.dest"
            transfer_mode=bcast
            broadcast_address=your_broadcast_address
```

例：マルチキャストとしての転送

この例では、転送メソッドをマルチキャスト（mcast）に設定します。マルチキャストは、マルチキャストアドレスが必要という点でブロードキャストに似ています。

```
dtsccli -agent "ipath=Gold::c:%dtstemp%file.src"
            "rpath=Silver::c:%dtstemp%file.dest"
            "rpath=Bronze::c:%dtstemp%file.dest"
            transfer_mode=bcast
            multicast_address=your_multicast_address
```

注：ブロードキャストアドレスおよびマルチキャストアドレスを計算する手順については、「*Data Transport Service 管理ガイド*」を参照してください。

dtsccli コマンド - ホップ転送の設定

dtsccli -agent（または -a）コマンドを使用すると、ホップ転送を作成できます。開始側コンピュータから応答側コンピュータへの直接のルートがない場合には、これは必要でないことがあります。ホップ転送では、第 3 のコンピュータを使用して転送を実行できます。

ホップ転送の構文

```
dtsccli -agent "ipath=InitiatorMachine::SourceFile"
            "rpath=ResponderMachine::DestinationFile"
            iuser=InitiatorUser::InitiatorPassword
            ruser=ResponderUser::ResponderPassword
            hop_hosts=HopMachine
```

例：ホップコンピュータを使用したファイルの転送

この例では、コンピュータ **Bronze** を使用してポイント ツー ポイント転送を実行します。ファイルは、最初に **Gold** から **Bronze** に転送され、次に **Bronze** から **Silver** に転送されます。

```
dtscli -agent "ipath=Gold::c:%dtstemp%file.src"
            "rpath=Silver::c:%dtstemp%file.dest"
            iuser=dts::dts
            ruser=dts::dts
            hop_hosts=bronze
```

例：複数のホップコンピュータを使用したファイルの転送

いくつかのホップ コンピュータを使用したホップが可能です。この操作を実行するには、ホップ コンピュータをスペースで区切ってリストにし、**hop_hosts** パラメータに追加します。この例では、転送は **Gold** から **Bronze** へ、次に **Bronze** から **Tin** へ、その次に **Tin** から **Silver** へと進みます。

```
dtscli -agent "ipath=Gold::c:%dtstemp%file.src"
            "rpath=Silver::c:%dtstemp%file.dest"
            iuser=dts::dts
            ruser=dts::dts
            "hop_hosts=bronze tin"
```

注： **hop_hosts** パラメータは引用符で囲む必要があります。

dtsccli コマンド - 複数の出力ファイルの送信 (エージェント間)

dtsccli -agent (または **-a**) コマンドによって、エージェント間転送を使用して、複数の出力ファイルを送信できます。入力および出力ファイルごとに **-agent** オペランドをスペースで区切って入力します。1 回の転送に最大 32 の出力ファイルを指定できます。

例 : 同じ出力ファイルを複数のコンピュータに送信

エージェント間転送を作成し、同じ出力ファイルを複数のコンピュータに送信する場合、入力パスを 1 回、および出力パスを応答側の指定に必要な数だけ入力できます。

```
dtsccli -a iuser=username::password ruser=username::password
    "ipath=steve::c:¥misc¥secrets.doc"
    "rpath=margaret::c:¥misc¥secrets.doc"
    "rpath=katie::c:¥misc¥secrets.doc"
    "rpath=roshni::c:¥misc¥secrets.doc"
```

転送において任意のコンピュータのセキュリティ モードが「fail」に設定されている場合は、特に注意して必要なユーザ名およびパスワードを指定します。指定できるユーザ ID およびパスワードは 1 つのみのため、転送を送信する前に、指定するユーザ ID とパスワードがすべての受信コンピュータについて有効であることを確認します。または、指定したユーザ ID とパスワードの組み合わせが有効ではない受信コンピュータで、セキュリティ モードが QUIET または WARN であることを確認します。これ以外の場合、受信側コンピュータの DTS エージェントのセキュリティ機能によって、そのコンピュータへの転送が失敗します。

例 : 複数の入力ファイルの送信

エージェント間転送を使用して複数の入力ファイルを送信する場合は、**dtsccli** コマンドを複数回使用します。

```
dtsccli -a "ipath=steve::c:¥misc¥secrets1.doc"
    "rpath=margaret::c:¥misc¥secrets1.doc"
dtsccli -a "ipath=steve::c:¥misc¥secrets2.doc"
    "rpath=margaret::c:¥misc¥secrets2.doc"
dtsccli -a "ipath=steve::c:¥misc¥secrets3.doc"
    "rpath=margaret::c:¥misc¥secrets3.doc"
```

関連項目：

[エージェント間転送](#) (P. 9)

dtscli コマンド -- 方向の変更

`direction` パラメータは転送の方向を決定します。このパラメータに有効な値は、以下のとおりです。

s

開始側コンピュータから応答側コンピュータにデータを送信します。

r

応答側コンピュータから開始側コンピュータにデータを送信します。

デフォルト：s

例：応答側から開始側へのファイルの転送

この例では、Silver（応答側）から file.src を Gold（開始側）の file.dest に転送するために、転送の方向を指定します。

```
dtscli -agent "ipath=Gold::c:%dtstemp%file.dest"
          "rpath=Silver::c:%dtstemp%file.src"
          direction=r
```

TCP プロトコルのパラメータ

エージェント間転送の `dtcli` コマンドに対して有効な TCP パラメータは、以下のとおりです。

`IPORT=nnnn`

DTS エージェントが TCP 接続をリスニングするレガシー ポート番号を指定します。

デフォルト : 8222

`NETADDR=nnn.nnn.nnn.nnn`

応答側のアドレスの代わりに使用する IP アドレスを示します。IPv6 アドレスの例は、`fe80::20d:56ff:fe18:5843` です。

HTTP プロトコルのパラメータ

Data Transport Service r2 および r3 では、HyperText Transfer Protocol (HTTP) プロトコルがサポートされています。

サーバに DTS エージェントがなくても、転送を作成、管理、実行して HTTP サーバまたはそのプロキシサーバからデータを受信できます。このようにして、DTS によってインターネットからのファイルのダウンロードがサポートされます。DTS を使用すると、HTTP サーバまたはそのプロキシサーバからデータを受信できますが、データをサーバに送信することはできません。

HTTP エージェント間転送を実行する構文は、以下のとおりです。すべてのオペランドが必要です。

```
dtcli -a|t
protocol=http "ipath=DTA_machine_name::destination_path"
"rpath=http_host_name::http_source_file_path_name"
iuser=username::password
direction=r
```

-a|t

転送のタイプを示します。エージェント間 (-a) または管理対象 (-t) です。

protocol=http

インターネット ダウンロードのプロトコルとして HTTP を指定します。

"ipath=DTA_machine_name::destination_path"

HTTP ダウンロードを受信する DTS エージェント コンピュータの名前を指定します。後ろにコロンを 2 つ付け、このコンピュータに新しくダウンロードされるファイルのパス名を続けます。

"rpath=http_host_name::http_source_file_path_name"

HTTP サーバ (またはそのプロキシサーバ) のホスト名を指定します。後ろにコロンを 2 つ付け、このサーバからダウンロードするファイルのパス名を続けます。

iuser=username::password

開始コンピュータ、すなわち、転送を開始する DTS エージェント コンピュータのユーザ名およびパスワードを指定します。

direction=r

開始コンピュータが転送を受信することを指定します。

PPP (Point-to-Point Protocol) パラメータ

PPP は、一般電話回線、ISDN、およびその他のほかの高速回線上で TCP/IP を使用するコンピュータ用の通信プロトコルです。

注： DTS は、最新の PPP 通信プロトコルである PPPv6 をサポートしません。

エージェント間転送の `dtsccli` コマンドに対して有効な PPP パラメータは、以下のとおりです。

Windows での PPP

`PPP_ENTRY=entry_name`

応答側への接続を確立するときに PPP プロトコルによって使用される、事前設定された電話帳エントリの名前を定義します。

`PPP_PHONENUMBER=phone_number`

応答側へダイヤルするのに必要な電話場号を指定します。

`PPP_USERNAME=login_name`

応答側との認証に使用するユーザ アカウントの名前を指定します。開始コンピュータが応答側へダイヤルするのに使用するユーザ アカウントの名前を入力します。このアカウントには、`dial-in` アクセス権が付与されている必要があります。

`PPP_PASSWORD=login_password`

応答側との認証に使用するユーザ アカウントのパスワードを指定します。`PPP_USERNAME` で指定したユーザ アカウントのパスワードを入力します。

Solaris Solstice での PPP

`PPP_ENTRY=entry_name`

`link.conf` で応答側の識別に使用する名前を指定します。

`PPP_USERNAME=login_name`

応答側のダイヤルインアカウントとの認証に使用するユーザ アカウントの名前を指定します。

PPP_PASSWORD=*login_password*

応答側のダイヤルインアカウントとの認証に使用するユーザアカウントのパスワードを指定します。

PPP_AUTHENTICATION=*PAP/CHAP/CLEARTEXT*

応答側との認証に使用する認証プロトコルを識別します。パスワード認証プロトコル (PAP) が最も一般的な選択です。

注： PAP は広く普及していますが、チャレンジハンドシェーク認証プロトコル (CHAP) よりも脆弱です。どの認証プロトコルを使用するかを決定するには、応答側の設定を参照します。 PPP 認証が必要ない場合は、値を CLEARTEXT に設定します。

PPP_DYNAMICIP=*YES/NO*

サーバが動的 IP アドレスをクライアントに割り当てるかどうかを示します。サーバが静的 IP アドレスをクライアントに割り当てる場合は、NO に設定します。次に、PPP_ASSIGNEDIP を使用して必要な IP アドレスを指定します。適切な値を決定するには、応答側の設定を参照します。

PPP_ASSIGNEDIP=*client_IP_address*

PPP_DYNAMICIP を NO に設定した場合に、応答側への接続のための IP アドレスを指定します。このパラメータは、リモートサーバが静的 IP アドレスをクライアントに割り当てる場合に使用します。

PPP_PHONENUMBER=*phone_number*

応答側の電話番号を指定します。この番号は、応答側へのアクセスを取得するために開始コンピュータによって使用されます。

例：Windows のポイント ツー ポイント転送

benjamin という名前の Windows コンピュータから reuben に PPP プロトコルを使用してポイント ツー ポイント転送を送信します。

```
dtsccli -a "ipath=benjamin::c:%nhmh.c" -protocol=ppp
"protocol_parameters=PPP_ENTRY=williamc
PPP_PHONENUMBER=16095551212
PPP_USERNAME=bill0627 PPP_PASSWORD=whisocks"
"rpath=reuben::c:%nhmh.c"
```

例：Solaris Solstice のポイント ツー ポイント転送

HQ501 という名前の Solaris Solstice コンピュータから Branch212 に PPP プロトコルを使用してポイント ツー ポイント転送を送信するには、以下のコマンドを入力します。

```
dtsccli -a "ipath=HQ501::c:¥nhmh.c" protocol=ppp
"protocol_parameters=PPP_ENTRY=williamc
PPP_USERNAME=bill062> PPP_PASSWORD=whisocks
PPP_DYNAMICIP=YES PPP_PHONENUMBER=19085551212"
"rpath=Branch212::c:¥nhmh.c"
```

この例では、応答側に動的 IP アドレスがあるため、**PPP_ASSIGNEDIP=client_IP_address** パラメータを指定する必要はありません。ただし、応答側が静的 IP アドレスの場合は **client_IP_address** パラメータは必須です。

UDP プロトコルのパラメータ

エージェント間転送の **dtsccli** コマンドに対して有効な UDP パラメータは、以下のとおりです。

IPOINT=nnnn

DTS エージェントが UDP 接続をリスニングするポート番号を指定します。

デフォルト：8223

NETADDR=nnn.nnn.nnn.nnn

応答側のアドレスの代わりに使用する IP アドレスを指定します。

例：UDP を使用するポイント ツー ポイント転送

1 台の Windows コンピュータから別のコンピュータに、UDP プロトコルおよびデフォルトとは異なるポート番号を使用してポイント ツー ポイント転送を送信するには、以下のコマンドを入力します。ほかのアプリケーションとの競合のため、応答側の DTS エージェントのデフォルトのリスニングポート番号が変更されている場合には、**IPOINT** を指定します。

```
dtsccli -a "ipath=ruth::c:¥ruth¥memo2.doc"
protocol=udp "protocol_parameters=IPOINT=3535"
"rpath=naomi::c:¥naomi¥memo2.doc" ...
```

CPI-C プロトコルのパラメータ

CPI-C プロトコルを使用することによってのみコンピュータがほかのコンピュータに接続できる場合、そのコンピュータは `dtsccli` コマンドを使用したデータ転送において応答側としてのみ機能します。ただし、**Data Transport Service** の機能を拡張するために設計された **Advantage Data Transport** などの **CA** ソリューションのユーザは、これらのコンピュータをデータ転送の開始側および応答側のマシンとして使用できます。このためには、`dtsccli` コマンドではなく **Data Transport** エクスプローラ GUI を使用して転送を実行します。

エージェント間転送の `dtsccli` コマンドに対して有効な **CPIC** パラメータは、以下のとおりです。

CPIC_LUNAME=*luname*

(必須) 応答側コンピュータの論理装置 (LU) 名。LU 名の先頭の文字には英字を指定する必要があります。

例：ポイントツーポイント転送

HQ501 という名前の Windows コンピュータから、Branch212 という名前の Windows コンピュータに CPI-C プロトコルを使用してポイントツーポイント転送を送信します。

```
dtsccli -a "ipath=HQ501::c:¥nhmh.c" protocol=cpic
"protocol_parameters=CPIC_LUNAME=D1234567"
"rpath=Branch212::c:¥nhmh.c"
```

CPI-C を使用する複数の出力ファイルの送信

出力ファイルごとにこれらの特殊要件に従うと、CPI-C を使用して複数の出力ファイルを送信できます。

次の手順に従ってください：

1. `"rpath=output_file"` オペランドで、応答側のコンピュータ名ではなく、応答側の **CPIC_LUNAME** を指定します。
`luname` を角かっこで囲む必要はありません。
2. プロトコルパラメータ オペランドで、**CPIC_LUNAME** を指定しないでください。

例：エージェント間転送

```
dtsccli -a "ipath=SWTZDAMY::DT0.DTS20.C(PWTCIP)"
"rpath=A44ISTD1::PUBLIC.AAKEHH.PWTCIP.TS094"
"rpath=A04ISTD1::PUBLIC.AAKEHH.PWTCIP.TS084"
"rpath=KWWIGP21::C:¥TDSTEST¥SNATEST¥FANOUT.C"
"rpath=KWWIGP51::/USR/AAKEHH/TDSTEST/SNATEST/FANOUT.C"
direction=s protocol=CPIC
```

例：管理対象転送

CPI-C ユーザにとっての、同等の管理対象転送は、以下のとおりです。この例は、Windows ユーザがファイルを複数のプラットフォーム経由で 4 台の異なるコンピュータに転送するケースに該当します。転送ごとに、-t オペランドとそのパラメータ（protocol= および retry_interval=）を完全な形で指定する必要があります。

```
dtsccli -t "ipath=SWTZDAMY::DT0.DTS20.C(PWTCIP)"
"rpath=A44ISTD1::PUBLIC.AAKEHH.PWTCIP.TS094"
protocol=CPIC retry_interval=30
-t "ipath=SWTZDAMY::DT0.DTS20.C(PWTCIP)"
"rpath=A04ISTD1::PUBLIC.AAKEHH.PWTCIP.TS084"
protocol=CPIC retry_interval=30
-t "ipath=SWTZDAMY::DT0.DTS20.C(PWTCIP)"
"rpath=KWWIGP21::C:¥TDSTEST¥SNATEST¥FANOUT.C"
protocol=CPIC retry_interval=30
-t "ipath=SWTZDAMY::DT0.DTS20.C(PWTCIP)"
"rpath=KWWIGP51::/USR/AAKEHH/TDSTEST/SNATEST/FANOUT.C"
protocol=CPIC retry_interval=30
```

SPX プロトコルのパラメータ

エージェント間転送の dtsccli コマンドに対して有効な SPX パラメータは、以下のとおりです。

SPX_IPORT=nnn

DTS エージェントが SPX 接続をリスニングするポート番号を指定します。

デフォルト：8226

NETADDR=nnnnnnnn

応答側のアドレスの代わりに使用する IPX アドレスの、ネットワーク番号部分を指定します。

NODEADDR=nnnnnnnnnnnn

応答側のアドレスの代わりに使用する IPX アドレスの、ノード番号部分を指定します。

IPX アドレスのフォーマットは *network-number.node-number* のように、8 桁.12 桁の書式で書式設定されます。たとえば、789EE001.00805F0EC8D8 となります。

開始側 DTS エージェントによって、応答側アドレスを 8 桁.12 桁の書式のアドレスとして認識できない場合、NETADDR および NODEADDR のプロトコルパラメータが検索されます。応答側アドレスが 8 桁.12 桁の書式の場合は、SPX プロトコルのパラメータを指定する必要はありません。ただし、応答側アドレスが IPX 書式ではない場合があるため、SPX のパラメータが用意されています。

プロトコルパラメータで SPX アドレスを使用する場合、以下のように、8 桁.12 桁の書式でアドレスを指定します。

```
"protocol_parameters=NETADDR=network-number NODEADDR=node-number"
```

以下に例を示します。

```
"protocol_parameters=NETADDR=89EE001 NODEADDR=00805F0EC8D8"
```

例：ポイントツーポイント転送

toro444 という名前のコンピュータから toky0001 に、SPX プロトコルを SPX NETADDR= および NODEADDR= パラメータと共に使用してポイントツーポイント転送を送信します。

```
dtsccli -a "ipath=toro444::c:¥jpegs¥cows¥bessie.jpeg" protocol=spx
"protocol_parameters=NETADDR=789EE001 NODEADDR=00805F0EC8D8"
"rpath=toky0001::c:¥jpegs¥cows¥bessie.jpeg"
```