CONFIDENTIAL

CspStat.Dll

CSP プリンターステータスAPI

関数リファレンス

Ver. 0.7.0.0_D

大日本印刷株式会社

2022.06.01

改訂履歴管理表		制定日 2007年 5月 31日			頁 1/8	
改訂頁	改 訂 内 容	改訂番号	ドキュメント番号	改定年月日	承認	作成
	CX、CXW（DS40,DS80）、CXM（タワー）対応初版 関数リファレンス仕様制定	0.4.0.0	0.40	2007/05/31		
P42～	コマンド追加 （電源電圧、最大電流、ソータ用紙位置センサー等）	0.5.0.0	0.50	2007/07/23		
P22 P35	ステータス取得コード変更 Gr.No SrNo.初期値設定仕様変更	0.5.1.0	0.51	2007/07/27		
P22、23	ステータス取得コード変更（GROPU_CXM1,2,3,他）	0.5.2.0	0.52	2007/08/10		
P34	ミニラボタワーステータスコード取得コマンド追加	0.5.3.0	0.53	2007/08/14		
P22 P34	ステータスコード追加 ミニラボタワーステータスコード変更	0.5.4.0	0.54	2007/08/22		
P24 P25	プリンターカウンタ値取得変更 プリンターカウンタ値設定変更	0.5.5.0	0.55	2007/08/23		
P13	タワーコントローラとの通信（Version,Serial No）も可能となりました。	0.5.6.0	0.56	2007/08/27		
P18	GetFirmwareVersion()の取得不具合を修正しました。	0.5.7.0	0.57	2007/08/29		
P8 P49	ダイレクトDLL使用時、Windowsドライバ要求Wizardを停止する方法について追記。 GetMiniLabPrinterAdd()関数追加	0.5.8.0	0.58	2007/08/30		

改訂履歴管理表		制定日 2007年 5月 31日			頁 2/8	
改訂頁	改 訂 内 容	改訂番号	ドキュメント番号	改定年月日	承認	作成
	V0.5.8.0 に対し GetStatus()関数使用時のメモリリークに関して、プリンターUSB ケーブル未接続時の処理を修正。	0.5.9.0	0.59	2007/10/03		
P14 P18 P19 P35 P39,40 P44	SetMiniLabPowerVolt、SetMiniLabECurrentMax GetMiniLabPowerVolt、GetMiniLabECurrentMax 関数削除 KIOSK モデルの API 関数動作について追加 画像サイズ情報追加 解像度の設定追加 色調整ファイルをテンポラリーに送信追加 ミニラボ タワーエラーステータスの変更 グループ、シリアル番号設定に文章追加	0.6.0.0	0.60	2008/06/02		
	~~タワー自己診断機能 API 追加~~	0.6.0.0	~~0.61~~	~~2008/06/11~~		
~~P38~~	~~マット印刷、オーバーコート仕上げ制御 API 追加~~	0.6.0.0	~~0.62~~	~~2008/08/07~~		
P28-29	カウンタアップの動作追加	0.6.0.0	0.621	2008/09/18		
P17	Ver 0.5.9.0 に対して ＜Linux 機及び高速 PC 対策＞ ※注）イメージデータ転送方法の改良	0.5.9.1	0.621_1	2008/12/05		
P17	Ver 0.6.0.0 に対して ＜Linux 機及び高速 PC 対策＞ ※注）イメージデータ転送方法の改良	0.6.0.1	0.621_1	2008/12/05		
P40	Ver 0.6.0.0 に対して マット印刷、オーバーコート仕上げ制御 API 追加 (←履歴表内、ページ番号修正)マット印刷	0.6.2.2	0.622	2008/11/06		
P6 P15 P18	Ver 0.6.2.2 に対して Windwos Vista を追記 8.ファームウェアと制御データアップデート手順について追加 〈Linux 機及び高速 PC 対策＞ 注）イメージデータ転送方法の改良	0.6.2.3	0.623	2009/01/09		
P13-15 P19 P20 P31 P33 P36 P47	Ver 0.6.2.3 に対して 6.CSP Status API 関数一覧変更 プリンターフィルター機能 追加 USB 通信タイムアウト機能 追加 プリンターステータス タイムアウトエラー 追加 マットカウンタ 追加 マットカウンタ初期化 追加 印画リトライ制御コマンド追加	0.6.3.0	0.630	2009/10/31		
P24	イメージ転送 タワープリンター位置による出力サイズ制限 追加	0.6.3.0	0.630_1	2009/11/20		
	【Dll 修正】 Dll 起動時、SetMediaSize()未指定でイメージ送信した場合のプリンターエラーを回避	0.6.3.1	0.631	2010/02/10		

改訂履歴管理表		制定日 2007年 5月 31日			頁 3/8	
改訂頁	改 訂 内 容	改訂番号	ドキュメント番号	改定年月日	承認	作成
P7 P9 P22 P23 P25 P41,42 43,45	（ドキュメントのみの変更） Windows Vista(32bit)、Windows 7(32bit)を追記 2.Windows ドライバ要求・・・記述修正 SetMediaSize()関数 記述追加 SendImageData()関数 記述追加 対応ファームウェア記述重複部分削除 ファームウェア、制御データ書き換えコマンド関係 記述追加	0.6.3.1	0.631_1	2010/05/20		
P14,15, P18	＜新機種追加＞ プリンター CY、CW02 に対応	0.6.3.2	0.632	2010/10/28		
P15, P48-50	・リージョンコードの取得機能追加 ・プリンターリージョンコードクリア機能追加 ・メディア種別取得機能追加 ・RF-ID 予備データ取得機能追加	0.6.3.3	0.633	2010/11/30		
P14,15, P18 P50	＜新機種追加＞ ・プリンター DS-RX1 に対応 ・メディア初期枚数取得機能追加	0.6.4.0	0.640	2011/01/28		
P15 P49	・プリンターリージョンコード設定機能の追加	0.6.4.1	0.641	2011/02/28		
P15 P49,50	・リージョンコードの関数を削除	0.6.4.2	0.642	2011/05/12		
P35 P45	・Copyright 情報の変更 ・カウントアップ一覧表に 2 インチカットの場合を追加 ・カッター制御コマンドに 2 インチカットモードを追加	0.6.4.3	0.643	2011/09/14		
P7 P21 P35	・著作権表記変更 ・メディアサイズ設定のヘッダー定義内容に L2 画面割付追加 ・カウントアップ一覧表にマルチカットモード 5x3.5x2 追加	0.6.4.4	0.644	2013/04/22		
P14,15, P18,21, P23,30, P35,38	＜新機種追加＞ プリンター DP-TC10 に対応	0.6.4.5	0.645	2013/06/14		
P14-18, 21,25,27 -29, 32,35,36 41,42,44 ,53,54	＜新機種追加＞ ・DP-DS80D（両面対応 8inch プリンタ）に対応	0.6.5.0	0.6.5.0	2013/06/28		
P31 P33 P34 iv - v	・両面エラーコード変更 ・反転ユニットステータス追加 ・カット紙情報取得追加 ・両面プリンター コンビネーションチャート追加	0.6.5.1	0.6.5.1	2013/08/05		
P35 P37 P48 P68	(ドキュメントのみの変更) ・カウントアップ動作の説明を修正 ・カウントアップ動作に DP-TC10 の L カードサイズ 2 画面連続印画時のカウント値を追加 ・カッター制御コマンドに DP-TC10 の L カードサイズ 2 画面連続印画指定を追加 ・補足に DP-TC10 の L カードサイズ 2 画面連続印画の説明を追加	0.6.5.1	0.6.5.1_1	2013/10/24		

改訂履歴管理表		制定日 2007年 5月 31日			頁 4/8	
改訂頁	改 訂 内 容	改訂番号	ドキュメント番号	改定年月日	承認	作成
P9,10, 20,23, 36,37 41, v - vi	・カット用紙、マルチカット用紙サイズ追加 ・カウンタ Duplex 追加 ・両面、裏面キャンセルと用紙排出コマンド追加 ・コンビネーションチャート変更 ・対応機能、確認結果変更	0.6.5.3	0.6.5.3	2013/11/14		
P56	・反転ユニットバージョン情報取得追加	0.6.5.4	0.6.5.4	2013/11/27		
P53 P73	(ドキュメントのみの変更) ・オーバーコート仕上げ制御コマンドに DP-TC10 の部分マット指定を追加 ・補足に DP-TC10 の部分マット印画についての説明を追加	0.6.5.4	0.6.5.4_1	2014/01/17		
v	(ドキュメントのみの変更) ・コンビネーションチャート変更	0.6.5.4	0.6.5.4_2	2014/03/28		
P25,46 P12 P12 P50 P56 P58 P65 P66 P67 P68 P69 P11	・DS80D カット紙 8x6x2 サイズ追加 ・対応機種に DP-DS620 を追加 以下のコマンドを追加 ・リボン巻き戻し機能を追加 ・リボン巻き戻し動作処理例を追記 ・L 判/PC 換算メディア残り発行枚数の取得 ・色制御データバージョン取得コマンド <種別指定> ・色制御データチェックサム取得コマンド <種別指定> ・フルカッターセットアップ ・スタンバイモード移行時間設定 ・スタンバイモード移行時間設定値の取得 ・メディアエンド保持モード設定 ・メディアエンド保持モード設定値の取得 ・USB iSerialNumber 有効／無効 設定 ・USB iSerialNumber 設定値の取得 ・Rewind モード設定 ・Rewind モードの取得 ・スタンバイモード時の動作について追記	0.6.6.1	0.6.6.1_1	2014/08/22		
v P17 P19 P54,55 P21	・コンビネーションチャート変更 ・6.CSP Status API 関数一覧変更 ・11-2 反転ユニットファームウェア書換え手順を追加 ・反転ユニットファームウェア書き換えデータの送信コマンドを追加 ・反転ユニットファームウェア書換え状態の取得を追加 ・DS-621 記述削除	0.6.7.0	0.6.7.0	2014/09/08		
P69 P12 P48,90 P14,16	・メディアエンド保持モードの工場出荷時設定値変更 (0:非保持→1:保持) ・L 巻き戻し に対応 ・DP-DS620 (A)では巻き戻し動作をしないことを明記 ・補足資料 (4) DP-DS620 リボンエンドチェック動作 追加 ・スタンバイモード中に使用可能な API にバージョン情報の取得を追加	0.6.7.0	0.6.7.0_1	2014/10/31		

改訂履歴管理表		制定日 2007年 5月 31日			頁 5/8	
改訂頁	改 訂 内 容	改訂番号	ドキュメント番号	改定年月日	承認	作成
P1 P19 P12,24 ,28,45 P50-51 ,92 P38 P40	・Windows2000 を対応 OS から削除 ・「11.ファームウェアと制御データアップデート手順について」の制御データアップデートシーケンスの順序を修正 ・DP-DS620 用紙サイズ追加(6x9、6x4.5、6x4.5x2) ・DP-DS620 6x9(A5W)メディア追加 ・ドキュメントの両面ユニットの文言を反転ユニットに変更 ・プリンターステータスの取得の説明から一部文言の削除 ・反転ユニット、ステータスの取得の説明から一部文言の削除	0.6.7.2	0.6.7.2_D	2015/02/27		
	・64bit OS に対応	0.6.7.2	0.6.7.2_D_2	2015/03/31		
P68 P17,69 P64 P62 P17,75 P92-93 P95-96	・フルカッターセットアップの 6x4、5x7、6x9 メディア対応 ・中間層設定に対応したフルカッターセットアップ拡張設定関数を追加 (SetFullCutterSetUpEX) ・オーバーコート仕上げ制御コマンドに、ファインマット、部分マット(マット)、部分マット(ファインマット)を追加 ・カッター制御コマンドにパノラマ印画動作を追加 ・パノラマ印画開始可能チェック関数を追加 (GetPanoramaPrintable) ・補足資料 (3) 部分マット印画についてに、DP-DS620 の部分マット機能の内容を追加 ・補足資料 (5) パノラマ印画を追加	0.6.7.3	0.6.7.3	2015/04/28		
P24, 28,45 P17, 68-70	・DS40 6x6 用紙サイズ追加 ・DS40 フルカッターセットアップ対応 (用紙サイズ 6x8 のみ)	0.6.7.5	0.6.7.5	2015/06/30		
P55 P17 P57-59	・DP-DS80D 反転ユニットファーム書換えモード注記追加 ・DP-DS80D 反転ユニットエラーログ取得機能追加 反転ユニットエラーログ作成要求 反転ユニットエラーログデータ長の取得 反転ユニットエラーログデータ受信 反転ユニットエラーログ作成状態の取得	0.6.7.6	0.6.7.6	2015/07/22		
P17, P70, P78, P76 (vi-vii)	・DS40 下記の機能を追加 メディア初期枚数取得 メディアオフセット枚数の取得 USB iSerialNumber 有効／無効設定 USB iSerialNumber 設定値の取得 ・DP-DS80D コンビネーションチャート、対応機能確認結果削除	0.6.7.7	0.6.7.7	2015/08/31		
P17, P78, P76 P67	・DS-RX1 下記の機能を追加 メディアオフセット枚数の取得 USB iSerialNumber 有効／無効設定 USB iSerialNumber 設定値の取得 ・オーバーコート仕上げ制御コマンドに、ラスターを追加	0.6.7.8	0.6.7.8	2015/12/25		
P17,80	・連続パノラマ印画設定関数を追加	0.6.7.9	0.6.7.9	2016/02/29		

改訂履歴管理表		制定日 2007年 5月 31日			頁 6/8	
改訂頁	改 訂 内 容	改訂番号	ドキュメント番号	改定年月日	承認	作成
P12,14 15-17 20-22 25,29,35, 37,44, 46-47 52-53 64, 66-83 102	＜新機種追加＞ DP-DS820 対応 ・プリント速度指定コマンドを追加 ・色制御データバージョン取得（種別、メディア種別）コマンドを追加 ・色制御データチェックサム取得（種別、メディア種別）コマンドを追加 ・PC/L 換算メディア残り発行枚数の取得 ⇒ ハーフサイズ換算メディア残り発行枚数の取得に変更	0.6.8.0	0.6.8.0	2016/02/29		
P17,18	・スタンバイモード中に使用可能な API にメディア種別の取得を追加。(DP-DS820)	0.6.8.0	0.6.8.0_1	2016/03/18		
P1	・Windows10に対応	0.6.8.1	0.6.8.1	2016/03/31		
ix P17 P88-89	・以下のAPI関数を追加 -ガンマテーブルの設定 -ガンマテーブルチェックサム取得 -データテーブルのクリア	0.6.8.2	0.6.8.2	2016/05/27		
P72 P14 P47	（ドキュメントのみの変更） メディア種別の取得に、メディア種別の説明を追記。 以下の項の表のDP-DS820関連部分を更新 － 8.スタンバイモード スタンバイモード使用可能API － カウンタ値の取得 カウンタL/A/B/P カウントアップの動作	0.6.8.2	0.6.8.2_2	2016/06/24		
P71, P81,82, P83, P106,107	以下の項が変更されました。 － 印刷リトライ制御コマンド － パノラマ印画開始可能チェック － 連続パノラマ印画設定 － 補足資料：余白ありパノラマ印画	0.6.8.3	0.6.8.3_D	2016/09/13		
P70,103	オーバーコート仕上げ制御コマンド“ファインマット“対応（DP-DS620）	0.6.8.3	0.6.8.3_D2	2017/02/28		
P25,29, 46 P37 P46 P52	・DS-RX1 5x5,6x6 用紙サイズ追加 ・メディアコードの説明の内容を一部変更 ・マルチカット（L2画面割付）の注釈にDS-RX1に関する内容を追記 ・印刷可能枚数一覧表を更新 － DS-RX1の6x4(PC)サイズに350画面を追加 － DS-RX1の5x7(2L)サイズに400画面を追加	0.6.8.5	0.6.8.5_3_D	2017/07/28		
P52	メディア残り発行枚数、DP-DS620 5x3.5(L) 対応 未使用パラメータの削除。(マット 2、3)	0.6.8.7	0.6.8.7_D	2019/02/28		

改訂履歴管理表		制定日 2007年 5月 31日			頁 7/8	
改訂頁	改 訂 内 容	改訂番号	ドキュメント番号	改定年月日	承認	作成
P14,17, 19-23, 25-27, 30,33,37 40,42-43, 53,57-58, 69-70 72-81,83, 87-88,93, P96 P96 P55 P40	＜新機種追加＞ DP-QW410 対応 以下の項が追加されました ・デカール制御設定 ・現在時刻通知 ・負の値を指定した場合の説明を追加 ・誤記の修正	0.6.9.1	0.6.9.1_1_D	2019/12/24		
P30,33, 42,53,57, 74,81	・DP-QW410 4.5x6対応	0.6.9.3	0.6.9.3_D	2020/04/24		
P21,23 P14, 21-23,25, 70-71, 73-74 P75 P80 P55 P55 P99 P16-19, 22-23,27 30,94-95 P117	・API 関数一覧を変更 ・DP-DS620 統合ファームウェア対応 (DP-DS620 ファームウェアバージョン 3.00 以降のもの) ・カッター制御コマンドに 2 面付系カットくず無し指定パラメータを追加 ・メディア種別の取得コマンドにメディア種別情報を追加 ・以下の API 関数を追加 - 拡張ライフカウンタ値の取得 - 拡張マットカウンタ値の取得 - 対応メディア情報の取得 ・誤記改訂 ・弊社確認パターンの図を変更	0.6.9.7	0.6.9.7_D	2020/09/09		
vii P23,27, 58,59,99	(ドキュメントのみの変更) ・DP-DS620 統合ファームウェア対応	0.6.9.7	0.6.9.7_1_D	2021/05/20		
P2	・DLL を使用した印画・制御方法のイメージ図を変更	0.6.9.8	0.6.9.8	2021/07/02		

改訂履歴管理表		制定日 2007年 5月 31日			頁 8/8	
改訂頁	改 訂 内 容	改訂番号	ドキュメント番号	改定年月日	承認	作成
P30,33, 54,77,85 P27-30, 32,37-43, 46, 48-50 57-68, 70-79, 81-93,95, 97-103, 105-114 P22,92, 93	・DP-QW410 に以下の用紙サイズを追加 4x3, 4x4.5, 4.5x3, 4.5x4, 4x3x2, 4.5x3x2, 4.5x4x2 ・各 API 関数の記載内容を改訂 ・DP-DS820 Rewindモード対応	0.6.9.9	0.6.9.9_D	2021/11/30		
	・【DLL修正】 ・内部のDP-QW410における多面付印刷に関する処理を修正	0.7.0.0	0.7.0.0_D	2022/6/1		

目次

はじめに *1*
1. DLL を使用した印画・制御方法のイメージ *2*
2. Windows のドライバ要求 Wizard 停止ファームウェアについて *3*
3. ポート番号指定について *3*
4. 印画処理フロー *4*
5. 印画フロー補足　マルチカット印画 *6*
6. 両面プリンター　アプリケーション画像生成とプリンター動作、用紙の出力イメージの関係 *8*
7. リボン巻戻し機能　（DP-DS620, DP-DS820） *12*
8. スタンバイモード　（DP-DS620, DP-DS820） *14*
9. CSP Status API 関数一覧 *15*
10. KIOSK モデルの API 関数動作について *24*
11. ファームウェアと制御データアップデート手順について *25*
12. コマンド仕様 *27*
プリンター識別番号、接続数の取得 27
プリンターフィルター機能設定 28
USB 通信タイムアウト時間設定 29
メディアサイズ設定 30
イメージデータ転送 32
解像度の設定 37
印画枚数指定 37
プリント開始 38
ページレイアウトの開始 38
ページレイアウトの終了 39
バージョン情報の取得 39
センサー情報の取得 40
解像度の取得 41
メディアコードの取得 42
プリンターステータスの取得 43
反転ユニット、ステータスの取得（DP-DS80D） 46
反転ユニット、カット紙情報取得（DP-DS80D） 48
反転ユニット、裏面印画キャンセルと用紙排出（DP-DS80D） 49
カウンタ値の取得 50
拡張ライフカウンタ値の取得(DP-DS620) 57
拡張マットカウンタ値の取得(DP-DS620) 57
カウンタ値のクリア 58
カウンタ値（P）の設定 58
空きバッファ数の取得 59
残り発行枚数の取得 59
メディア残り発行枚数の取得 60
ハーフサイズ換算メディア残り発行枚数の取得（DP-DS620, DP-DS820) 61
メディアロット間色バラツキ補正値の取得 62

メディアタグ情報の取得 62
プリンターシリアルナンバーの取得 63
ファームウェア書き換えモードへの切り替えコマンド 63
ファームウェア書き換えデータの送信コマンド 64
反転ユニットファームウェア書き換えモードへの切り替えコマンド（DP-DS80D） 65
反転ユニットファームウェア書き換えデータの送信コマンド（DP-DS80D） 65
反転ユニットファームウェア書換え状態の取得（DP-DS80D） 66
反転ユニットエラーログ作成要求（DP-DS80D） 67
反転ユニットエラーログデータ長の取得（DP-DS80D） 67
反転ユニットエラーログデータ受信（DP-DS80D） 68
反転ユニットエラーログ作成状態の取得（DP-DS80D） 68
反転ユニットエラーログデータ取得手順 69
色制御データクリアコマンド 70
色制御データの送信コマンド 70
色制御データバージョン設定コマンド 71
色制御データバージョン取得コマンド 71
色制御データバージョン取得コマンド <種別指定> (DP-DS620, DP-DS820, DP-QW410) 72
色制御データバージョン取得コマンド<種別、メディア指定> (DP-DS620, DP-DS820, DP-QW410) 73
色制御データバージョンをテンポラリーへ設定（CW) 74
色制御データをテンポラリーへ送信（CW) 74
色制御データチェックサム取得コマンド 75
色制御データチェックサム取得コマンド <種別指定> (DP-DS620, DP-DS820, DP-QW410) 75
色制御データチェックサム取得コマンド <種別、メディア指定> (DP-DS620, DP-DS820, DP-QW410) 76
カッター制御コマンド 77
メディア ID 設定情報取得 78
一枚画像保持印画　制御コマンド（DS40) 78
オーバーコート仕上げ制御コマンド 79
印画リトライ制御コマンド 81
メディア種別の取得 (DP-DS620, DP-DS820) 82
RF-ID タグ予備データの取得（DS-RX1, DP-TC10, DP-DS620, DP-DS820, DP-QW410) 83
メディア初期枚数の取得（DS-RX1, DP-TC10, DS40, DP-DS620, DP-DS820, DP-QW410) 84
反転ユニット、ファームウェアバージョン情報の取得（DP-DS80D） 84
フルカッターセットアップDS40, DP-DS620, DP-DS820, DP-QW410) 85
フルカッターセットアップ拡張設定 (DS40, DP-DS620, DP-DS820, DP-QW410) 87
スタンバイモード移行時間設定（DP-DS620, DP-DS820) 89
スタンバイモード移行時間設定値の取得（DP-DS620, DP-DS820) 89
メディアエンド保持モード設定（DP-DS620, DP-DS820, DP-QW410) 90
メディアエンド保持モード設定値の取得（DP-DS620, DP-DS820, DP-QW410) 90
USB iSerialNumber 有効／無効 設定（DS40, DS-RX1, DP-DS620, DP-DS820, DP-QW410) 91
USB iSerialNumber 設定値の取得（DS40, DS-RX1, DP-DS620, DP-DS820, DP-QW410) 91
Rewind モード設定 (DP-DS620, DP-DS820) 92
Rewind モードの取得 (DP-DS620, DP-DS820) 93

Dai Nippon Printing Co., Ltd.

パノラマ印画開始可能チェック（DP-DS620, DP-DS820）..... 94
メディアオフセット枚数の取得（DS40, DS-RX1）..... 95
連続パノラマ印画設定（DP-DS620, DP-DS820）..... 96
プリント速度指定コマンド（DP-DS820, DP-QW410）..... 97
ガンマテーブルの設定（DP-DS820）..... 98
ガンマテーブルチェックサム取得（DP-DS820）..... 99
データテーブルのクリア（DP-DS820）..... 100
デカール制御設定..... 100
現在時刻通知..... 100
対応メディア情報の取得(DP-DS620)..... 101
汎用設定系コマンド..... 102
汎用取得系コマンド..... 102
ミニラボ タワーエラーステータスの取得..... 103
ミニラボ 印画データグループ No.　シリアル No. 初期値設定..... 105
ミニラボ 印画データグループ No. 設定..... 108
ミニラボ 印画データシリアル No. 設定..... 108
ミニラボ 裏面印刷データ 設定..... 109
印画中止コマンド..... 111
用紙排出先指定..... 111
ソータ用紙 MAX 位置検出センサーON/OFF　設定..... 112
ソータ用紙 MAX 位置検出センサーON/OFF　取得..... 112
ミニラボ グループ No. 取得..... 113
ミニラボ シリアル No. 取得..... 113
プリンター位置の取得..... 114
13.　補足資料..... 115
(1)DP-TC10 シングルカットモードについて..... 115
(2)DP-TC10 Lカードサイズ２画面連続印画について..... 116
(3)部分マット印画について..... 118
(4)DP-DS620、DP-DS820 リボンエンドチェック動作..... 120
(5)余白ありパノラマ印画..... 121

はじめに

本書の著作権は権利者にあります。一部または全部の無断転載・無断複製を禁止します。

本書の内容は予告なく変更することがあります。

Microsoft, Windows は、米国 Microsoft Corporation の米国およびその他の国における商標または登録商標です。

適用の範囲

本書は、CSP ステータス API 関数の取り扱いに関して記述したものです。

対応 OS、動作環境

本 API は WindowsXP、及び Windows Vista(32bit)、Windows 7(32/64bit) 、Windows 8(32/64bit) 、Windows 10(32/64bit) で動作します。

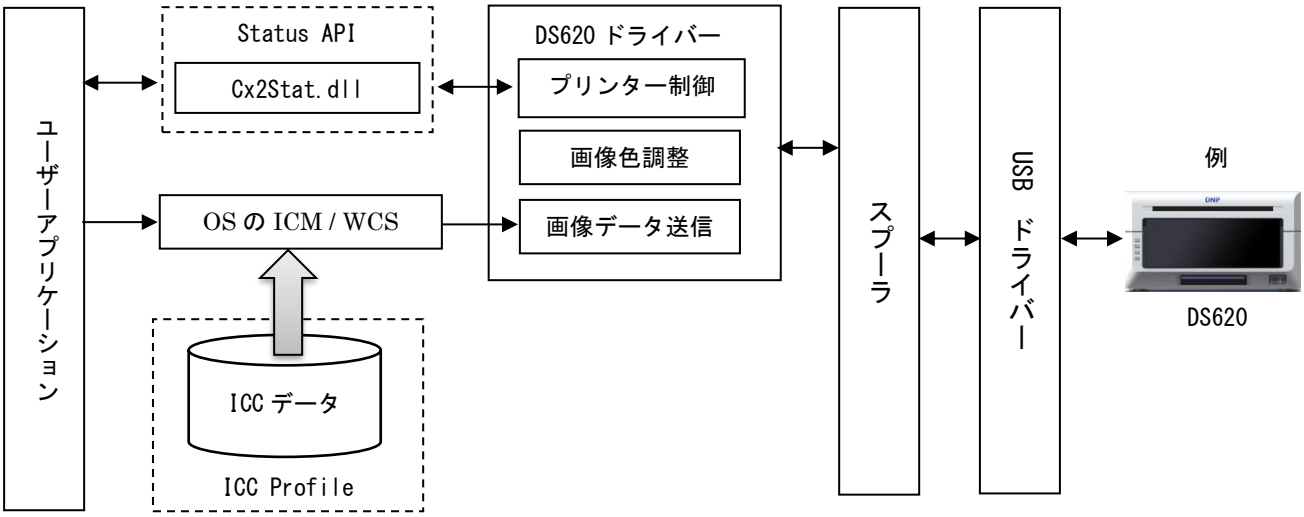
注意：本 API は 32bit 版と 64bit 版の 2 種類の DLL があります。

1. DLL を使用した印画・制御方法のイメージ

CSP_Status API はデータを直接ＵＳＢに転送しプリンターの印画・制御をします。プリンタードライバを介して行う印画・制御とは次のように異なります。

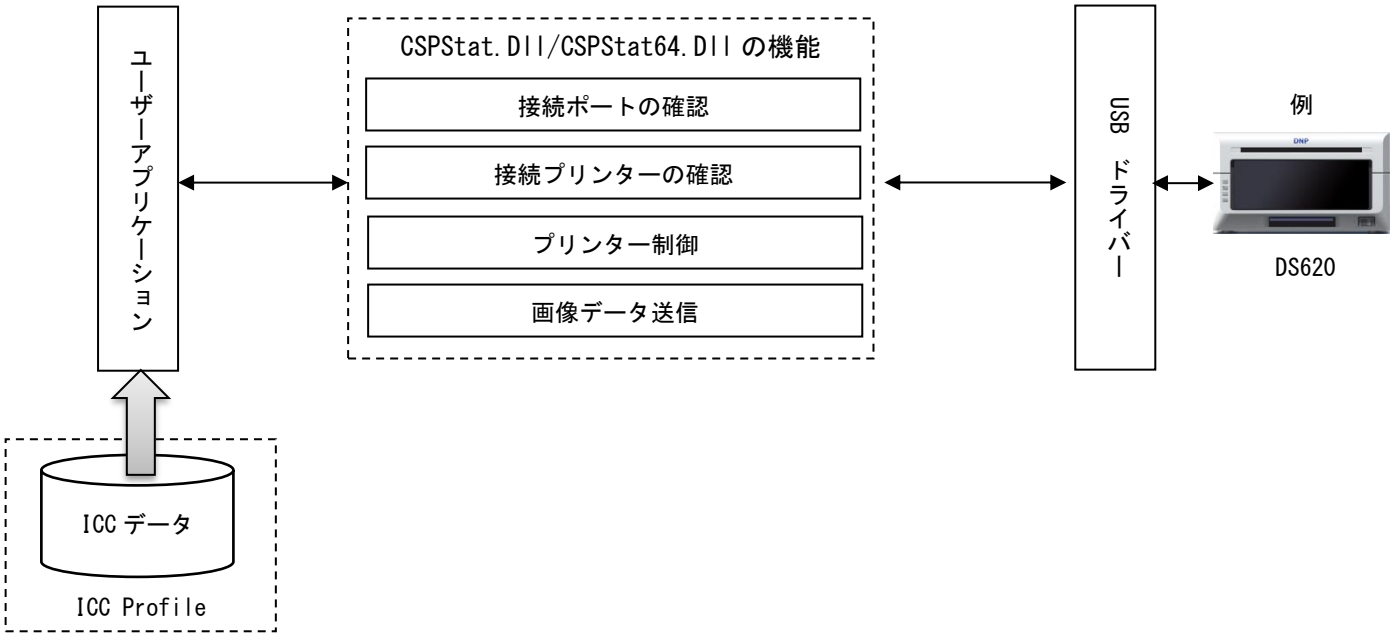
① プリンタードライバを介した印刷・制御方法

印画・制御データは機種毎のプリンタードライバを介して処理します。



② データを直接 USB に転送する印刷・制御方法

印画・制御データ共に直接 USB へ転送します。



2. Windows のドライバ要求 Wizard 停止ファームウェアについて

ユーザーがダイレクト DLL を使用する場合、タワー用ファームウェアを使用すると USB 接続時のドライバ要求 Wizard を停止することが可能です。

通常 Windows はプリンターが接続された時、USB の接続（ポート作成）⇒プリンターが返すデバイス名の認識⇒ドライバ要求 Wizard の起動を行いますが、タワー用ファームウェアは接続の時、プリンターからデバイス名を返さないので Windows はドライバ要求 Wizard が起動しなくなります。
Windows はポートを動的に割り振るので、新しいポートにプリンターが接続される度にドライバの要求を行いますが、タワー用ファームは、新しい USB ポートへプリンターを接続した場合でも、ドライバ要求 Wizard を停止することが出来ます。

注意）本ファームウェアでプリンタードライバから印刷したい場合は、手動でプリンタードライバをインストールする必要が有ります。

3. ポート番号指定について

①プリンタードライバを介した制御では、ポートイニシャライズで、ドライバに関連付けられたポートを指定していました。

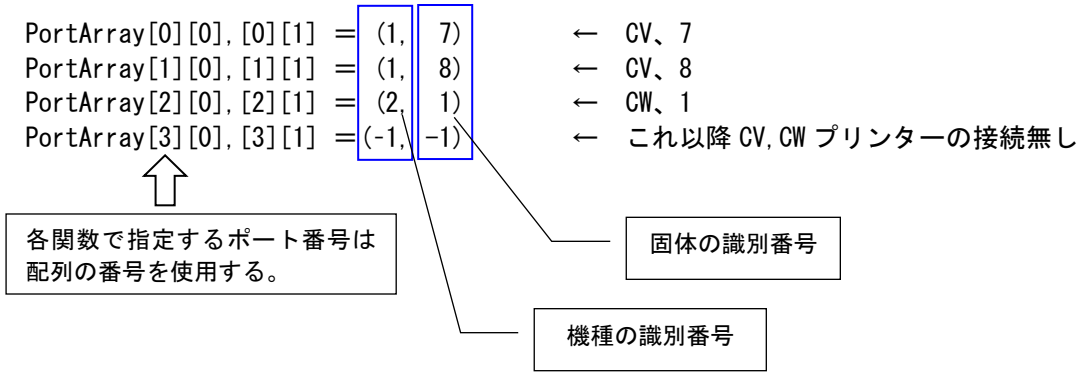
例）PortNo ＝ PortInitialize (StrPrt("USB001"))；

DLL の各関数ではこの API 関数で取得した PortNo を使用していました。“USB00n”はプリンタードライバに関連付けられたポート名ですので、プリンタードライバを介さない場合は次の②の関数を使います。

②データを USB に直接転送する方法では、アプリケーションが接続状態を確認してポート番号を指定する必要があります。
GetPrinterPortNum 関数を使用すると、以下のようにプリンターの接続状態が分かります。
DLL の各関数で使用するポート番号は、以下のような配列の番号となります。

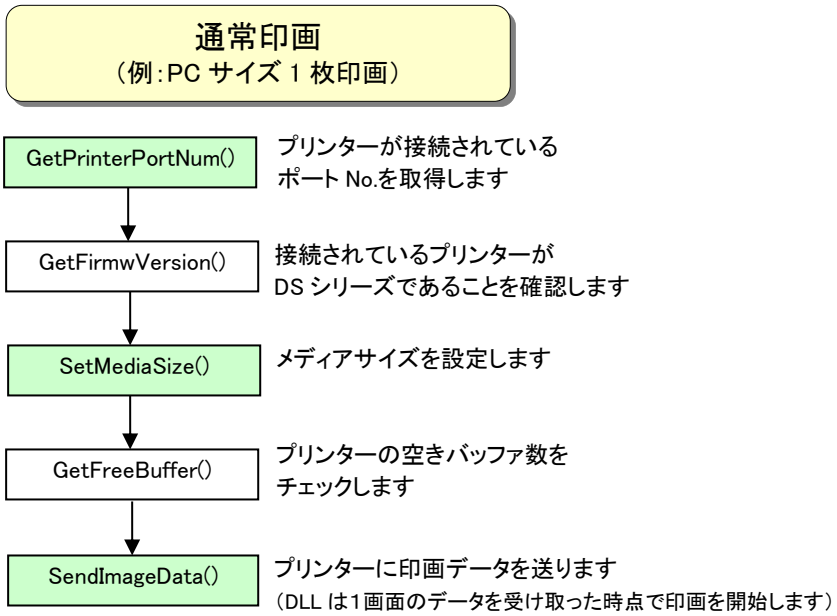
PortNum = GetPrinterPortNum(PortArray[ ][ ]);

例）各ポートに CV,CW が計３台接続されている場合。PortNum は 3 が入ります。

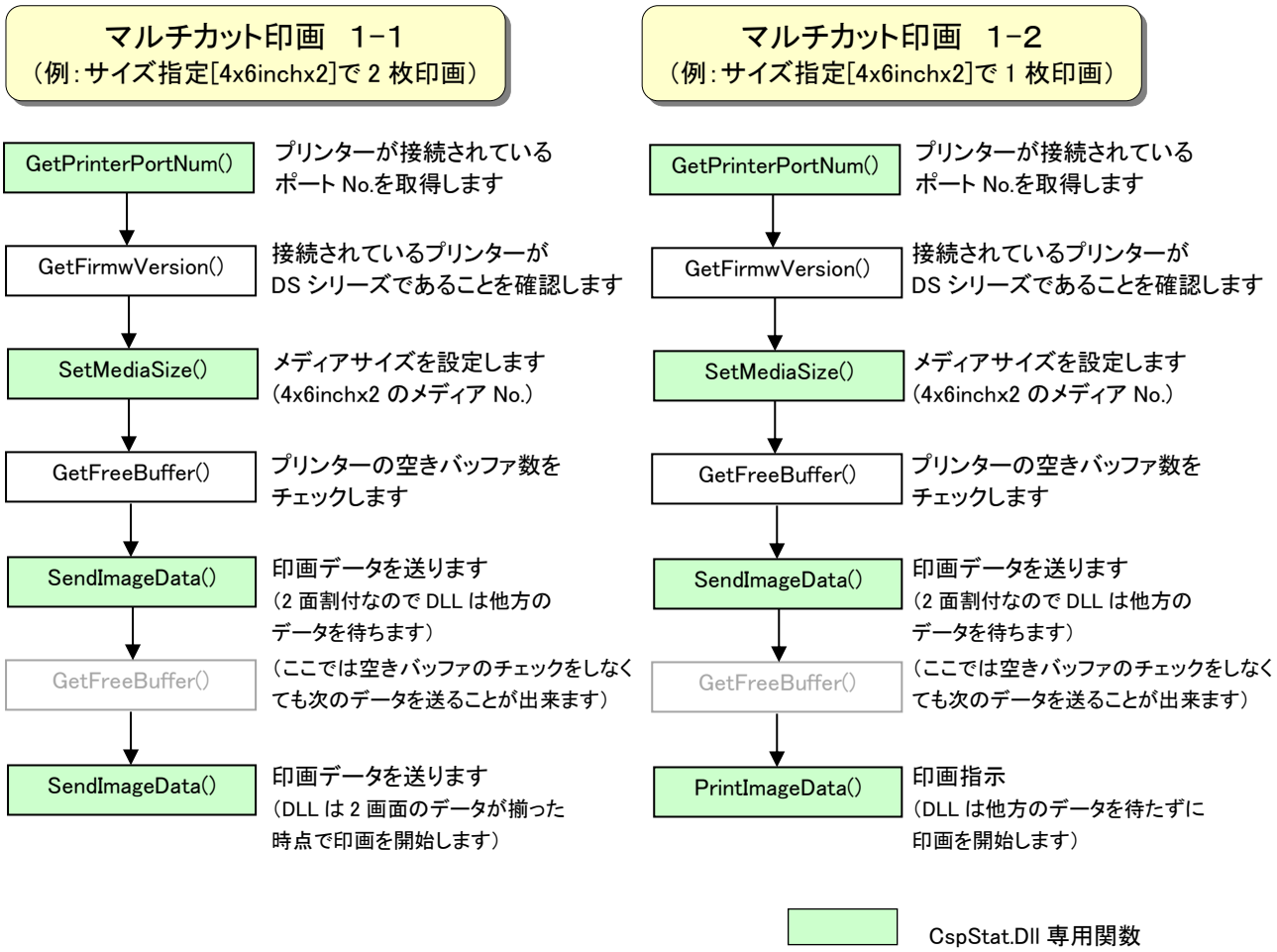


4. 印画処理フロー

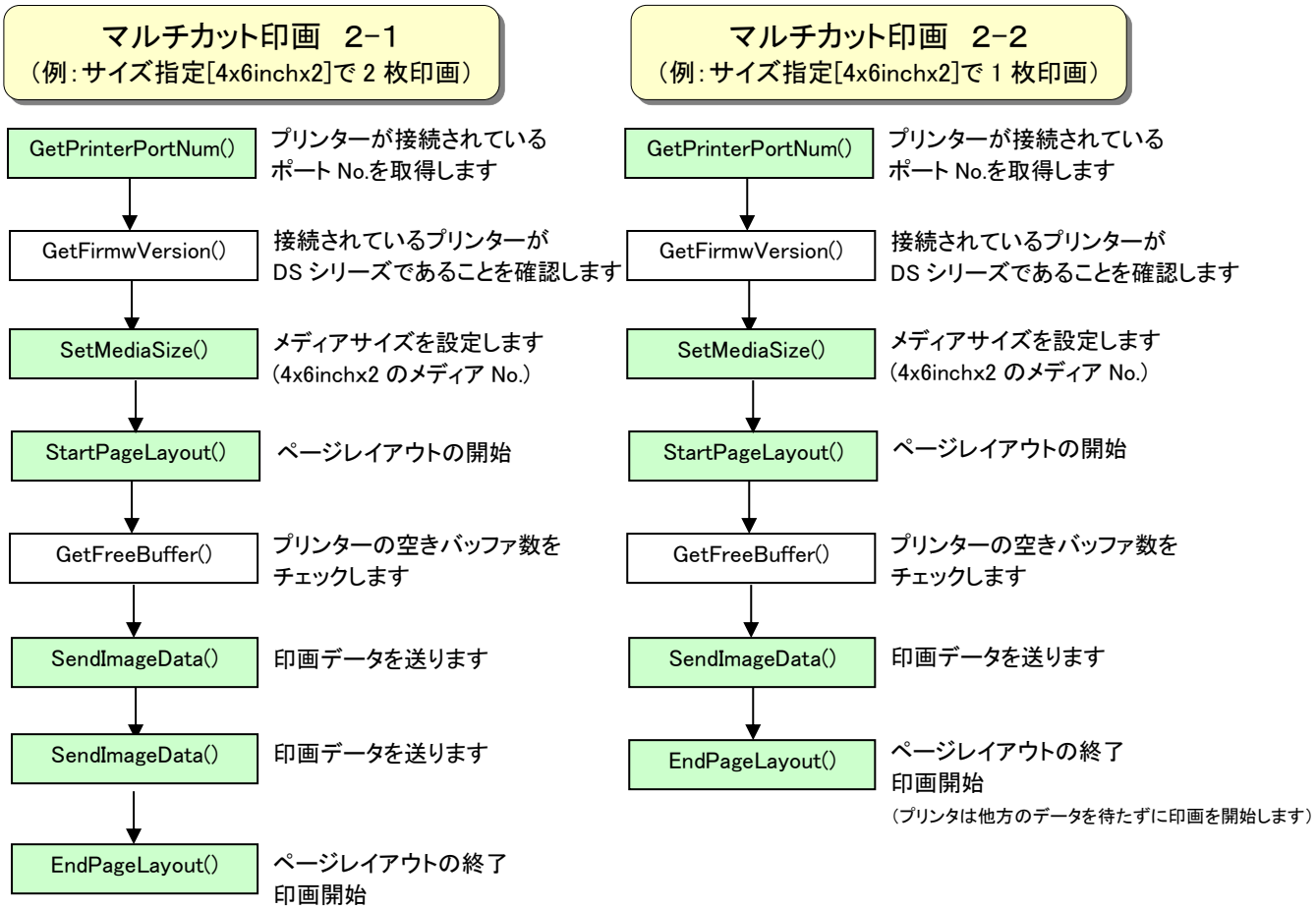
CspStat API の印画処理フローは次のようになります。



①ページレイアウト関数を使用しない場合のマルチカット印画処理



②ページレイアウト関数を使用した場合のマルチカット印画処理



マルチカット印画は①、②どちらの方法でも指定可能です。プリンターは一回に印刷するページデータを一括転送する方法なので、DLL、プリンタードライバ内でマルチプリントデータを合成して転送します。どちらのコマンドを使用しても、必要画面数が揃った場合、データを転送します。

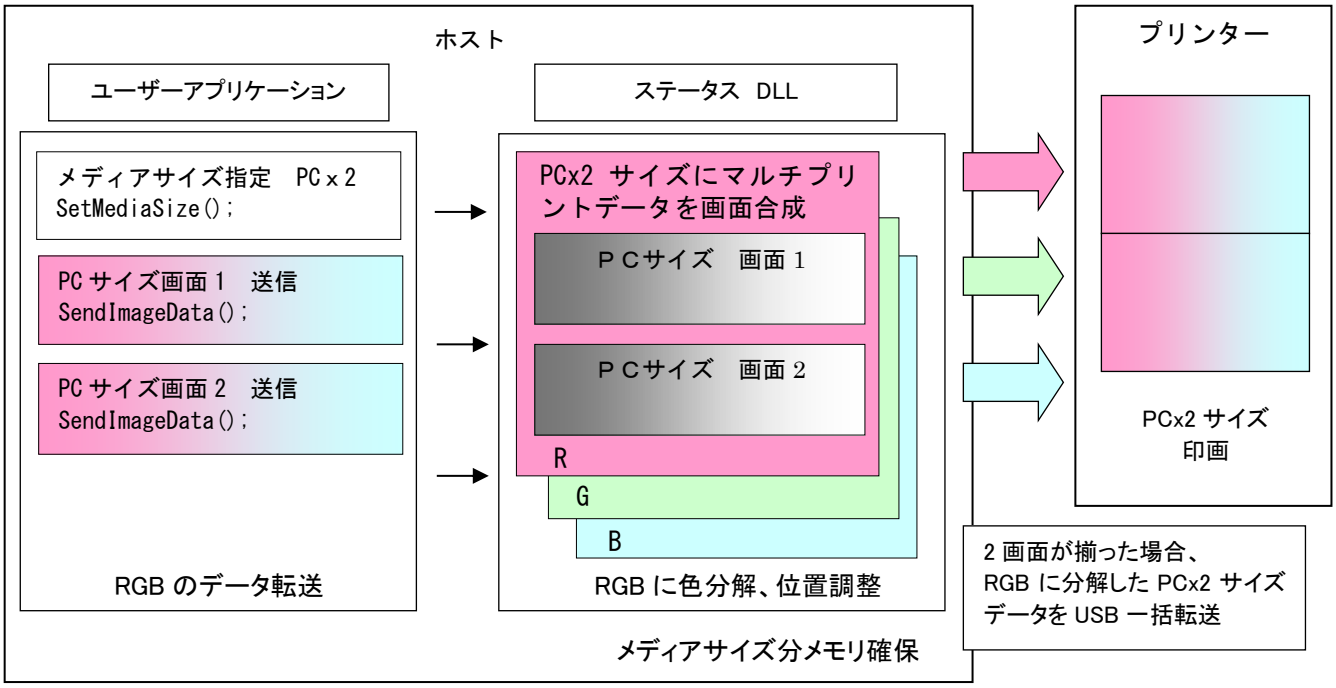
5. 印画フロー補足　マルチカット印画

マルチカット印画は下記の①、②どちらの方法でも指定可能です。プリンターは一回に印刷するページデータを一括転送する方法なので、DLL 又はプリンタードライバはホストのメモリ内にマルチプリントデータを合成してからプリンターへ転送します。

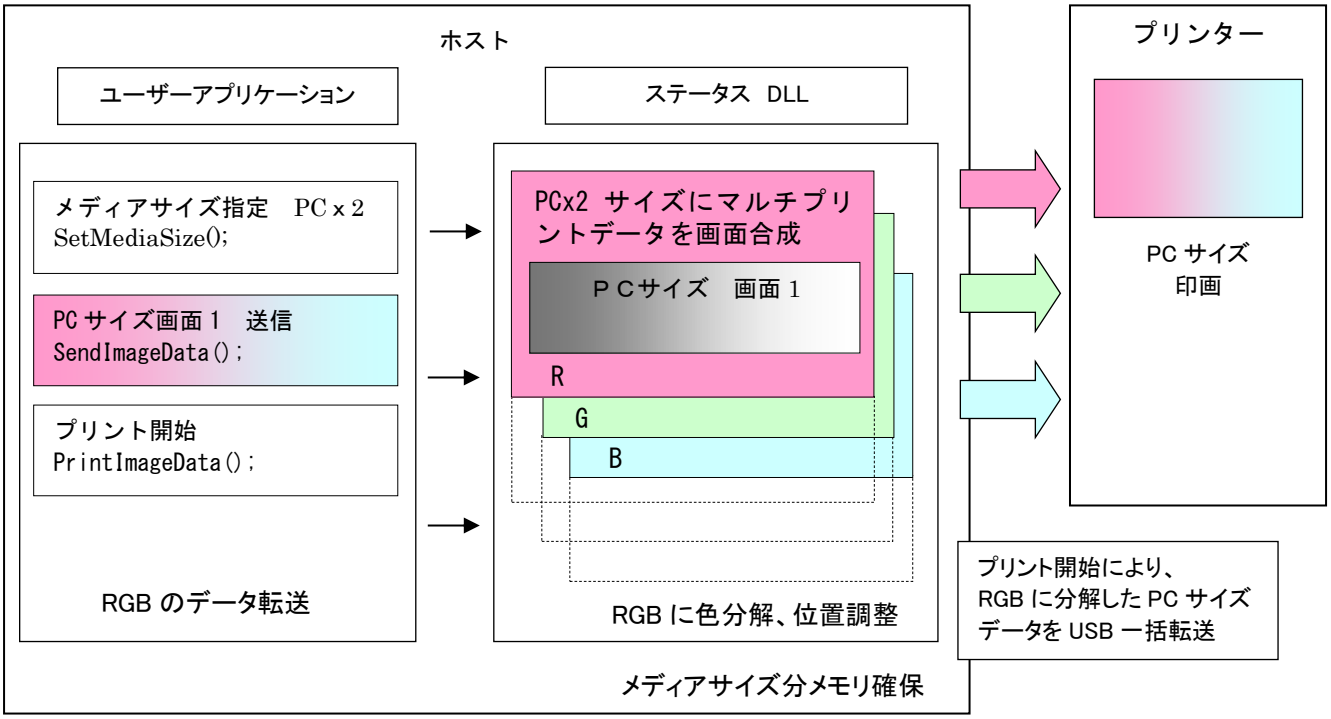
① ページレイアウト関数を使用しない場合のマルチカット印画処理

ページレイアウト関数を使用しない印画では、ステータス DLL が確保したメモリ上に必要画面数が揃った場合、データをプリンターへ転送します。

1-1　2 面割付の場合のホスト処理イメージ



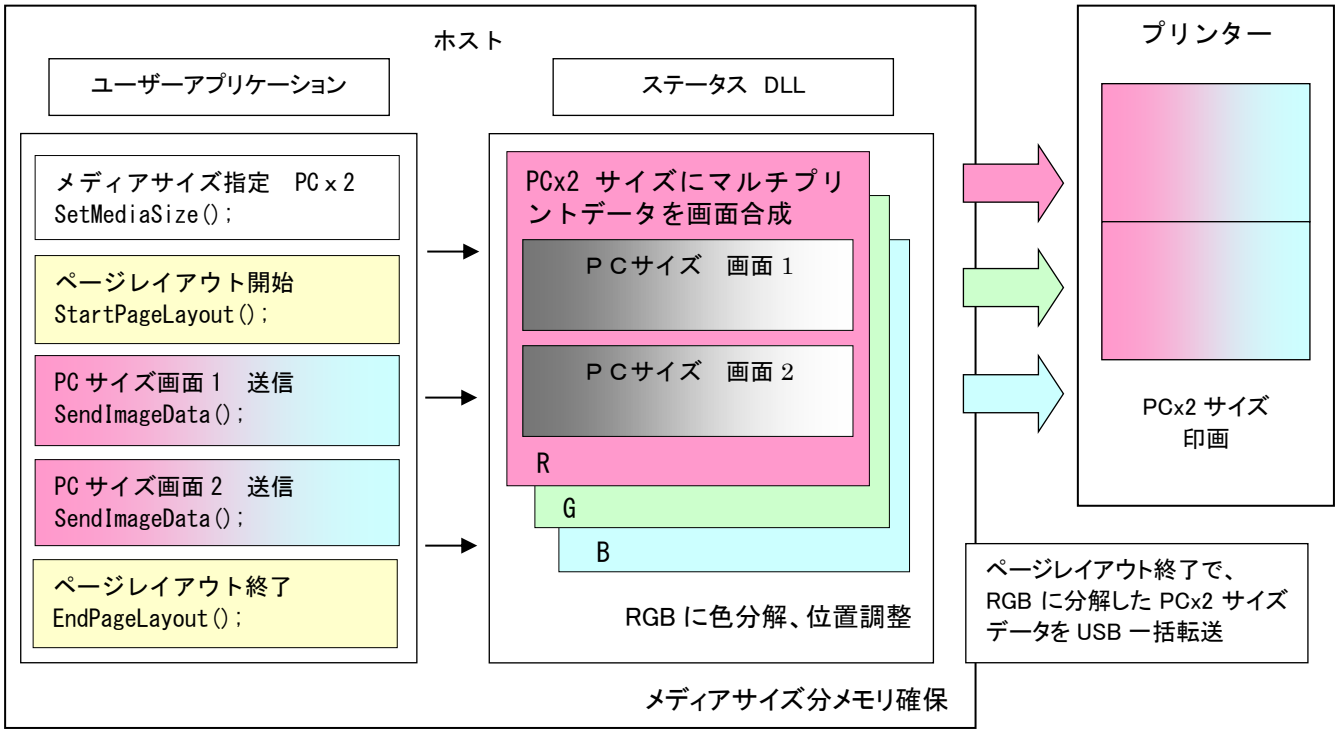
1-2　一画面途中でプリント開始を指定した場合のホスト処理イメージ



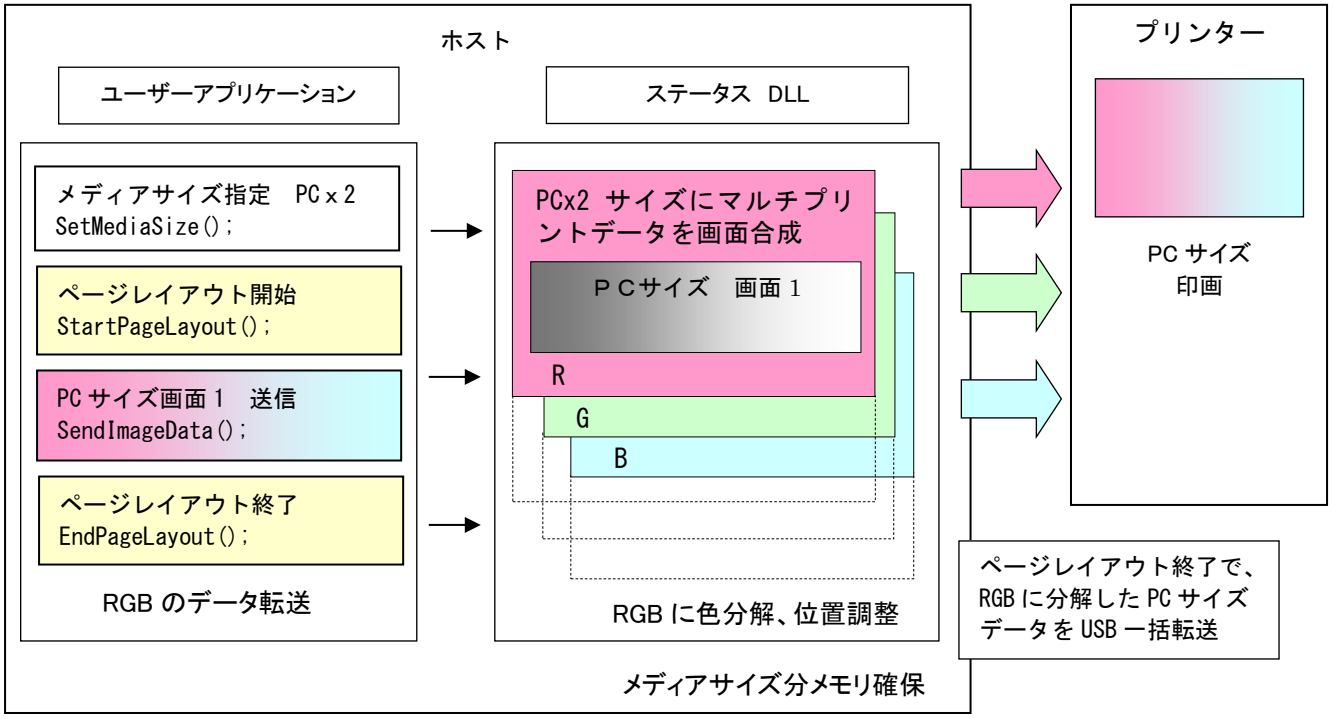
② ページレイアウト関数を使用した場合のマルチカット印画処理

ページレイアウト関数を使用した場合、ホストはページレイアウトが開始されると、終了されるまでプリンターへのデータ転送をしません。これによりアプリケーションはデータ転送を明示的に行うことが出来ます。

2-1　2 面割付の場合のホスト処理イメージ



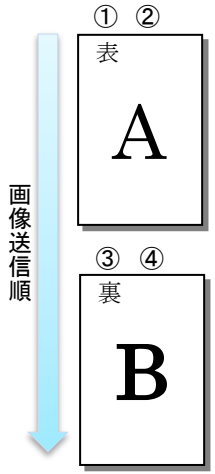
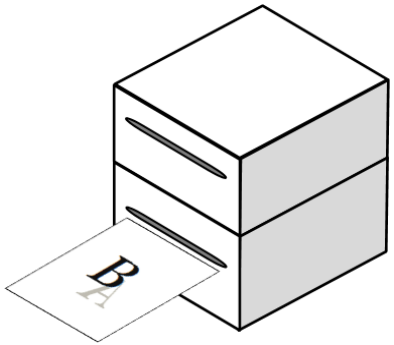
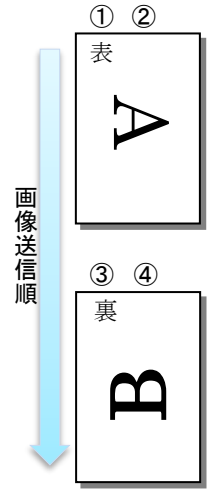
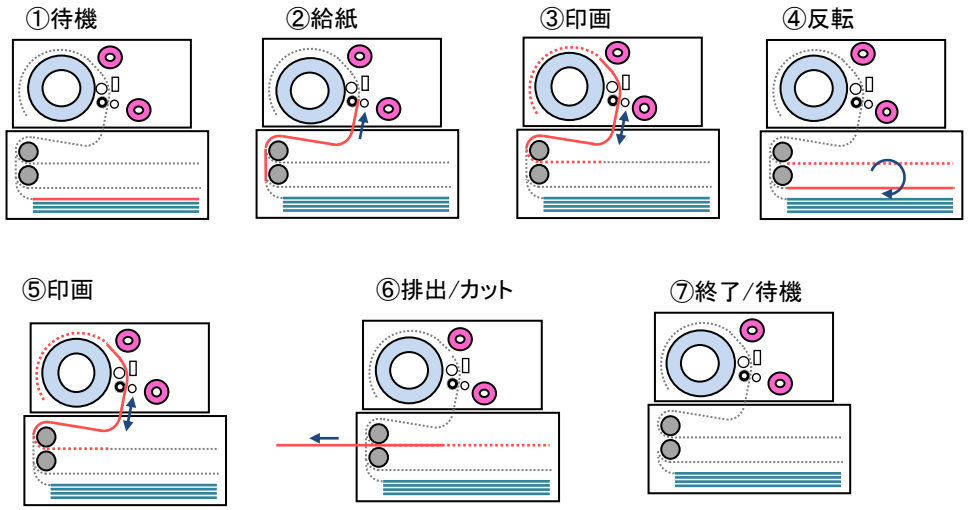
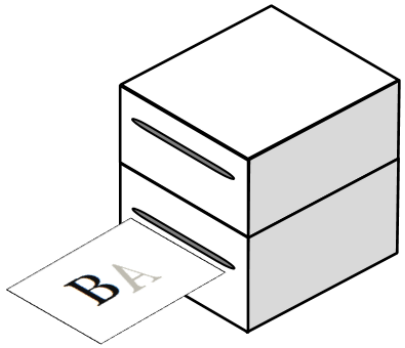
2-2　一画面途中でプリント開始を指定した場合のホスト処理イメージ



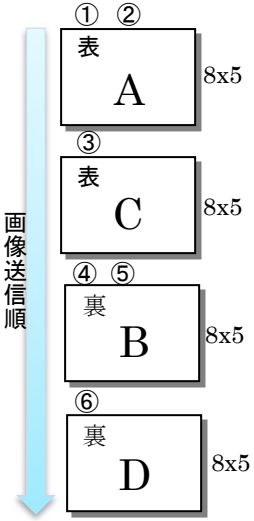
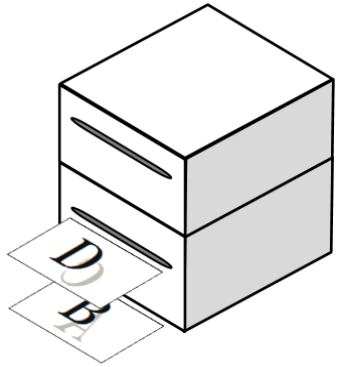
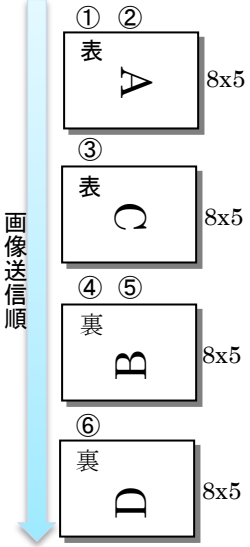
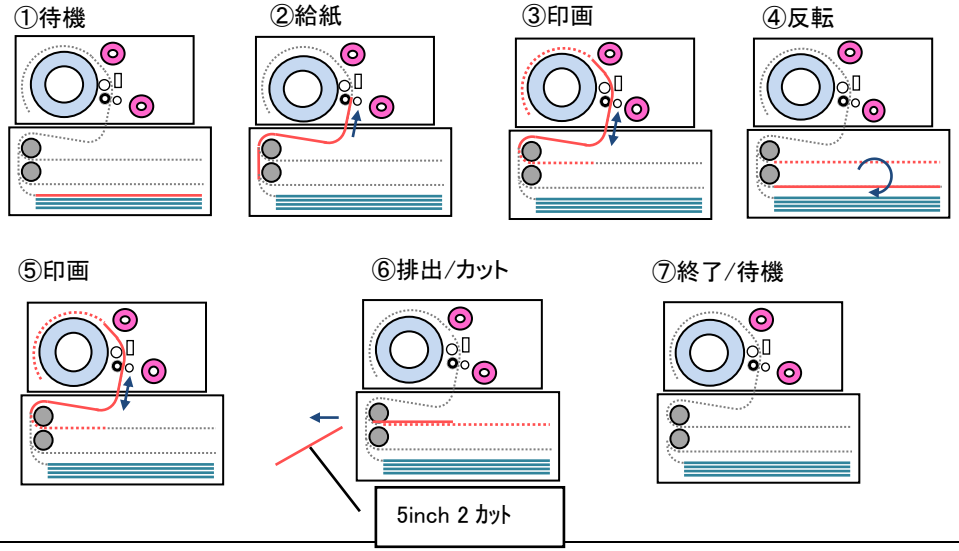
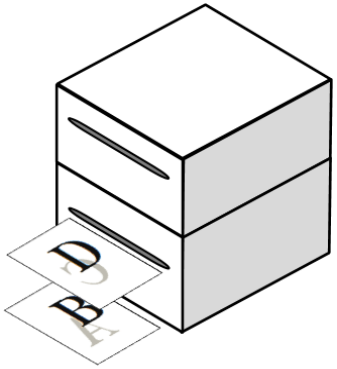
6. 両面プリンター　アプリケーション画像生成とプリンター動作、用紙の出力イメージの関係

アプリケーションは、意図する出力結果に応じて画像をプリンターへ送信する順番を決めて、綴じ方向などの条件により、回転処理を加えて出力サイズに応じた転送画像を生成します。
プリンターへ送信する画像の向きと出力される印刷物の関係を下表に示します。

◆ 両面印画を行う場合　（両面カット紙、面付けなし。片面印画、もう一方にオーバーコート処理を行う場合も同様。）

用紙種 出力サイズ	両面指定	画像生成、 画面割付と送信順		画像送信手順とプリンターの動作	用紙出力イメージ
両面 カット紙 1 面割付	長辺綴じ		⇒	画像送信手順 ①用紙サイズ指定　SetMediaSize()　例：8x10（表面）の場合　引数 231 を指定 ②BMP 画像 A 送信　SendImageData() ③用紙サイズ指定　SetMediaSize()　例：8x10（裏面）の場合　引数 331 を指定 ④BMP 画像 B 送信　SendImageData() （片面印画、もう一方にオーバーコート処理を行う場合の例） ※用紙の片面に印画、もう一方にオーバーコート処理のみ行いたい場合、表、裏面どちらかに空白 BMP 画像を送信して下さい。 ①用紙サイズ指定　SetMediaSize()　例：8x10（表面）の場合　引数 231 を指定 ②BMP 画像 A 送信　SendImageData() ③用紙サイズ指定　SetMediaSize()　例：8x10（裏面）の場合　引数 331 を指定 ④空白 BMP 画像送信　SendImageData()	
	短辺綴じ		⇒		

◆ 両面印画を行う場合（両面カット紙、2 面付。片面印画、もう一方にオーバーコート処理を行う場合も同様。）

用紙種 出力サイズ	両面指定	画像生成、 画面割付と送信順		プリンターの動作と画像送信手順	用紙出力イメージ
両面 カット紙 多面割付	長辺綴じ		⇒	画像送信手順 ①用紙サイズ指定　SetMediaSize()　例：8x5x2（表面）　引数 238 を指定 ②BMP 画像 A 送信　SendImageData() ③BMP 画像 C 送信　SendImageData() ④用紙サイズ指定　SetMediaSize()　例：8x5x2（裏面）　引数 338 を指定 ⑤BMP 画像 B 送信　SendImageData() ⑥BMP 画像 D 送信　SendImageData() （片面印画、もう一方にオーバーコート処理を行う場合の例） ※用紙の片面に印画、もう一方にオーバーコート処理のみ行いたい場合、表、裏面どちらかに空白 BMP 画像を送信して下さい。 ①用紙サイズ指定　SetMediaSize()　例：8x5x2（表面）　引数 238 を指定 ②BMP 画像 A 送信　SendImageData() ③BMP 画像 C 送信　SendImageData() ④用紙サイズ指定　SetMediaSize()　例：8x5x2（表面）　引数 338 を指定 ⑤空白 BMP 画像送信　SendImageData() ⑥空白 BMP 画像送信　SendImageData()	
	短辺綴じ		⇒		

◆ 片面印画を行う場合（カット紙、反転ユニットを使用）

用紙種 出力サイズ	両面指定	画像生成、 画面割付と送信順		プリンターの動作と画像送信手順	用紙出力イメージ
カット紙 1 面割付	なし	① ② 表 A	⇒	画像送信手順 ①用紙サイズ指定　SetMediaSize()　例：8x10 の場合　引数 131 を指定 ②BMP 画像 A 送信　SendImageData() ①待機　②給紙　③印画　④排出/カット	A
カット紙 多面割付	なし	① ② 画像送信順 表 A 8x5 表 B 8x5	⇒	画像送信手順 ①用紙サイズ指定　SetMediaSize()　例：8x5x2　引数 138 を指定 ②BMP 画像 A 送信　SendImageData() ①待機　②給紙　③印画　④排出/カット 5inch 2 カット	B A

◆ 片面印画を行う場合（ロール紙、反転ユニットは未使用）

用紙種 出力サイズ	両面指定	画像生成、 画面割付と送信順		プリンターの動作と画像送信手順	用紙出力イメージ
ロール紙 1 面割付	なし	① ② 表 A	⇒	画像送信手順 ①用紙サイズ指定　SetMediaSize()　例：8x10 の場合　引数 31 を指定 ②BMP 画像 A 送信　SendImageData() ①待機　②印画　③排出/カット	A
ロール紙 多面割付	なし	① ② 画像送信順 表 A 8x6 表 B 8x6	⇒	画像送信手順 ①用紙サイズ指定　SetMediaSize()　例：8x6x2　引数 39 を指定 ②BMP 画像 A 送信　SendImageData() ①待機　②印画　③排出/カット 6inch 2 カット	B A

7. リボン巻戻し機能（DP-DS620, DP-DS820）

・ホストは、リボンパネルのハーフサイズ換算メディア残り発行枚数取得コマンド（新規追加コマンド）による残り発行枚数情報から、リボンのパネル状態を確認する事が出来ます。
対応メディアは DP-DS620⇒5x7*1、6x8、6x9*2、DP-DS820⇒8x10、8x12、A4 です。

・プリンターから取得したハーフサイズ換算メディア残り発行枚数が偶数の場合はリボンのパネルは未使用状態で、奇数の場合はリボンのパネルは半分使用された状態です。

ハーフサイズ換算メディア残り発行枚数	リボンのパネル状態	
偶数	未使用状態	
奇数	半分使用状態	

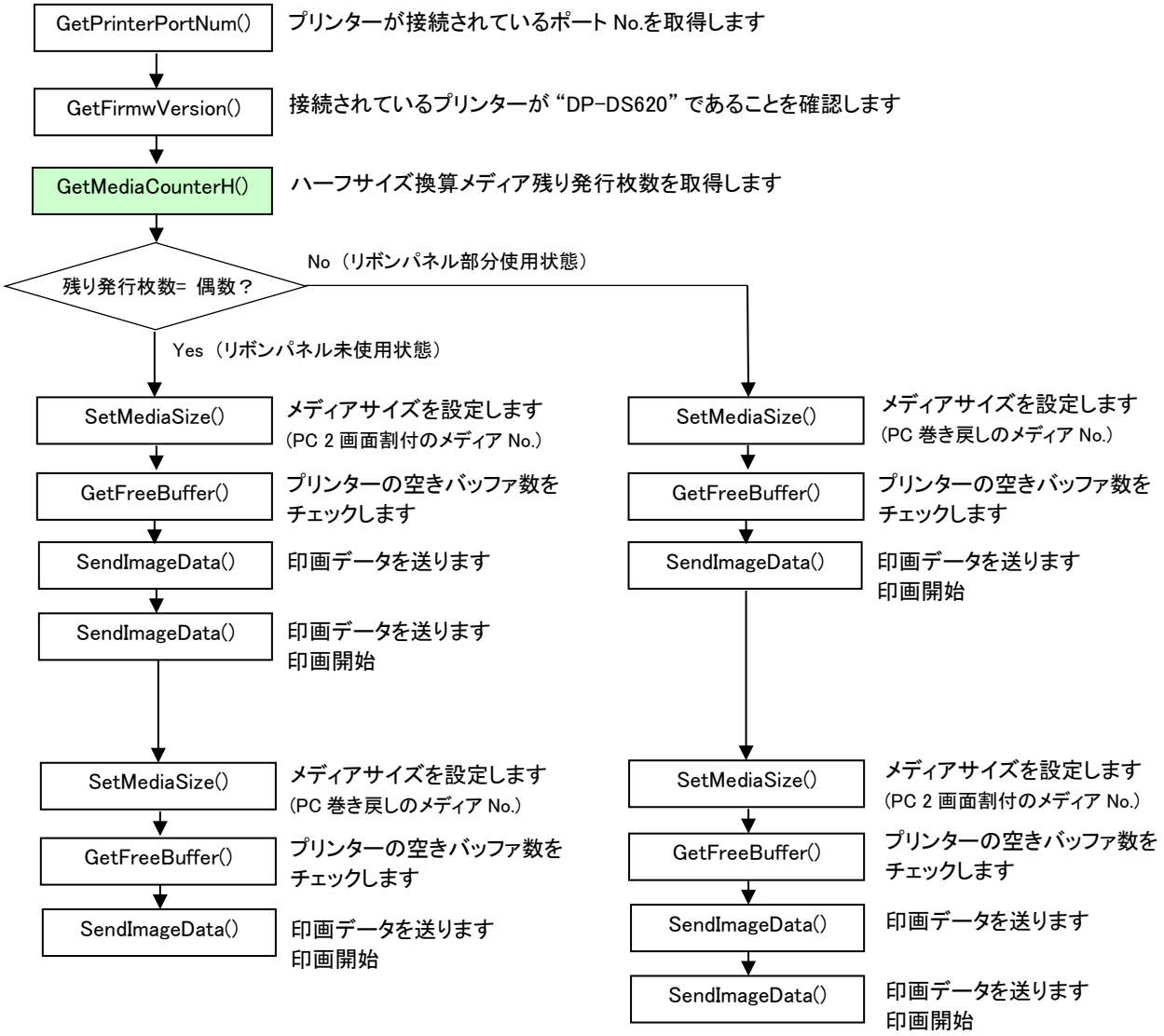
※DP-DS620（A)は「PC 巻き戻し（CSP_PC_REWIND)」を選択した場合、リボンの巻き戻しを伴わない印刷を行います。
「(L 巻き戻し) CSP_L_REWIND*1」「(6x4.5 巻き戻し) CSP_6X4P5_REWIND*2」を選択した場合も同様です。
また DP-DS820（A）で「巻き戻し」を選択した場合も同様です。

注記
※1 ファームウェアバージョン 0.31 以降
※2 ファームウェアバージョン 1.10 以降

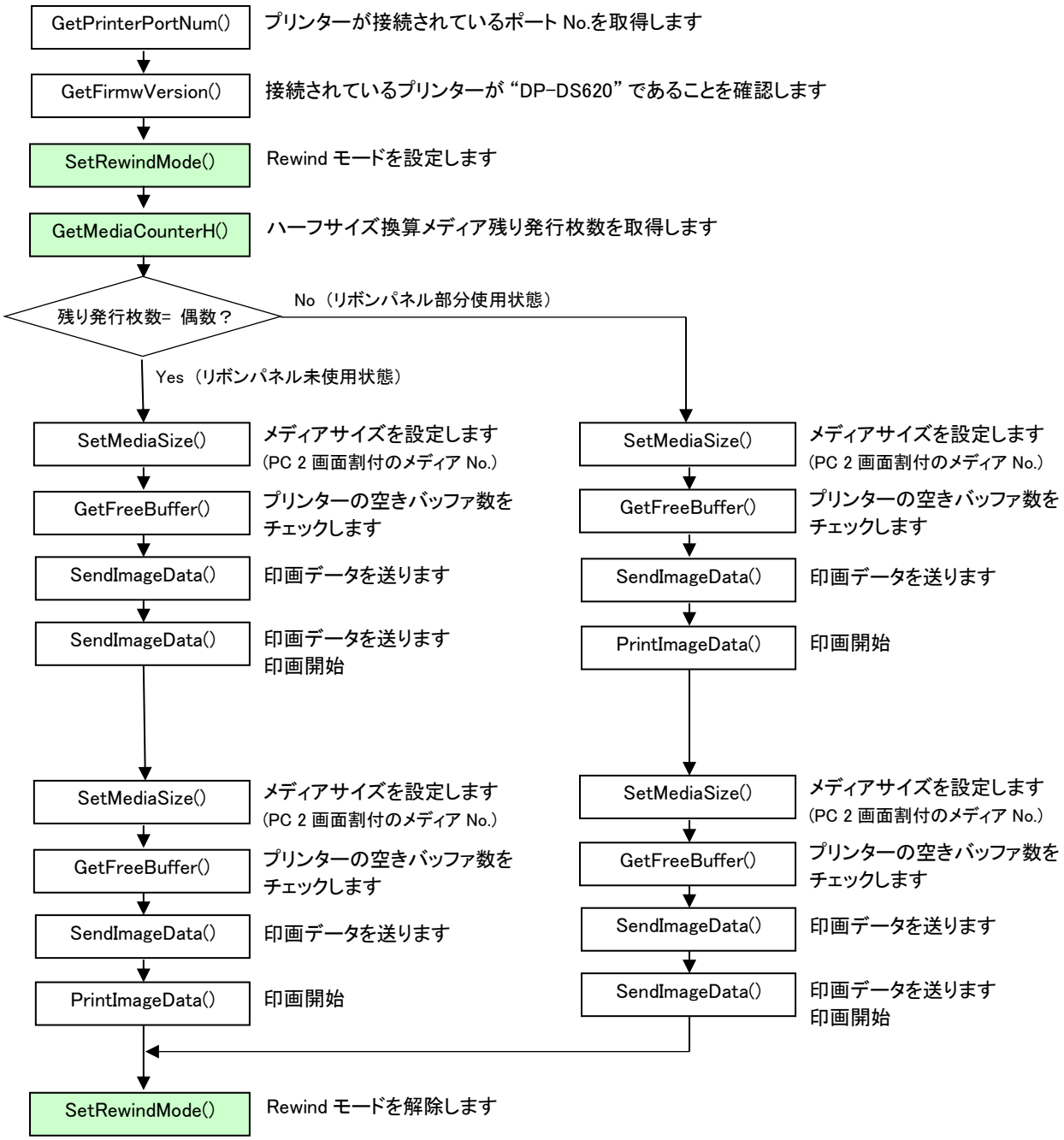
・リボンの巻戻し動作を利用して印画する例を以下に示します。

リボン巻戻し動作処理例

1） メディアサイズへ “PC 巻き戻し（CSP_PC_REWIND)” 及び “PC 2 画面割付（CSP_PCx2)” を指定し、6 x 4 サイズを 3 枚印画する場合の処理例を示します。



２）　メディアサイズへ　“PC 2 画面割付（CSP_PCx2)”　を指定し、“Rewind モード”　にて 6ｘ4 サイズを 3 枚印画する場合の処理例を示します。



8. スタンバイモード（DP-DS620, DP-DS820）

・プリンターは、「Idle」状態が 10 分間 （スタンバイ移行時間設定コマンドで時間を変更している場合はその時間） 続くと、スタンバイモードに移行します。

・スタンバイモード中のプリンターは、プリンターステータスが「STATUS_USUALLY_STANDBY_MODE」となり、緑 LED が点灯から点滅になります。

・スタンバイモード中でも下記 API は使用することが出来ます。

＜スタンバイモード使用可能 API＞

機能	API
プリンターステータスの取得	CvGetStatus() / GetStatus()
プリンタシリアル No の取得	CvGetSerialNo() / GetSerialNo()
メディア初期枚数の取得	GetInitialMediaCount()
メディア残り発行枚数の取得	CvGetMediaCounter() / GetMediaCounter()
ハーフサイズ換算　メディア残り発行枚数の取得	GetMediaCounterH()
メディアタグ情報の取得	CvGetMediaLotNo() / GetMediaLotNo()
メディアコードの取得	CvGetMedia() / GetMedia()
バージョン情報の取得	CvGetVersion() /GetFirmwVersion() *1
メディア種別の取得	GetRfidMediaClass() *2

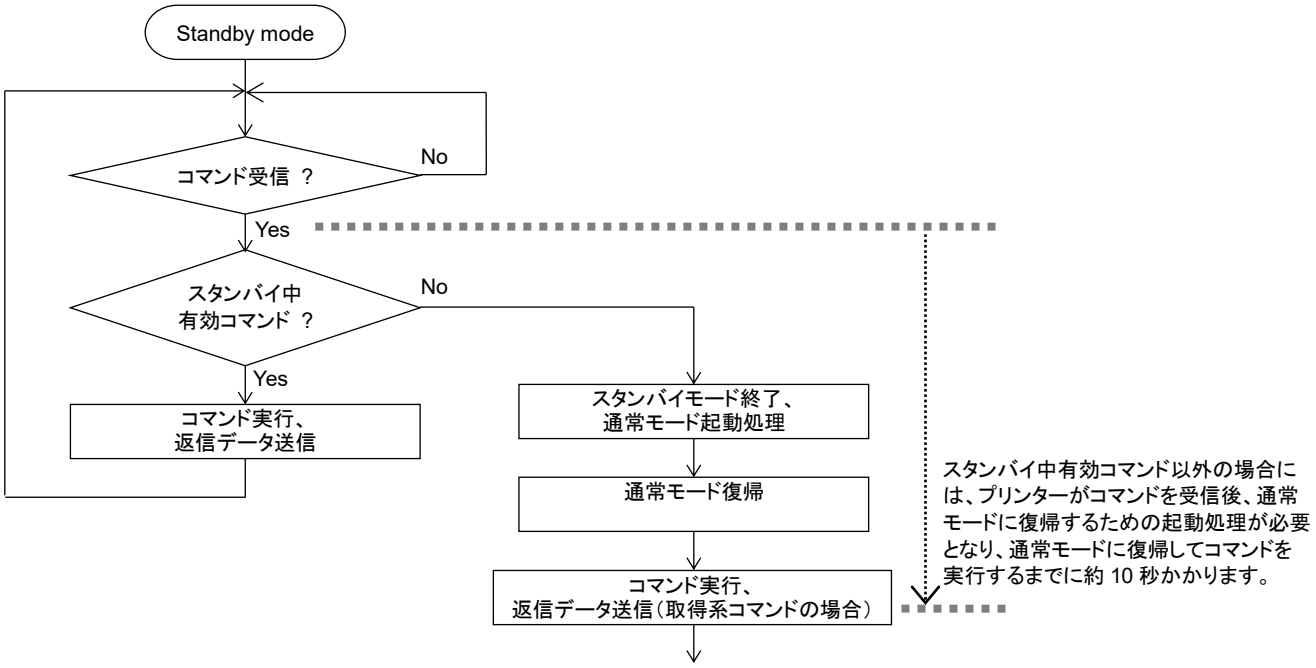
*1:DP-DS620 ファームウェアバージョン 0.31 以降

*2:DP-DS620 ファームウェアバージョン 3.00 未満を除く

・スタンバイモード中に上記以外の API を実行した場合、またはカバーオープンした場合には、スタンバイモードから復帰します。

・スタンバイモード中に上記以外の API を実行した場合は、通常モードに復帰してからその処理を実行します。この場合、プリンターは通常モード復帰のための起動処理が必要で、処理の実行までに約 10 秒かかります。（下図参照）

スタンバイ中にコマンドを受信した場合のプリンター動作



プリンターからの返信データがある取得系コマンドでは、ホストがコマンドを発行してから約 10 秒後に返信データがプリンターから送信されます。従って、ホストはコマンド発行後、プリンターから返信データが返るまでの間のウェイトが必要です。

9. CSP Status API 関数一覧

各 API 関数は接続された機種及びファームウェア対応バージョンにより動作が異なります。CSP Status API Ver.0.6.3.0 以降は、接続されている機種、バージョン確認を DLL が行うフィルター機能を有します。フィルター機能を使用すると、各 API 関数がプリンター未対応の場合 False 又は(-1)を返し、プリンターへコマンドを発行しません。フィルター機能は P28 SetPrinterFilter()関数を使用すると有効に出来ます。

フィルター機能が無効のまま、アプリがプリンター未対応の API 関数を使用した場合の動作は保証されません。フィルター機能を無効とする場合、アプリがプリンター未対応の API 関数を使わないよう、十分注意して下さい。

9-1. CSP Status API 関数一覧 for CV、CW、CWO2

○：ファームウェア対応機能　×：ファームウェア未対応機能（False or (-1)）

数値：対応ファームウェアバージョン（これ以降で動作）

API 機能	Direct Dll(CSP Stat.Dll)関数名	注記	スタンドアロン		
			CV	CW	CWO2
プリンター接続番号・接続数の取得	GetPrinterPortNum();	-	○	○	○
メディアサイズ設定	SetMediaSize() ;	-	○	○	○
イメージデータ転送	SendImageData() ;	-	○	○	○
解像度の設定	SetResolution() ;	-	×	×	○
印画枚数指定	SetPQTY() ;	-	○	○	○
プリント開始	PrintImageData() ;	-	○	○	○
ページレイアウトの開始	StartPageLayout() ;	-	○	○	○
ページレイアウトの終了	EndPageLayout() ;	-	○	○	○
USB 通信タイムアウト時間設定	SetUSBTimeout() ;	-	○	○	○
ミニラボ タワーエラーステータスの取得	GetMiniLabTowerStatis();	-	×	×	×
ミニラボ印画データグループ No. シリアル No. 初期値設定	SetMiniLabGroupNoIni() ;	-	×	×	×
ミニラボ印画データグループ No. 設定	SetMiniLabGroupNo();	-	×	×	×
ミニラボ印画データシリアル No. 設定	SetMiniLabSerialNo();	-	×	×	×
ミニラボ裏面印刷データ設定	SetMiniLabBackprintData();	-	×	×	×
印画中止コマンド	StopPrint();	-	×	×	×
用紙排出先指定	SetMiniLabOutputDest();	-	×	×	×
ソータ用紙 MAX 位置センサー設定	SetMiniLabSortSensor();	-	×	×	×
ソータ用紙 MAX 位置センサー取得	GetMiniLabSortSensor();	-	×	×	×
タワーグループ No 取得	GetMiniLabGroupNo();	-	×	×	×
タワーシリアル No 取得	GetMiniLabSerialNo();	-	×	×	×
プリンター位置の取得	GetMiniLabPrinterAdd();	-	×	×	×

API 機能	Direct Dll(CSP Stat.Dll)関数名		注記	スタンドアロン		
				CV	CW	CWO2
バージョン情報の取得	CvGetVersion();	GetFirmwVersion();	-	○	○	○
センサー情報の取得	CvGetSensorInfo();	GetSensorInfo();	-	○	○	○
解像度の取得	CvGetResolutionH(); CvGetResolutionV();	GetResolutionH(); GetResolutionV();	-	○	○	○
メディアコードの取得	CvGetMedia();	GetMedia();	-	○	○	○
プリンターステータスの取得	CvGetStatus();	GetStatus();	-	○	○	○
カウンタ値の取得	CvGetCounterL(); CvGetCounterA(); CvGetCounterB();	GetCounterL(); GetCounterA(); GetCounterB();	-	○	○	○
		GetCounterP();	-	×	×	○
		GetCounterMatte() GetCounterM()	-	×	×	○
カウンタ値（P）の設定	SetCounterP();		-	×	×	○
カウンタ値のクリア	CvSetClearCounterA(); CvSetClearCounterB();	SetClearCounterA(); SetClearCounterB();	-	○	○	○
カウンタ値のクリア（Matte）	SetClearCounterM()		-	×	×	○
空きバッファ数の取得	CvGetFreeBuffer();	GetFreeBuffer();	-	○	○	○

API 機能	Direct Dll(CSP Stat.Dll)関数名		注記	スタンドアロン		
				CV	CW	CWO2
残り発行枚数の取得	CvGetPQTY();	GetPQTY();	-	○	○	○
メディア残り発行枚数の取得	CvGetMediaCounter();	GetMediaCounter();	-	○	○	○
ハーフサイズ換算メディア残り発行枚数の取得		GetMediaCounterH();	-	×	×	×
メディアロット間色バラツキ補正値の取得	CvGetMediaColorOffset();	GetMediaColorOffset();	-	○	○	○
メディアタグ情報の取得	CvGetMediaLotNo();	GetMediaLotNo();	-	○	○	○
プリンターシリアルナンバーの取得	CvGetSerialNo();	GetSerialNo();	-	○	○	○
ファームウェア書き換えモードへの切り替えコマンド	CvSetFirmwUpdateMode();	SetFirmwUpdateMode();	-	○	○	○
ファームウェア書き換えデータの送信コマンド	CvSetFirmwDataWrite();	SetFirmwDataWrite();	-	○	○	○
色制御データクリアコマンド	CvSetColorDataClear();	SetColorDataClear();	-	○	○	○
色制御データの送信コマンド	CvSetColorDataWrite();	SetColorDataWrite();	-	○	○	○
色制御データバージョン設定コマンド	CvSetColorDataVersion();	SetColorDataVersion();	-	○	○	○
色制御データをテンポラリーへ送信コマンド	SetColorDataWriteTmp();		-	×	○	×
色制御データテンポラリーバージョン設定コマンド	SetColorDataVersionTmp();		-	×	○	×
色制御データバージョン取得コマンド	CvGetColorDataVersion();	GetColorDataVersion();	-	○	○	○
色制御データバージョン取得コマンド<種別指定>	GetColorDataVersionRes();		-	×	×	×
色制御データバージョン取得コマンド<種別、メディア指定>	GetColorDataVersionResEX();		-	×	×	×
色制御データチェックサム取得コマンド	CvGetColorDataChecksum();	GetColorDataChecksum();	-	○	○	○
色制御データチェックサム取得コマンド<種別指定>	GetColorDataChecksumRes();		-	×	×	×
色制御データチェックサム取得コマンド<種別、メディア指定>	GetColorDataChecksumResEX();		-	×	×	×
カッター制御コマンド	SetCutterMode();		-	○	○	○
メディア ID 設定値取得コマンド	GetMediaIdSetInfo();		-	×	○	○
一枚画像保持印画　制御コマンド	SetRetentionMode();		-	×	×	×
オーバーコート仕上げ	SetOvercoatFinish();		-	×	×	○
印画リトライ機能	SetRetryControl();		-	×	×	○
メディア種別の取得	GetRfidMediaClass();		-	×	×	×
RF-ID 予備データの取得	GetRfidReserveData();		-	×	×	×
メディア初期枚数の取得	GetInitilaMediaCount();		-	×	×	×
メディアオフセット枚数の取得	GetMediaCountOffset();		-	×	×	×
反転ユニットファームウェアバージョン情報の取得	GetVersionDuplex();		-	×	×	×
フルカッターセットアップ	SetFullCutterSetUp();		-	×	×	×
フルカッターセットアップ拡張設定	SetFullCutterSetUpEX();		-	×	×	×
スタンバイモード移行時間設定	SetStandbyTime();		-	×	×	×
スタンバイモード移行時間設定値の取得	GetStandbyTime();		-	×	×	×
メディアエンド保持モード設定	SetEndKeepMode ();		-	×	×	×
メディアエンド保持モード設定値の取得	GetEndKeepMode ();		-	×	×	×
USB iSerialNumber 有効／無効設定	SetUSBSerialEnable ();		-	×	×	×
USB iSerialNumber 設定値の取得	GetUSBSerialEnable ();		-	×	×	×
Rewind モード設定	SetRewindMode ();		-	×	×	×
Rewind モードの取得	GetRewindMode ();		-	×	×	×
パノラマ印画開始可能チェック	GetPanoramaPrintable();		-	×	×	×
連続パノラマ印画設定	SetContPanorama();		-	×	×	×
プリント速度指定	SetPrintSpeed();		-	×	×	×
ガンマテーブルの設定	SetGammaTable();		-	×	×	×
ガンマテーブルチェックサムの取得	GetGammaTableChecksum();		-	×	×	×
データテーブルのクリア	SetDataTableClear();		-	×	×	×
反転ユニットファームウェア書き換えモードへの切り替えコマンド	SetDunitFirmwUpdateMode();		-	×	×	×
反転ユニットファームウェア書き換えデータの送信コマンド	SetDunitFirmwDataWrite();		-	×	×	×
反転ユニットファームウェア書換え状態の取得	GetDunitUpdateStatus();		-	×	×	×
反転ユニットエラーログ作成要求	DunitErrorLogCreateReq();		-	×	×	×
反転ユニットエラーログデータ長の取得	GetDunitErrorLogDataLength();		-	×	×	×
反転ユニットエラーログデータ受信	GetDunitErrorLogData();		-	×	×	×

API 機能	Direct Dll(CSP Stat.Dll)関数名		注記	スタンドアロン		
				CV	CW	CW02
反転ユニットエラーログ作成状態の取得	GetDunitErrorLogStatus();		-	×	×	×
デカール制御設定	SetDecurlCtrl()		-	×	×	×
現在時刻通知	SetSysTime()		-	×	×	×
対応メディア情報の取得	GetSupportedMediaInfo();		-	×	×	×
汎用設定系コマンド	CvSetCommand();	SetCommand();	-	○	○	○
汎用取得系コマンド	CvGetCommandEX();	GetCommandEX();	-	○	○	○

9-2. CSP Status API 関数一覧 for DS40、DS80、DS40T、DS80T

○：ファームウェア対応機能　×：ファームウェア未対応機能（False or (-1)）

数値：対応ファームウェアバージョン（これ以降で動作）

API 機能	Direct Dll(CSP Stat.Dll)関数名	注記	スタンドアロン		タワー用	
			DS40	DS80	DS40T	DS80T
プリンター接続番号・接続数の取得	GetPrinterPortNum();	-	○	○	○	○
メディアサイズ設定	SetMediaSize() ;	-	○	○	○	○
イメージデータ転送	SendImageData() ;	-	○	○	○	○
解像度の設定	SetResolution() ;	-	○	○	○	○
印画枚数指定	SetPQTY() ;	-	○	○	○	○
プリント開始	PrintImageData() ;	-	○	○	○	○
ページレイアウトの開始	StartPageLayout() ;	-	○	○	○	○
ページレイアウトの終了	EndPageLayout() ;	-	○	○	○	○
USB 通信タイムアウト時間設定	SetUSBTimeout() ;	-	○	○	○	○
ミニラボ タワーエラーステータスの取得	GetMiniLabTowerStatis();	-	×	×	○	○
ミニラボ印画データグループ No. シリアル No. 初期値設定	SetMiniLabGroupNoIni() ;	-	×	×	○	○
ミニラボ印画データグループ No. 設定	SetMiniLabGroupNo();	-	×	×	○	○
ミニラボ印画データシリアル No. 設定	SetMiniLabSerialNo();	-	×	×	○	○
ミニラボ裏面印刷データ設定	SetMiniLabBackprintData();	-	×	×	○	○
印画中止コマンド	StopPrint();	-	×	×	○	○
用紙排出先指定	SetMiniLabOutputDest();	-	×	×	○	○
ソータ用紙 MAX 位置センサー設定	SetMiniLabSortSensor();	-	×	×	○	○
ソータ用紙 MAX 位置センサー取得	GetMiniLabSortSensor();	-	×	×	○	○
タワーグループ No 取得	GetMiniLabGroupNo();	-	×	×	○	○
タワーシリアル No 取得	GetMiniLabSerialNo();	-	×	×	○	○
プリンター位置の取得	GetMiniLabPrinterAdd();	-	×	×	○	○

API 機能	Direct Dll(CSP Stat.Dll)関数名		注記	スタンドアロン		タワー用	
				DS40	DS80	DS40T	DS80T
バージョン情報の取得	CvGetVersion();	GetFirmwVersion();	-	○	○	○	○
センサー情報の取得	CvGetSensorInfo();	GetSensorInfo();	-	○	○	○	○
解像度の取得	CvGetResolutionH(); CvGetResolutionV();	GetResolutionH(); GetResolutionV();	-	○	○	○	○
メディアコードの取得	CvGetMedia();	GetMedia();	-	○	○	○	○
プリンターステータスの取得	CvGetStatus();	GetStatus();	-	○	○	○	○
カウンタ値の取得	CvGetCounterL(); CvGetCounterA(); CvGetCounterB();	GetCounterL(); GetCounterA(); GetCounterB();	-	○	○	○	○
		GetCounterP();	-	1.04	1.02	0.21	0.21
		GetCounterMatte() GetCounterM()	-	1.30	1.30	0.31	0.31

API 機能	Direct Dll(CSP Stat.Dll)関数名		注記	スタンドアロン		タワー用	
				DS40	DS80	DS40T	DS80T
カウンタ値（P）の設定	SetCounterP();		-	1.04	1.02	0.21	0.21
カウンタ値のクリア	CvSetClearCounterA(); CvSetClearCounterB();	SetClearCounterA(); SetClearCounterB();	-	○	○	○	○
カウンタ値のクリア（Matte）	SetClearCounterM()		-	1.30	1.30	0.31	0.31
空きバッファ数の取得	CvGetFreeBuffer();	GetFreeBuffer();	-	○	○	○	○
残り発行枚数の取得	CvGetPQTY();	GetPQTY();	-	○	○	○	○
メディア残り発行枚数の取得	CvGetMediaCounter();	GetMediaCounter();	-	○	○	○	○
ハーフサイズ換算メディア残り発行枚数の取得		GetMediaCounterH();	-	×	×	×	×
メディアロット間色バラツキ補正値の取得	CvGetMediaColorOffset();	GetMediaColorOffset();	-	○	○	○	○
メディアタグ情報の取得	CvGetMediaLotNo();	GetMediaLotNo();	-	○	○	○	○
プリンターシリアルナンバーの取得	CvGetSerialNo();	GetSerialNo();	-	○	○	○	○
ファームウェア書き換えモードへの切り替えコマンド	CvSetFirmwUpdateMode();	SetFirmwUpdateMode();	-	○	○	○	○
ファームウェア書き換えデータの送信コマンド	CvSetFirmwDataWrite();	SetFirmwDataWrite();	-	○	○	○	○
色制御データクリアコマンド	CvSetColorDataClear();	SetColorDataClear();	-	○	○	○	○
色制御データの送信コマンド	CvSetColorDataWrite();	SetColorDataWrite();	-	○	○	○	○
色制御データバージョン設定コマンド	CvSetColorDataVersion();	SetColorDataVersion();	-	○	○	○	○
色制御データをテンポラリーへ送信コマンド	SetColorDataWriteTmp();		-	×	×	×	×
色制御データテンポラリーバージョン設定コマンド	SetColorDataVersionTmp();		-	×	×	×	×
色制御データバージョン取得コマンド	CvGetColorDataVersion();	GetColorDataVersion();	-	○	○	○	○
色制御データバージョン取得コマンド<種別指定>	GetColorDataVersionRes();		-	×	×	×	×
色制御データバージョン取得コマンド<種別、メディア指定>	GetColorDataVersionResEX();		-	×	×	×	×
色制御データチェックサム取得コマンド	CvGetColorDataChecksum();	GetColorDataChecksum();	-	○	○	○	○
色制御データチェックサム取得コマンド<種別指定>	GetColorDataChecksumRes();		-	×	×	×	×
色制御データチェックサム取得コマンド<種別、メディア指定>	GetColorDataChecksumResEX();		-	×	×	×	×
カッター制御コマンド	SetCutterMode();		-	○	○	○	○
メディア ID 設定値取得コマンド	GetMediaIdSetInfo();		-	○	○	○	○
一枚画像保持印画　制御コマンド	SetRetentionMode();		-	○	×	○	×
オーバーコート仕上げ	SetOvercoatFinish();		-	1.30	1.30	0.31	0.31
印画リトライ機能	SetRetryControl();		-	1.30	1.30	×	×
メディア種別の取得	GetRfidMediaClass();		-	×	×	×	×
RF-ID 予備データの取得	GetRfidReserveData();		-	×	×	×	×
メディア初期枚数の取得	GetInitilaMediaCount();		-	2.00	×	×	×
メディアオフセット枚数の取得	GetMediaCountOffset();		-	2.00	×	×	×
反転ユニットファームウェアバージョン情報の取得	GetVersionDuplex();		-	×	×	×	×
フルカッターセットアップ	SetFullCutterSetUp();		-	1.60	×	×	×
フルカッターセットアップ拡張設定	SetFullCutterSetUpEX();		-	1.60	×	×	×
スタンバイモード移行時間設定	SetStandbyTime();		-	×	×	×	×
スタンバイモード移行時間設定値の取得	GetStandbyTime();		-	×	×	×	×
メディアエンド保持モード設定	SetEndKeepMode ();		-	×	×	×	×
メディアエンド保持モード設定値の取得	GetEndKeepMode ();		-	×	×	×	×
USB iSerialNumber 有効／無効設定	SetUSBSerialEnable ();		-	2.00	×	×	×
USB iSerialNumber 設定値の取得	GetUSBSerialEnable ();		-	2.00	×	×	×
Rewind モード設定	SetRewindMode ();		-	×	×	×	×
Rewind モードの取得	GetRewindMode ();		-	×	×	×	×
パノラマ印画開始可能チェック	GetPanoramaPrintable();		-	×	×	×	×
連続パノラマ印画設定	SetContPanorama();		-	×	×	×	×
プリント速度指定	SetPrintSpeed();		-	×	×	×	×
ガンマテーブルの設定	SetGammaTable();		-	×	×	×	×
ガンマテーブルチェックサムの取得	GetGammaTableChecksum();		-	×	×	×	×
データテーブルのクリア	SetDataTableClear();		-	×	×	×	×
反転ユニットファームウェア書き換えモードへの切り替えコマンド	SetDunitFirmwUpdateMode();		-	×	×	×	×
反転ユニットファームウェア書き換えデータの送信コマンド	SetDunitFirmwDataWrite();		-	×	×	×	×

API 機能	Direct Dll(CSP Stat.Dll)関数名		注記	スタンドアロン		タワー用	
				DS40	DS80	DS40T	DS80T
反転ユニットファームウェア書換え状態の取得	GetDunitUpdateStatus();		-	×	×	×	×
反転ユニットエラーログ作成要求	DunitErrorLogCreateReq();		-	×	×	×	×
反転ユニットエラーログデータ長の取得	GetDunitErrorLogDataLength();		-	×	×	×	×
反転ユニットエラーログデータ受信	GetDunitErrorLogData();		-	×	×	×	×
反転ユニットエラーログ作成状態の取得	GetDunitErrorLogStatus();		-	×	×	×	×
デカール制御設定	SetDecurlCtrl()		-	×	×	×	×
現在時刻通知	SetSysTime()		-	×	×	×	×
対応メディア情報の取得	GetSupportedMediaInfo();		-	×	×	×	×
汎用設定系コマンド	CvSetCommand();	SetCommand();	-	○	○	○	○
汎用取得系コマンド	CvGetCommandEX();	GetCommandEX();	-	○	○	○	○

9-3. CSP Status API 関数一覧 for DS-RX1、DP-TC10、DP-DS80D、DP-DS620、DP-DS820、DP-QW410

○：ファームウェア対応機能　×：ファームウェア未対応機能（False or (-1)）　数値：対応ファームウェアバージョン（これ以降で動作）

API 機能	Direct Dll(CSP Stat.Dll)関数名	注記	スタンドアロン					
			DS-RX1	DP-TC10	両面 DP-DS80D	DP-DS620	DP-DS820	DP-QW410
プリンター接続番号・接続数の取得	GetPrinterPortNum();	-	○	○	○	○	○	○
メディアサイズ設定	SetMediaSize() ;	-	○	○	○	○	○	○
イメージデータ転送	SendImageData() ;	*3	○	○	○	○	○	○
解像度の設定	SetResolution() ;	-	○	○	○	○	○	○
印画枚数指定	SetPQTY() ;	*3	○	○	○	○	○	○
プリント開始	PrintImageData() ;	*3	○	○	○	○	○	○
ページレイアウトの開始	StartPageLayout() ;	-	○	○	○	○	○	○
ページレイアウトの終了	EndPageLayout() ;	-	○	○	○	○	○	○
USB 通信タイムアウト時間設定	SetUSBTimeout() ;	-	○	○	○	○	○	○
ミニラボ タワーエラーステータスの取得	GetMiniLabTowerStatis();	-	×	×	×	×	×	×
ミニラボ印画データグループ No. シリアル No. 初期値設定	SetMiniLabGroupNoIni() ;	-	×	×	×	×	×	×
ミニラボ印画データグループ No. 設定	SetMiniLabGroupNo();	-	×	×	×	×	×	×
ミニラボ印画データシリアル No. 設定	SetMiniLabSerialNo();	-	×	×	×	×	×	×
ミニラボ裏面印刷データ設定	SetMiniLabBackprintData();	-	×	×	×	×	×	×
印画中止コマンド	StopPrint();	-	×	×	×	×	×	×
用紙排出先指定	SetMiniLabOutputDest();	-	×	×	×	×	×	×
ソータ用紙 MAX 位置センサー設定	SetMiniLabSortSensor();	-	×	×	×	×	×	×
ソータ用紙 MAX 位置センサー取得	GetMiniLabSortSensor();	-	×	×	×	×	×	×
タワーグループ No 取得	GetMiniLabGroupNo();	-	×	×	×	×	×	×
タワーシリアル No 取得	GetMiniLabSerialNo();	-	×	×	×	×	×	×
プリンター位置の取得	GetMiniLabPrinterAdd();	-	×	×	×	×	×	×

API 機能	Direct Dll(CSP Stat.Dll)関数名		注記	スタンドアロン					
				DS-RX1	DP-TC10	両面 DP-DS80D	DP-DS620	DP-DS820	DP-QW410
バージョン情報の取得	CvGetVersion(); CvGetSensorInfo();	GetFirmwVersion();	*1	○	○	○	○	○	○
センサー情報の取得		GetSensorInfo();	*2	○	○	○	○	○	○
解像度の取得	CvGetResolutionH(); CvGetResolutionV();	GetResolutionH(); GetResolutionV();	*2	○	○	○	○	○	○
メディアコードの取得	CvGetMedia();	GetMedia();	*1	○	○	○	○	○	○
プリンターステータスの取得	CvGetStatus();	GetStatus();	*1	○	○	○	○	○	○
カウンタ値の取得	CvGetCounterL(); CvGetCounterA(); CvGetCounterB();	GetCounterL(); GetCounterA(); GetCounterB();	*2	○	○	○	○	○	○
		GetCounterP();	*2	○	○	○	○	○	○
		GetCounterMatte(); GetCounterM();	*2	○	○	○	○	○	○
拡張ライフカウンタ値の取得	GetCounterLifeEX();		*3	×	×	×	4.00	×	×
拡張マットカウンタ値の取得	GetCounterMatteEX();		*3	×	×	×	4.00	×	×
カウンタ値（P）の設定	SetCounterP();		*3	○	○	○	○	○	○
カウンタ値のクリア	CvSetClearCounterA(); CvSetClearCounterB();	SetClearCounterA(); SetClearCounterB();	*3	○	○	○	○	○	○
カウンタ値のクリア（Matte）	SetClearCounterM();		*3	○	○	○	○	○	○
空きバッファ数の取得	CvGetFreeBuffer();	GetFreeBuffer();	*2	○	○	○	○	○	○
残り発行枚数の取得	CvGetPQTY();	GetPQTY();	*2	○	○	○	○	○	○
メディア残り発行枚数の取得	CvGetMediaCounter();	GetMediaCounter();	*1	○	○	○	○	○	○
ハーフサイズ換算メディア残り発行枚数の取得		GetMediaCounterH();	*1	×	×	×	○	○	×
メディアロット間色バラツキ補正値の取得	CvGetMediaColorOffset();	GetMediaColorOffset();	*2	○	○	○	○	○	○
メディアタグ情報の取得	CvGetMediaLotNo();	GetMediaLotNo();	*1	○	○	○	○	○	○
プリンターシリアルナンバーの取得	CvGetSerialNo();	GetSerialNo();	*1	○	○	○	○	○	○
ファームウェア書き換えモードへの切り替えコマンド	CvSetFirmwUpdateMode();	SetFirmwUpdateMode();	*3	○	○	○	○	○	○
ファームウェア書き換えデータの送信コマンド	CvSetFirmwDataWrite();	SetFirmwDataWrite();	-	○	○	○	○	○	○
色制御データクリアコマンド	CvSetColorDataClear();	SetColorDataClear();	*3	○	○	○	○	○	○
色制御データの送信コマンド	CvSetColorDataWrite();	SetColorDataWrite();	*3	○	○	○	○	○	○
色制御データバージョン設定コマンド	CvSetColorDataVersion();	SetColorDataVersion();	*3	○	○	○	○	○	○
色制御データをテンポラリーへ送信コマンド	SetColorDataWriteTmp();		-	×	×	×	×	×	×
色制御データテンポラリーバージョン設定コマンド	SetColorDataVersionTmp();		-	×	×	×	×	×	×
色制御データバージョン取得コマンド	CvGetColorDataVersion();	GetColorDataVersion();	*2	○	○	○	○	○	○
色制御データバージョン取得コマンド<種別指定>	GetColorDataVersionRes();		*2	×	×	×	○	○	○
色制御データバージョン取得コマンド<種別、メディア指定>	GetColorDataVersionResEX();		*2	×	×	×	3.00	○	○
色制御データチェックサム取得コマンド	CvGetColorDataChecksum();	GetColorDataChecksum();	*2	○	○	○	○	○	○
色制御データチェックサム取得コマンド<種別指定>	GetColorDataChecksumRes();		*2	×	×	×	○	○	○

API 機能	Direct Dll(CSP Stat.Dll)関数名	注記	スタンドアロン					
			DS-RX1	DP-TC10	両面 DP-DS80D	DP-DS620	DP-DS820	DP-QW410
色制御データチェックサム取得コマンド<種別、メディア指定>	GetColorDataChecksumResEX();	*2	×	×	×	3.00	○	○
カッター制御コマンド	SetCutterMode();	*3	○	○	○※1	○	○	○
メディア ID 設定値取得コマンド	GetMediaIdSetInfo();	*2	○	○	○	○	○	○
一枚画像保持印画　制御コマンド	SetRetentionMode();	-	×	×	×	×	×	×
オーバーコート仕上げ	SetOvercoatFinish();	*3	○	○	○	○	○	○
印画リトライ機能	SetRetryControl();	*3	○	○	○※1	○	○	○
メディア種別の取得	GetRfidMediaClass();	*5	○	○	×	○	○	○
RF-ID 予備データの取得	GetRfidReserveData();	*2	○	○	×	○	○	○
メディア初期枚数の取得	GetInitilaMediaCount();	*1	○	○	×	○	○	○
メディアオフセット枚数の取得	GetMediaCountOffset();	-	2.00	×	×	×	×	×
反転ユニットファームウェアバージョン情報の取得	GetVersionDuplex();	-	×	×	○	×	×	×
フルカッターセットアップ	SetFullCutterSetUp();	*3	×	×	×	○	○	○
フルカッターセットアップ拡張設定	SetFullCutterSetUpEX();	*3	×	×	×	1.20	○	○
スタンバイモード移行時間設定	SetStandbyTime();	*3	×	×	×	○	○	×
スタンバイモード移行時間設定値の取得	GetStandbyTime();	*2	×	×	×	○	○	×
メディアエンド保持モード設定	SetEndKeepMode();	*3	×	×	×	○	○	○
メディアエンド保持モード設定値の取得	GetEndKeepMode();	*2	×	×	×	○	○	○
USB iSerialNumber 有効／無効設定	SetUSBSerialEnable();	*3	2.00	×	×	○	○	○
USB iSerialNumber 設定値の取得	GetUSBSerialEnable();	*2	2.00	×	×	○	○	○
Rewind モード設定	SetRewindMode();	-	×	×	×	○	○	×
Rewind モードの取得	GetRewindMode();	-	×	×	×	○	○	×
パノラマ印画開始可能チェック	GetPanoramaPrintable();	*2	×	×	×	1.20	○	×
連続パノラマ印画設定	SetContPanorama();	*3	×	×	×	1.31	○	×
プリント速度指定	SetPrintSpeed();	*3	×	×	×	×	○	○
ガンマテーブルの設定	SetGammaTable();	*3	×	×	×	×	0.50	×
ガンマテーブルチェックサムの取得	GetGammaTableChecksum();	*2	×	×	×	×	0.50	×
データテーブルのクリア	SetDataTableClear();	*3	×	×	×	×	0.50	×
反転ユニットファームウェア書き換えモードへの 切り替えコマンド	SetDunitFirmwUpdateMode();	-	×	×	1.11	×	×	×
反転ユニットファームウェア書き換えデータの 送信コマンド	SetDunitFirmwDataWrite();	-	×	×	1.11	×	×	×
反転ユニットファームウェア書換え状態の取得	GetDunitUpdateStatus();	-	×	×	1.11	×	×	×
反転ユニットエラーログ作成要求	DunitErrorLogCreateReq();	-	×	×	1.17	×	×	×

API 機能	Direct Dll(CSP Stat.Dll)関数名		注記	スタンドアロン					
				DS-RX1	DP-TC10	両面 DP-DS80D	DP-DS620	DP-DS820	DP-QW410
反転ユニットエラーログデータ長の取得	GetDunitErrorLogDataLength();		-	×	×	1.17	×	×	×
反転ユニットエラーログデータ受信	GetDunitErrorLogData();		-	×	×	1.17	×	×	×
反転ユニットエラーログ作成状態の取得	GetDunitErrorLogStatus();		-	×	×	1.17	×	×	×
デカール制御設定	SetDecurlCtrl();		-	×	×	×	×	×	○
現在時刻通知	SetSysTime();		-	×	×	×	×	×	○
対応メディア情報の取得	GetSupportedMediaInfo();		*3	×	×	×	4.00	×	×
汎用設定系コマンド	CvSetCommand();	SetCommand();	*4	○	○	○	○	○	○
汎用取得系コマンド	CvGetCommandEX();	GetCommandEX();	*4	○	○	○	○	○	○

※1： ロール紙の印刷時のみ対応

注記 （対象モデル= DP-DS620、DP-DS820）

*1: スタンバイ中でもプリンターは返信データの送信が可能です。

*2: スタンバイ中の場合、プリンターは通常状態に復帰してからコマンドを実行し返信データを送信します（約10秒後に送信）。
ホストは、プリンターが返信データを送信するまでのウェイトが必要です。

*3: スタンバイ中の場合、プリンターは通常状態に復帰してからコマンドを実行します（約10秒後に実行）。

*4: スタンバイ中は発行するコマンドの種類によって動作が異なります。

*5: DP-DS820、及び、DP-DS620　ファームウェアバージョン3.00以降は *1と同じ動作。DP-DS620 ファームウェアバージョン3.00未満は *2と同じ動作。

(8. スタンバイモード参照)

10. KIOSK モデルの API 関数動作について

タワーコントローラに KIOSK F/W を書き込み、タワー用 API 関数をコールした場合の動作

	コマンド	CSP Printer Status API 関数	DS40T	DS80T
設定系	印画データグループ No.設定	SetMiniLabGroupNo( )	○	○
	印画データグループ No.、シリアル No.初期値設定	SetMiniLabGroupNoIni( )	○	○
	印画データシリアル No.設定	SetMiniLabSerialNo( )	○	○
	裏面印刷データ 設定	SetMiniLabBackprintData( )	●	●
	用紙排出先指定	SetMiniLabOutputDest( )	●	●
	印画停止指定	StopPrint( )	○	○
	ソータ用紙 MAX 位置検出センサーON/OFF 設定	SetMiniLabSrtSensor( )	●	●
	プリントカウンタ(P)値の設定	SetCounterP( )	○	○
取得系	タワーステータス取得	GetMiniLabTowerStatus ( )	○	○
	グループ No.取得	GetMiniLabGroupNo( )	○	○
	シリアル No.取得	GetMiniLabSerialNo( )	○	○
	プリンタードレスの取得	GetMiniLabPrinterAdd( )	○	○
	ソータ用紙 MAX 位置検出センサーON/OFF 取得	GetMiniLabSrtSensor( )	○	○
	プリントカウンタ(P)値の取得	GetCounterP( )	○	○
動作が異なるコマンド				
取得系	プリンターステータスの取得	GetStatus( )	△	△

○ = 相違なし
● = 設定値は無視される
△ = プリンターが印画物を排出（”shooting” ステータス）後、”backprint” ステータスへの変化なし

11. ファームウェアと制御データアップデート手順について

11-1 書換え手順

お客様が API を使用してファームウェアを書き込む時は、次の手順で行って下さい。書き込み後は書換えが正常に終了した事を、バージョンとチェックサムの確認で行って下さい。

参照頁

API 使用例)

1	SetFirmwUpdateMode()	ファームウェア書換えモードへ移行	P63
2	SetFirmwDataWrite()	ファームウェアデータの書き込み	P64
3	GetFirmwVersion()	ファームウェアのバージョンの確認	P39
4	SetColorDataClear()	制御データクリア	P70
5	SetColorDataVersion()	300dpi の制御データバージョンセット	P71
6	SetColorDataWrite()	300dpi の制御データ書き込み	P70
7	GetColorDataVersion()	300dpi の制御データバージョンの確認	P71
8	GetColorDataChecksum()	300dpi の制御データチェックサムの確認	P75
9	SetColorDataVersion()	600dpi の制御データバージョンセット	P71
10	SetColorDataWrite()	600dpi の制御データ書き込み	P70
11	GetColorDataVersion()	600dpi の制御データバージョンの確認	P71
12	GetColorDataChecksum()	600dpi の制御データチェックサムの確認	P75

■DP-DS620(ファームウェアバージョン 3.00 未満)

参照頁

API 使用例)

1	SetFirmwUpdateMode()	ファームウェア書換えモードへ移行	P63
2	SetFirmwDataWrite()	ファームウェアデータの書き込み	P64
3	GetFirmwVersion()	ファームウェアのバージョンの確認	P39
4	SetColorDataClear()	制御データクリア	P70
5	SetColorDataVersion()	300dpi の制御データバージョンセット	P71
6	SetColorDataWrite()	300dpi の制御データ書き込み	P70
7	GetColorDataVersionRes()	300dpi の制御データバージョンの確認	P72
8	GetColorDataChecksumRes()	300dpi の制御データチェックサムの確認	P75
9	SetColorDataVersion()	600dpi の制御データバージョンセット	P71
10	SetColorDataWrite()	600dpi の制御データ書き込み	P70
11	GetColorDataVersionRes()	600dpi の制御データバージョンの確認	P72
12	GetColorDataChecksumRes()	600dpi の制御データチェックサムの確認	P75
13	SetColorDataVersion()	低速の制御データバージョンセット	P71
14	SetColorDataWrite()	低速の制御データ書き込み	P70
15	GetColorDataVersionRes()	低速の制御データバージョンの確認	P72
16	GetColorDataChecksumRes()	低速の制御データチェックサムの確認	P75

■DP-DS820、DP-QW410、DP-DS620(ファームウェアバージョン 3.00 以降)

DP-DS820、DP-QW410、DP-DS620(ファームウェアバージョン 3.00 以降)は複数メディアを装着できるため、メディア種別ごと色制御データを書き込む必要が有ります。

(DP-QW410 は、解像度 600dpi、および、高濃度に対応していないため、9-12、17-20 の手順を除きます。
DP-DS620(ファームウェアバージョン 3.00 以降)は、高濃度に対応していないため、17-20 の手順を除きます。)

参照頁

1	SetFirmwUpdateMode()	ファームウェア書換えモードへ移行	P63
2	SetFirmwDataWrite()	ファームウェアデータの書き込み	P64
3	GetFirmwVersion()	ファームウェアのバージョンの確認	P39
4	SetColorDataClear()	制御データクリア	P70
メディア種別 1			
5	SetColorDataVersion()	300dpi の制御データバージョンセット	P71
6	SetColorDataWrite()	300dpi の制御データ書き込み	P70
7	GetColorDataVersionResEx()	300dpi の制御データバージョンの確認	P73
8	GetColorDataChecksumResEx()	300dpi の制御データチェックサムの確認	P76
9	SetColorDataVersion()	600dpi の制御データバージョンセット	P71
10	SetColorDataWrite()	600dpi の制御データ書き込み	P70
11	GetColorDataVersionResEX()	600dpi の制御データバージョンの確認	P73
12	GetColorDataChecksumResEX()	600dpi の制御データチェックサムの確認	P76
13	SetColorDataVersion()	低速の制御データバージョンセット	P71
14	SetColorDataWrite()	低速の制御データ書き込み	P70

15	GetColorDataVersionResEX()	低速の制御データバージョンの確認	P73
16	GetColorDataChecksumResEX()	低速の制御データチェックサムの確認	P76
17	SetColorDataVersion()	高濃度の制御データバージョンセット	P71
18	SetColorDataWrite()	高濃度の制御データ書き込み	P70
19	GetColorDataVersionResEX()	高濃度の制御データバージョンの確認	P73
20	GetColorDataChecksumResEX()	高濃度の制御データチェックサムの確認	P76

メディア種別 2
5-20 を繰り返し

11-2 反転ユニットファームウェア書換え手順

お客様が API を使用してファームウェアを書き込む時は、次の手順で行って下さい。書き込み後は書換えが正常に終了した事をバージョンの確認で行って下さい。

参照頁

API 使用例）

1	SetDunitFirmwUpdateMode()	反転ユニットファームウェア書換えモード移行	P65
2	GetDunitUpdateStatus()	反転ユニットファームウェア書換え状態の取得	P66
3	SetDunitFirmwDataWrite()	反転ユニットファームウェアデータの書き込み	P65
4	GetDunitUpdateStatus()	反転ユニットファームウェア書換え状態の取得	P66
5	GetVersionDuplex()	反転ユニット、ファームウェアバージョン情報の取得	P84

11-3 バージョンとチェックサムについて

制御データのバージョンを読み出すと、300dpi の制御データ書込み後は、300dpi のバージョンとチェックサム、600dpi の制御データ書込み後は、600dpi のバージョンとチェックサムが読み出されます。
プリンターの状態と読み出される制御データのバージョンとチェックサムの関係は次のようになります。

プリンターの状態	種別	制御データの バージョン＆チェックサム
制御データ書込み後	300dpi	300dpi
	600dpi	600dpi
電源投入後	－	300dpi
印画後	300dpi	300dpi
	600dpi	600dpi

■DP-DS620、DP-DS820、DP-QW410
プリンターの状態と読み出される制御データのバージョンとチェックサムの関係は次のようになります。

プリンターの状態	種別	制御データの バージョン＆チェックサム	
		種別指定なし	種別指定あり
制御データ書込み後	300dpi	300dpi	300dpi
	600dpi *2	600dpi	600dpi
	低速	低速	低速
	高濃度 *1	高濃度	高濃度
電源投入後 印画後	300dpi	300dpi	300dpi
	600dpi *2	読み出しできません （300dpi のみ）	600dpi
	低速		低速
	高濃度*1		高濃度
	300dpi	300dpi	300dpi
	600dpi *2	600dpi	600dpi
	低速	読み出しできません （印刷に指定した解像度 300dpi 又は 600dpi のみ）	低速
	高濃度*1		高濃度

*1：DP-DS820 のみ
*2：DP-QW410 を除く

12. コマンド仕様

プリンター識別番号、接続数の取得

【書式】 long GetPrinterPortNum (char *pArray, int arraysize) ;

【引数】 pArray : 配列を格納する先頭のポインタ
arraysize : 配列の数を指定

【戻り値】 成功 : 接続プリンター台数
失敗 : 0

【説明】 現在接続されている機種の識別番号と、固体の識別番号を返します。
成功すると、接続されているプリンター台数を返します。

機種識別番号	機種
1	CV
2	CW
3	DS40
4	DS80
5	DS-RX1
6	CW02
7	DP-TC10
8	DP-DS80D
10	TOWER_CONTROLER
20	DP-DS620
30	DP-DS820
35	DP-QW410
90-99	予約領域

【文例】

```
< Visual C >
long Pnum;
char PnumArray[3][2];
if(( Pnum = GetPrinterPortNum( PnumArray , sizeof(PnumArray) )) < 0 ){
        // error
}

//機種、固体識別番号格納順
//PnumArray[0][0]    機種
//PnumArray[0][1]    固体識別番号
//PnumArray[1][0]    機種
//PnumArray[1][1]    固体識別番号
//PnumArray[2][0]    機種
//PnumArray[2][1]    固体識別番号

< Visual Basic >
Dim Pnum
Dim PnumArray (2,3)
Pnum = GetPrinterPortNum(PnumArray(0,0), 6)
If Pnum < 0 Then GoTo Error

'VC と配列順が異なります。
'PnumArray(0,0)      機種
'PnumArray(1,0)      固体識別番号
'PnumArray(0,1)      機種
'PnumArray(1,1)      固体識別番号
'PnumArray(0,2)      機種
'PnumArray(1,2)      固体識別番号
```

■タワーを接続する際の注意

GetPrinterPortNum()を実行すると、タワーコントローラを認識しUnitID No 10を返します。
プリンターと同様、タワーと直接通信が可能ですが、タワーで使用出来るAPIは次の関数に限られます。

GetFirmwVersion()
GetSerialNo()

プリンターフィルター機能設定

【書式】 long SetPrinterFilter(long Filter_Sw);

【引数】 Filter_Sw : フィルター機能設定

PFILTER_OFF (0) フィルター機能無効（初期値）
PFILTER_ON (1) フィルター機能有効

【戻り値】 フィルター機能有効 1
フィルター機能無効 0

【説明】 フィルター機能の有効／無効を設定します。
フィルター機能は、プリンターの機種、ファームウェアバージョンの違いによるAPIの対応／未対応をDLLが判断し、プリンターへのコマンド送信又は、発行の制限を行うものです。各関数の詳細は「6.CSP Status API関数一覧」を参照して下さい。

フィルター	機能
無効 （初期値）	DLLはプリンターへのコマンド発行を制限しません。 アプリケーションがプリンター未対応のAPIを誤って使用した場合、プリンターが停止する場合が有ります。フィルター機能の無い旧DLLはこちらに該当します。
有効	プリンター未対応のAPIの場合、DLLはプリンターへコマンドの発行を制限します。 スタンドアロンプリンターを使用する時は、こちらを指定して下さい。

【文例】

```
< Visual C >
if( SetPrinterFilter( PFILTER_ON ) == 1 ){
        // Filter ON
}

< Visual Basic >
Dim Result As Long
Result = SetPrinterFilter( PFILTAER_ON )
```

USB 通信タイムアウト時間設定

【書式】 long SetUSBTimeout(long lPortNum, long lTime);

【引数】 Time : 通信タイムアウト設定（秒）

0 : 通信タイムアウト無し（初期値）
10～120 : API が USB 通信（データ読み込み）が停止してから、タイムアウトするまでの時間。
設定は 1 秒単位。

【戻り値】 成功 : 1
失敗 : 0

【説明】 USB 通信タイムアウトの時間設定を行います。
USB 通信タイムアウトは、各 API がプリンターから返信データを読み込めなくなった場合に発生します。

設定値	機能
タイムアウト無し 0（初期値）	USB 通信が停止した場合、API はデータが読み込み出来るまでタイムアウトしないで待ちます。 プリンター1 台で通信が停止すると API が待ち状態のまま、他のプリンターの通信に入れないためシステム停止となります。 タイムアウト設定の無い旧 DLL はこちらに該当します。
タイムアウト時間 10～120 秒	USB 通信が停止した場合、API は設定時間を越えるとタイムアウトエラーとなり以降タイムウトが発生したプリンターへの通信は行いません。 プリンター1台が停止しても、他のプリンターへの通信は可能です。

以下の表は、タイムアウトまでの設定の参考時間です。
DS40T/80T の Firmware Ver0.21 までは、アプリケーションが印画データを送信してから、ステータス取得など返信要求をすると、通信データの読み込みが出来なくなる時間が発生します。タイムアウトの時間設定は、これよりも長い時間を設定して下さい。

プリンター動作	DS40T/80T Firm Ver0.21	タイムアウト設定 参考時間
通常印画	1秒以下	60 秒
用紙初期化時 （初期化中、次の画像転送～ステータス受信）	約 50 秒	90 秒
停電発生時リカバリ処理時 （リカバリ中、次の画像転送～ステータス受信）	約 100 秒	120 秒

【文例】

```
< Visual C >
if( SetUSBTimeout( PortNo、60 ) == 1 ){
        // TIMEOUT  60S
}

< Visual Basic >
Dim Result As Long
Result = SetUSBTimeout( PortNo, 60 )
```

メディアサイズ設定

【書式】 long SetMediaSize(long lPortNum,long Media);

【引数】 lPortNum： ポートナンバーの指定
Media： 用紙サイズ、画面割付方法指定 （ヘッダーにて定義、下図参照）

【戻り値】 成功： 1
失敗： 0 メディア範囲外(6inch プリンターに 8inch サイズの割付をした場合等)

【説明】 印画処理用に出力ページサイズを指定します。
出力ページサイズは、プリンターに搭載されているメディアサイズと比較し、同等かそれ以下のサイズを指定して下さい。又 P32 SendImageData にてイメージ転送を行う場合必ず本コマンドにて出力サイズを指定して下さい。

【文例】

```
< Visual C >
if( SetMediaSize(PortNo, CSP_PCx2) <= 0 ){
        // error
}

< Visual Basic >
Dim Result As Long
Result = SetMediaSize(PortNo, CSP_PCx2)
If Result <= 0 Then GoTo Error
```

■ヘッダー定義内容

Media	サイズ 割り付け	Media	サイズ 割り付け	Media	サイズ 割り付け
01	CSP_L	31	CSP_8x10	54	CSP_4x4
02	CSP_2L	32	CSP_8x12	55	CSP_4x6
03	CSP_PC	33	CSP_8x4		
04	CSP_A5	34	CSP_8x5	57	CSP_4P5x4P5
05	CSP_A5W	35	CSP_8x6	58	CSP_4P5x6
06	CSP_PCx2（PC 2 画面割付）	36	CSP_8x8	59	CSP_4P5x8
07	CSP_Lx2（L 2 画面割付）	37	CSP_8x4x2	60	CSP_4x3
08	CSP_PC_REWIND (PC 巻き戻し) ※1	38	CSP_8x5x2	61	CSP_4x4P5
09	CSP_L_REWIND （L 巻き戻し）※1	39	CSP_8x6x2	62	CSP_4P5x3
10	CSP_5x5 ※1	40	CSP_8x5_8x4	63	CSP_4P5x4
11	CSP_6x6 ※3	41	CSP_8x6_8x4	64	CSP_4x3x2
12	CSP_6x4P5 ※2	42	CSP_8x6_8x5	65	CSP_4P5x3x2
13	CSP_6x4P5x2 ※2	43	CSP_8x8_8x4	66	CSP_4P5x4x2
14	CSP_6x4P5_REWIND (6x4.5 巻き戻し) ※2	44	CSP_8x4x3	71	CSP_A5FORMAT
		45	CSP_A4_LENGTH	72	CSP_A5x2
21	CSP_TC86x61	46	CSP_8x7	73	CSP_A5_REWIND
22	CSP_TC127x61	47	CSP_8x9	74	CSP_A4x5
		48	CSP_8x4_REWIND	75	CSP_A4x6
		49	CSP_8x5_REWIND	76	CSP_A4x8
		50	CSP_8x6_REWIND	77	CSP_A4x10
				78	CSP_A4FORMAT
				79	CSP_A4x5x2
				80	CSP_A4x5_REWIND

※1：DP-DS620、及び DS-RX1 ファームウェアバージョン 2.06 以降対応

※2：DP-DS620 ファームウェアバージョン 1.10 以降のみ対応

※3：DP-DS620、及び DS40 ファームウェアバージョン 1.60 以降、
及び DS-RX1 ファームウェアバージョン 2.06 以降対応

■ヘッダー定義内容（両面 DP-DS80D 用　カット紙指定）

Media	サイズ　割り付け	Media	サイズ　割り付け
片面印画		両面印画　表面指定	
131	CSP_8x10_CutPaper	231	CSP_8x10_DuplexCutPaper_F
132	CSP_8x12_CutPaper	232	CSP_8x12_DuplexCutPaper_F
133	CSP_8x4_CutPaper	233	CSP_8x4_DuplexCutPaper_F
134	CSP_8x5_CutPaper	234	CSP_8x5_DuplexCutPaper_F
135	CSP_8x6_CutPaper	235	CSP_8x6_DuplexCutPaper_F
136	CSP_8x8_CutPaper	236	CSP_8x8_DuplexCutPaper_F
137	CSP_8x4x2_CutPaper	237	CSP_8x4x2_DuplexCutPaper_F
138	CSP_8x5x2_CutPaper	238	CSP_8x5x2_DuplexCutPaper_F
139	CSP_8x6x2_CutPaper	239	CSP_8x6x2_DuplexCutPaper_F
151	CSP_8x10_50_CutPaper	251	CSP_8x10_50_DuplexCutPaper_F
152	CSP_8x10_75_CutPaper	252	CSP_8x10_75_DuplexCutPaper_F
153	CSP_8x4x3_CutPaper	253	CSP_8x4x3_DuplexCutPaper_F
		両面印画　裏面指定	
		331	CSP_8x10_DuplexCutPaper_R
		332	CSP_8x12_DuplexCutPaper_R
		333	CSP_8x4_DuplexCutPaper_R
		334	CSP_8x5_DuplexCutPaper_R
		335	CSP_8x6_DuplexCutPaper_R
		336	CSP_8x8_DuplexCutPaper_R
		337	CSP_8x4x2_DuplexCutPaper_R
		338	CSP_8x5x2_DuplexCutPaper_R
		339	CSP_8x6x2_DuplexCutPaper_R
		351	CSP_8x10_50_DuplexCutPaper_R
		352	CSP_8x10_75_DuplexCutPaper_R
		353	CSP_8x4x3_DuplexCutPaper_R

イメージデータ転送

【書式】
```
long SendImageData(long lPortNum,LPSTR Buff,
                   long XPos,long YPos,long Width,long Height);
```

【引数】

lPortNum：	ポート番号
Buff：	RGB ビットマップ　Raw データの先頭のポインタ （データは RGB　24bit カラーデータ）
Xpos：	出力先の座標指定（Ｘ軸、Dot）
Ypos：	出力先の座標指定（Ｙ軸、Dot）
Height：	イメージの幅（Dot）
Width：	イメージの高さ(Dot)

【戻り値】

成功：	1
失敗：	0

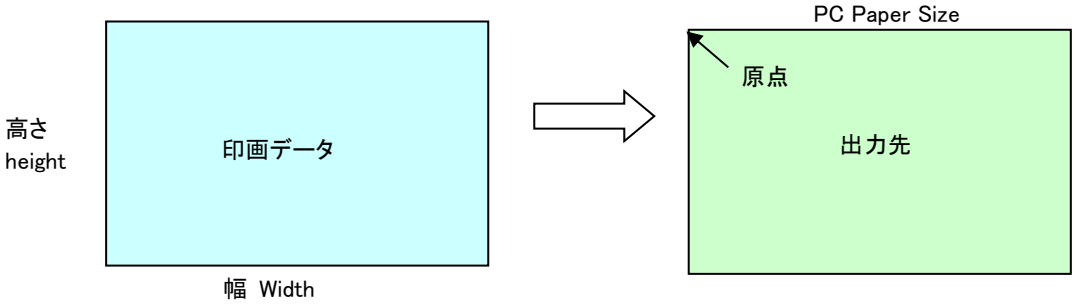
【説明】

イメージデータを送信します。
本コマンドでイメージデータを送信する際、必ず P30 SetMediaSize()コマンドにて出力サイズの指定を行って下さい。画像サイズはプリンターに搭載されているメディアサイズと比較し同等かそれ以下のサイズを指定して下さい。詳細は 4.印画処理フローを参照して下さい。

【文例】

```
< Visual C >
char RGBData[8478720]        //印画データのバッファ、
                             //例）数字は PC 画像サイズの場合 2048 x 1380 x 2
if( SetMediaSize(PortNo, CSP_PCx2) > 0 ){
        if(( SendImageData(PortNo, RGBData, 0, 0, 2048, 1380) <= 0 ){
                // error
        }
}

< Visual Basic >
Dim Result As Long
Dim RGBData(8478720) As Byte'印画データのバッファ、数字は PC 画像サイズの場合
                                        '例）数字は PC 画像サイズの場合 2048 x 1380 x 2
Result = SetMediaSize(PortNo, CSP_PCx2)
If Result <= 0 Then GoTo Error

Result = SendImageData(PortNo,VarPtr(RGBData(0)), 0, 0, 2048, 1380)
If Result <= 0 Then GoTo Error
```

※SendImageData の引数は、印画データの幅、高さ、出力先の座標指定が可能です。

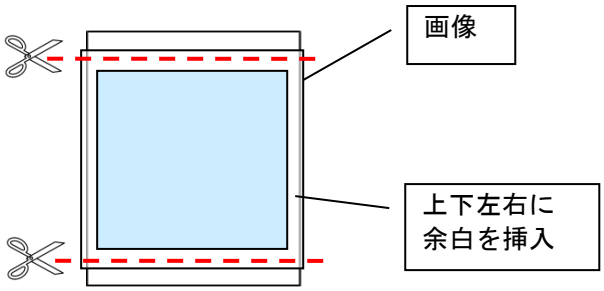


■イメージデータ転送方法の改良
　本コマンドにおいて通信動作安定化のため、Ver 0.591 及び Ver 0.601 以降のバージョンは，プレーンのデータの終端がショートパケット（512 バイト未満）になるようにデータを転送します。

■画像サイズ（CV, CW）

機種	用紙種類	印画エリアサイズ 横(ヘッド幅方向)× 縦(紙送り方向)	画像ピクセルサイズ	
			解像度 ： 334×600dpi	解像度 ： 334×334dpi
CV	DSC	121.5×92.0 mm	1598×2176 pixel	-
	L	130.0×92.0 mm	1710×2176 pixel	-
	PC	155.0×105.0 mm	2038×2480 pixel	-
	2DSC	121.5×181.0 mm	1598×4276 pixel	-
	2L	130.0×181.0 mm	1710×4276 pixel	-
	A5 (6x8)	155.0×206.0 mm	2038×4870 pixel	-
	A5W (6x9)	155.0×232.0 mm	2038×5480 pixel	-
CW	DSC	121.5×92.0 mm	1598×2176 pixel	1598×1210 pixel
	L	130.0×92.0 mm	1710×2176 pixel	1710×1210 pixel
	PC	155.0×105.0 mm	2038×2480 pixel	2038×1380 pixel
	2DSC	121.5×181.0 mm	1598×4276 pixel	1598×2380 pixel
	2L	130.0×181.0 mm	1710×4276 pixel	1710×2380 pixel
	A5 (6x8)	155.0×206.0 mm	2038×4870 pixel	2038×2710 pixel
	A5W (6x9)	155.0×232.0 mm	2038×5480 pixel	2038×3050 pixel
ふち余白分上下（参考値）		上下各 4.5 mm	上下各 120 pixel	上下各 60 pixel
ふち余白分左右（参考値）		左右各 4.5 mm	左右各 60 pixel	左右各 60 pixel

印画物の上下左右にふちを付けたい場合、画像の上下左右に余白を挿入します。余白のサイズは表中の値を参考にして下さい。



■画像サイズ（DP-QW410）

機種	用紙種類	印画エリアサイズ 横(ヘッド幅方向)× 縦(紙送り方向)	画像ピクセルサイズ	
			解像度 ： 300×600dpi	解像度 ： 300×300dpi
DP-QW410	4x3 (*1)	107.2×79.3 mm		1266×936 pixel
	4x4	107.2×104.7 mm		1266×1236 pixel
	4x4.5 (*1)	107.2×117.4 mm		1266×1386 pixel
	4x6	107.2×155.5 mm		1266×1836 pixel
	4.5x3 (*1)	119.3×79.3 mm		1408×936 pixel
	4.5x4 (*1)	119.3×104.7 mm		1408×1236 pixel
	4.5x4.5	119.3×117.4 mm		1408×1386 pixel
	4.5x6	119.3×155.5 mm		1408×1836 pixel
	4.5x8	119.3×206.3 mm		1408×2436 pixel
ふち余白分上下(参考値)		上下各 4.5 mm		上下各 54 pixel
ふち余白分左右(参考値)		左右各 5.54 mm		左右各 66 pixel

*1: ファームウェアバージョン 1.09 以降

■画像サイズ（DS40/80,DS40T/80T,DS-RX1,CW02,DP-DS80D,DP-DS620）

機種	用紙種類	印画エリアサイズ 横(ヘッド幅方向)× 縦(紙送り方向)	画像ピクセルサイズ	
			解像度 ： 300×600dpi	解像度 ： 300×300dpi
DS40/40T DS-RX1, CW02 DP-DS620	5x3.5(L)	131.1× 92.1 mm	1548×2176 pixel	1548×1088 pixel
	6x4(PC)	156.1×105.0 mm	1844×2480 pixel	1844×1240 pixel
	5x7(2L)	131.1×181.0 mm	1548×4276 pixel	1548×2138 pixel
	6x8(A5)	156.1×206.2 mm	1844×4872 pixel	1844×2436 pixel
DS40/40T CW02 DP-DS620 *1	6x9(A5W)	156.1×232.0 mm	1844×5480 pixel	1844×2740 pixel
DS-RX1 *3, DP-DS620	5x5	131.1×130.4 mm	1548×3080 pixel	1548×1540 pixel
DP-DS620 *1	6x4.5	156.1×117.3 mm	1844×2772 pixel	1844×1386 pixel
DS40 *2, DS-RX1 *3, DP-DS620	6x6	156.1×155.4 mm	1844×3672 pixel	1844×1836 pixel
DS80/80T DP-DS80D DP-DS820	(8x4)	207.3×104.6 mm	2448×2472 pixel	2448×1236 pixel
	(8x5)	207.3×130.0 mm	2448×3072 pixel	2448×1536 pixel
	(8x6)	207.3×155.4 mm	2448×3672 pixel	2448×1836 pixel
	(8x8)	207.3×206.2 mm	2448×4872 pixel	2448×2436 pixel
	(8x10)	207.3×257.0 mm	2448×6072 pixel	2448×3036 pixel
	(8x12)	207.3×307.8 mm	2448×7272 pixel	2448×3636 pixel
	A4 Length	207.3×300.0 mm	2448×7088 pixel	2448×3544 pixel
DP-DS820	(8x7)	207.3×180.8 mm	2448×4272 pixel	2448×2136 pixel
	(8x9)	207.3×231.6 mm	2448×5472 pixel	2448×2736 pixel
	A5 Format	214.0×151.0 mm	2528×3568 pixel	2528×1784 pixel
	A4x5	214.0×130.0 mm	2528×3072 pixel	2528×1536 pixel
	A4x6	214.0×155.4 mm	2528×3672 pixel	2528×1836 pixel
	A4x8	214.0×206.2 mm	2528×4872 pixel	2528×2436 pixel
	A4x10	214.0×257.0 mm	2528×6072 pixel	2528×3036 pixel
	A4 Format	214.0×300.0 mm	2528×7088 pixel	2528×3544 pixel
ふち余白分上下(参考値)		上下各 4.5 mm	上下各 108 pixel (PCは120 pixel)	上下各 54 pixel (PCは60 pixel)
ふち余白分左右(参考値)		左右各 5.0 mm	左右各 60 pixel	左右各 60 pixel

*1 DP-DS620 ファームウェアバージョン 1.10 以降
*2 DS40 ファームウェアバージョン 1.60 以降
*3 DS-RX1 ファームウェアバージョン 2.06 以降

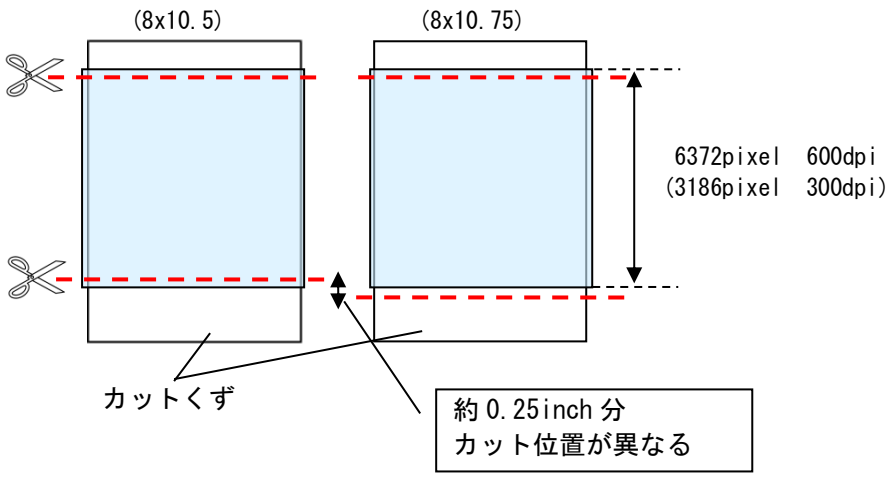
■画像サイズ（DP-TC10）

機種	用紙種類	印画エリアサイズ 横(ヘッド幅方向)× 縦(紙送り方向)	画像ピクセルサイズ	
			解像度 ： 300×600dpi	解像度 ： 300×300dpi
DP-TC10	TC86x61	90.1× 64.0 mm	1064x1512 pixel	1064x 756 pixel
	TC127x61	131.1× 64.0 mm	1548x1512 pixel	1548x 756 pixel
ふち余白分上下(参考値)		上下各 4.5 mm	上下各 108 pixel	上下各 54 pixel
ふち余白分左右(参考値)		左右各 5.0 mm	左右各 60 pixel	左右各 60 pixel

■画像サイズ（DP-DS80D,Cut Paper）

機種	用紙種類	印画エリアサイズ 横(ヘッド幅方向)× 縦(紙送り方向)	画像ピクセルサイズ	
			解像度 ： 300×600dpi	解像度 ： 300×300dpi
DP-DS80D	(8x4)	207.3x104.6 mm	2448x2472 pixel	2448x1236 pixel
	(8x5)	207.3x130.0 mm	2448x3072 pixel	2448x1536 pixel
	(8x6)	207.3x155.4 mm	2448x3672 pixel	2448x1836 pixel
	(8x8)	207.3x206.2 mm	2448x4872 pixel	2448x2436 pixel
	(8x10)	207.3x257.0 mm	2448x6072 pixel	2448x3036 pixel
	(8x10.5) ※	207.3x269.7 mm	2448x6372 pixel	2448x3186 pixel
	(8x10.75) ※	207.3x269.7 mm	2448x6372 pixel	2448x3186 pixel
	(8x12)	207.3x307.8 mm	2448x7272 pixel	2448x3636 pixel
ふち余白分上下(参考値)		上下各 4.5 mm	上下各 108 pixel	上下各 54 pixel
ふち余白分左右(参考値)		左右各 5.0 mm	左右各 60 pixel	左右各 60 pixel

※(8x10.5)、(8x10.75)の画像ピクセルサイズは同じです。出力される印画物が次のように異なります。



■タワープリンター設置位置による出力サイズ制限

以下のように、タワー内のプリンターDS40T では設置位置による大判サイズの出力制限、DS80T では設置位置の制限が有り、プリンターファームウェアのバージョンにより動作が異なります。

プリンター	旧ファームウェア	ファームウェア　Ver 0.32 以降～
DS40T	1 段目と 3 段目は L サイズと PC サイズのみに制限	1 段目は L サイズと PC サイズのみに制限
DS80T	1～3 段目への設置禁止	1 段目への設置禁止

DS40T プリンター位置別サイズ対応表（○＝印画可能、×＝印画不可、は　Ver 0.32 以降、印画に対応）

設置位置	印画物サイズ				
	5x3.5(L)	5x7(2L)	6x4(PC) ※	6x8(A5)	6x9(A5W)
1 段目	○	×	○	×	×
2 段目	○	○	○	○	○
3 段目	○	○	○	○	○
4 段目	○	○	○	○	○

※マルチカットによる印画を含む

1 段目の印画不可サイズの印画を指定した場合、プリンターは “Paper Definition Error” になります。

DS80T プリンター位置別サイズ対応表（○＝印画可能、×＝印画不可、は　Ver 0.32 以降、印画に対応）

設置位置	印画物サイズ						
	8x4 ※	8x5 ※	8x6 ※	8x8 ※	8x10	8x11.7	8x12
1 段目	×	×	×	×	×	×	×
2 段目	○	○	○	○	○	○	○
3 段目	○	○	○	○	○	○	○
4 段目	○	○	○	○	○	○	○

※マルチカットによる印画を含む

1 段目に設置した場合、印画指定時にプリンターは “Paper Definition Error” になります。

解像度の設定

【書式】 long SetResolution(long lPortNum, long lResolution);

【引数】
lPortNum : ポート番号
Media : 解像度指定　ヘッダーにて定義
RESOLUTION300 or RESOLUTION600

【戻り】
成功 : 1
失敗 : 0

【説明】
画像解像度を指定します。
DP-QW410 は RESOLUTION300 のみ有効です。
本関数は CV,CW には無効です。

【文例】

```
< Visual C >
if(( SetResolution( PortNo, RESOLUTION300)) <= 0 ){
        //Error
}

< Visual Basic >
Dim Result As Long
Result = SetResolution(PortNo, RESOLUTION300)
If Result <= 0 Then GoTo Error
```

印画枚数指定

【書式】 long SetPQTY(long lPortNum、long pqty);

【引数】
lPortNum : ポートナンバーの指定
pqty : 印画枚数の指定

【戻り値】
成功 : 1
失敗 : 0

【説明】
プリンターの印画枚数を指定します。

【文例】

```
< Visual C >
if( SetPQTY(PortNo, 2) <= 0 ){
        // error
}

< Visual Basic >
Dim Result As Long
Result = SetPQTY(PortNo, 2)
If Result <= 0 Then GoTo Error
```

プリント開始

【書式】 long PrintImageData(long lPortNum);

【引数】 lPortNum： ポート番号

【戻り値】 成功： 1
失敗： 0

【説明】 プリントを開始します。(複数画面割付ページの端数画像を印刷)

【文例】

```
< Visual C >
if( PrintImageData(PortNo) <= 0 ){
        // error
}

< Visual Basic >
Dim Result As Long
Result = PrintImageData(PortNo);
If Result <= 0 Then GoTo Error
```

ページレイアウトの開始

【書式】 long StartPageLayout(long lPortNum);

【引数】 lPortNum： ポート番号

【戻り値】 成功： 1
失敗： 0

【説明】 ページレイアウトを開始します。
必ずページレイアウトの終了と組み合わせて使用します。

【文例】

```
< Visual C >
if( StartPageLayout(PortNo) <= 0 ){
        // error
}
/* 画像処理 */
EndPageLayout(PortNo) ;

< Visual Basic >
Dim Result As Long
Result = StartPageLayout(PortNo)
If Result <= 0 Then GoTo Error
'画像処理
EndPageLayout(PortNo)
```

ページレイアウトの終了

【書式】 long EndPageLayout(long lPortNum);

【引数】 lPortNum : ポート番号

【戻り値】 成功 : 1
失敗 : 0

【説明】 ページレイアウトを終了します。
必ずページレイアウトの開始と組み合わせて使用します。

【文例】

```
< Visual C >
StartPageLayout(PortNo) ;
/* 画像処理 */
if( EndPageLayout(PortNo) <= 0 ){
        // error
}

< Visual Basic >
Dim Result As Long
StartPageLayout(PortNo)
'画像処理
Result = EndPageLayout(PortNo)
If Result <= 0 Then GoTo Error
```

バージョン情報の取得

【書式】 long GetFirmwVersion(long lPortNum, LPSTR p);
long CvGetVersion(long lPortNum, LPSTR p);

【引数】 lPortNum : ポート番号
p : 受信バッファへのポインタ

【戻り】 成功 : バッファ p へ受信した文字数
失敗 : -1

【説明】 プリンターのバージョン情報を文字列バッファへ受信します。

【文例】

```
< Visual C >
char rbuf[256];
if( GetFirmwVersion( PortNo, (LPSTR)rbuf ) > 0 ){
        // 処理
}

< Visual Basic >
Dim rb As String * 255, Ver As String
Dim n As Long
n = GetFirmwVersion(PortNo, rb)
If n > 0 Then Ver = Left(rb, n)        '文字列を切出す場合
```

【バージョン情報】
DS *.** ファームウェアのバージョン情報

センサー情報の取得

【書式】
```
long GetSensorInfo( long lPortNum, LPSTR p );
long CvGetSensorInfo( long lPortNum, LPSTR p );
```

【引数】

lPortNum：　ポート番号
p：　受信バッファへのポインタ

【戻り】

成功：　バッファ p へ受信した文字数
失敗：　-1

【説明】

各種センサーの情報（ＡＤ値）を文字列バッファへ受信します。

【文例】

```
< Visual C >
char rbuf[256];
if( GetSensorInfo( PortNo, (LPSTR)rbuf ) > 0 ){
        // 処理
}

< Visual Basic >
Dim rb As String * 255, Sen As String
Dim n As Long
n = GetSensorInfo(PortNo, rb)
If n > 0 Then Sen = Left(rb, n)        '文字列を切出す場合
```

【センサー情報】

HDT-***;	ヘッド温度検出センサー	
MDT-***;	メディア温度検出センサー	
PMK-***;	用紙マーク検出センサー	
RML-***;	左リボンマーク検出センサー	(CV,CW 機以外は未使用、常に 000 を返します)
RMC-***;	中央リボンマーク検出センサー	
RMR-***;	右リボンマーク検出センサー	
PSZ-***;	用紙サイズ検出センサー	(CV,CW 機以外は未使用、常に 000 を返します)
PNT-***;	用紙ノッチ検出センサー	(CV,CW 機以外は未使用、常に 000 を返します)
PJM-***;	用紙ジャム検出センサー	(CV,CW 機以外は未使用、常に 000 を返します)
PED-***;	ペーパーエンド検出センサー	
PET-***;	ペーパーエンプティ検出センサー	(CV,CW 機以外は未使用、常に 000 を返します)
HDV-***;	ヘッド電圧検出センサー	
HMD-***:	湿度センサー	(CV を除く)
RP1-***:	ロール紙先端検出センサー1	(DP-DS80D のみ対応)
RP2-***:	ロール紙先端検出センサー2	(DP-DS80D のみ対応)
CSR-***:	カラーセンサ(赤)	(DP-DS620、DP-DS820 でのみ対応)
CSG-***:	カラーセンサ(緑)	(DP-DS620、DP-DS820 でのみ対応)
CSB-***:	カラーセンサ(青)	(DP-DS620、DP-DS820 でのみ対応)
DC5-***:	USB 給電電圧検出センサー	(DP-QW410 のみ対応)

解像度の取得

【書式】

```
long GetResolutionH( long lPortNum );
long CvGetResolutionH( long lPortNum );
long GetResolutionV( long lPortNum );
long CvGetResolutionV( long lPortNum );
```

【引数】

lPortNum：　ポート番号

【戻り】

成功：　水平、垂直解像度(dpi)
失敗：　-1

【説明】

GetResolutionH(),CvGetResolutionH()は水平解像度(dpi)を返します。
GetResolutionV(),CvGetResolutionV()は垂直解像度(dpi)を返します。

【文例】

< Visual C >

```
long rh;
if(( rh = GetResolutionH( PortNo )) >= 0 ){
        // rhに水平解像度値
}
```

< Visual Basic >

```
Dim rh As Long
rh = GetResolutionH(PortNo)
```

メディアコードの取得

【書式】
```
long GetMedia( long lPortNum, LPSTR p );
long CvGetMedia( long lPortNum, LPSTR p );
```

【引数】
lPortNum：　ポート番号
p：　受信バッファへのポインタ

【戻り】
成功：　バッファ p へ受信した文字数
失敗：　-1

【説明】
メディアコードを文字列バッファへ受信します。

【文例】
```
< Visual C >
char rbuf[256];
if( GetMedia( PortNo, (LPSTR)rbuf ) > 0 ){
        // 処理
}

< Visual Basic >
Dim rb As String * 255, MediaCode As String
Dim n As Long
n = GetMedia(PortNo, rb)
If n > 0 Then MediaCode = Left(rb, n) '文字列を切出す場合
```

【メディアコード】
本プリンターでは、使用するメディアを全て 5 桁の 10 進数で表します。
メディアコード取得コマンドにより返信するデータは、各メディアに割り当てた 5 桁のコード(ASCII 数字)になります。

5、4 桁目 (nn000)　予約	3、2 桁目 (00nn0)　用紙サイズ	1 桁目 (0000n)
nn = 00	00150 番　4 インチ幅	0:4x6
	00160 番　4.5 インチ幅	1:4.5x8 3:4.5x6
	00200 番　5x3.5 (L) 00210 番　5x7 (2L) 00300 番　6x4 (PC) 00310 番　6x8 (A5) 00400 番　6x9 (A5W) 00500 番　8x10 00510 番　8x12 00600 番　A4 00120 番　TC86x61 00130 番　TC127x61	n = 0 または 1 (1:6x4 のみ)

(メディアのサイズ以外に関する情報はメディア拡張コード取得コマンドにて対応)

コード例

用紙サイズ	(サイズ: 幅 x 長さ)	メディアコード
5x3.5 (L)	(127.0 x 89.0 mm)	00200
6x4 (PC)	(152.0 x 102.0 mm)	00301
5x7 (2L)	(127.0 x 178.0 mm)	00210
6x8 (A5)	(152.0 x 203.0 mm)	00310
6x9 (A5W)	(152.0 x 229.0 mm)	00400

プリンターステータスの取得

【書式】
long GetStatus(long lPortNum);
long CvGetStatus(long lPortNum);

【引数】
lPortNum : ポート番号

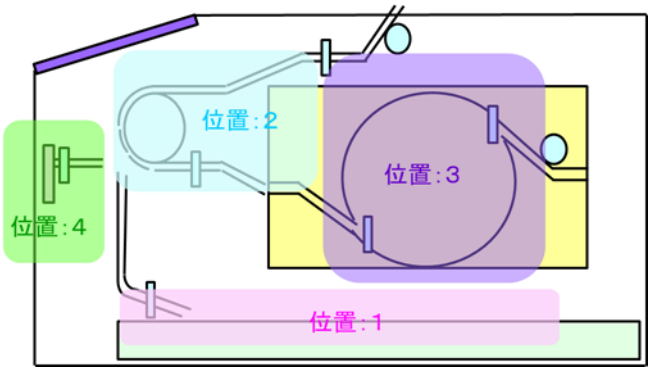
【戻り】
成功 : ステータス
失敗 : STATUS_ERROR または CSPSTATUS_TIMEOUT

【説明】
プリンターステータスを返します。
ステータスのビット位置は CspStat.h にマクロ定義されています。
シンボルの意味は下記の通りです。

GROUP_USUALLY	[通常のステータス]	グループを示すビット
GROUP_SETTING	[ユーザー設定エラー]	グループを示すビット
GROUP_HARDWARE	[ハードエラー]	グループを示すビット
GROUP_SYSTEM	[内部エラー]	グループを示すビット
GROUP_FLSHPROG	[書換えモード]	グループを示すビット
GROUP_DUNIT	[両面機エラー]	グループを示すビット
STATUS_USUALLY_STANDBY_MODE	スタンバイモード（DP-DS620）	
STATUS_ERROR	ステータス受信エラー	
CSPSTATUS_TIMEOUT	タイムアウトエラー（コマンドが取得出来ない）	
STATUS_USUALLY_IDLE	アイドル	
STATUS_USUALLY_PRINTING	印画中	
STATUS_USUALLY_PAPER_END	ペーパーエンド	
STATUS_USUALLY_RIBBON_END	リボンエンド	
STATUS_USUALLY_COOLING	ヘッド冷却中	
STATUS_USUALLY_MOTCOOLING	モータ冷却中	
STATUS_USUALLY_SHOOTING	シューティング中	
STATUS_USUALLY_BACKPRINT	バックプリント中	
STATUS_SETTING_COVER_OPEN	カバーオープン	
STATUS_SETTING_PAPER_JAM	ペーパージャム	
STATUS_SETTING_RIBBON_ERR	リボンエラー（検出ミス、リボン切れ）	
STATUS_SETTING_PAPER_ERR	用紙設定ミス	
STATUS_SETTING_DATA_ERR	データエラー（不正コマンド）	
STATUS_SETTING_SCRAPBOX_ERR	クズボックスエラー	
STATUS_HARDWARE_ERR01	ヘッド電圧異常	
STATUS_HARDWARE_ERR02	ヘッド位置エラー	
STATUS_HARDWARE_ERR03	電源ファン停止	
STATUS_HARDWARE_ERR04	カッターエラー（噛み込み等）	
STATUS_HARDWARE_ERR05	ピンチローラー位置エラー	
STATUS_HARDWARE_ERR06	ヘッド温度異常	
STATUS_HARDWARE_ERR07	メディア温度異常	
STATUS_HARDWARE_ERR08	リボンテンション異常	
STATUS_HARDWARE_ERR09	RFID モジュール異常	
STATUS_HARDWARE_ERR10	巻き取りモータ温度異常	
STATUS_CXM_RS422_STOP	CXM タワー RS422 通信ストップ	
STATUS_HARDWARE_ERR12	USB 給電電圧異常	
STATUS_SYSTEM_ERR01	システムエラー	
STATUS_FLSHPROG_IDLE	書換えデータ受信待ち	
STATUS_FLSHPROG_DATA_ERR1	転送データ関連エラー	
STATUS_FLSHPROG_DEVICE_ERR1	デバイス関連エラー	
STATUS_FLSHPROG_OTHERS_ERR1	その他エラー	

反転ユニットのエラー

UNIT_ERROR_JAMMING_SUPPLY	反転ユニット給紙部での紙詰まり（位置：1）
UNIT_ERROR_JAMMING_PASS	反転ユニット上部搬送部付近での紙詰まり（位置：2）
UNIT_ERROR_JAMMING_SHELL	反転ユニットシェル付近での紙詰まり（位置：3）
UNIT_ERROR_JAMMING_EJECT	反転ユニット排紙口付近での紙詰まり（位置：4）
UNIT_ERROR_CAPSTAN_MOTOR	反転ユニット紙搬送モータ異常
UNIT_ERROR_SHELL_MOTOR	反転ユニット反転ブロック（シェル）内搬送モータ異常
UNIT_ERROR_PINCH	反転ユニットピンチ動作異常
UNIT_ERROR_PASS_GUIDE	反転ユニット可動パスガイド動作異常
UNIT_ERROR_SKEW_GUIDE	反転ユニットスキューガイド動作異常
UNIT_ERROR_SKEW_REJECT	反転ユニットスキュー修正異常
UNIT_ERROR_SHELL_ROTATE	反転ユニット反転ブロック（シェル）動作異常
UNIT_ERROR_LEVER	反転ユニット給紙レバー動作異常
UNIT_ERROR_CUTTER	反転ユニットカット動作異常
UNIT_ERROR_TRAY_OUT	反転ユニットトレイエラー
UNIT_ERROR_COVER_OUT	反転ユニットメンテナンス用蓋オープン
UNIT_ERROR_SYSTEM	反転ユニットシステムエラー



【文例】

```
< Visual C >
long stat;
stat = GetStatus( PortNo );
if( stat & GROUP_USUALLY ){ // 通常のステータスグループ
        switch( stat ){
                case STATUS_USUALLY_IDLE: ;
                case STATUS_USUALLY_PRINTING: ;
                case STATUS_USUALLY_PAPER_END: ;
                        :
        }
}
if( stat & GROUP_SETTING ){ // 設定エラーステータスグループ
        switch( stat ){
                case STATUS_SETTING_COVER_OPEN: ;
                        :
                        :
        }
}
< Visual Basic >
Dim stat As Long
stat = GetStatus(PortNo)
If stat And GROUP_USUALLY Then        '通常のステータスグループ
        Select Case stat
                Case STATUS_USUALLY_IDLE: Text1.Text = "IDLE"
                Case STATUS_USUALLY_PRINTING: Text1.Text = "PRINTING"
                Case STATUS_USUALLY_PAPER_END: Text1.Text = "PAPER_END"
                Case STATUS_USUALLY_RIBBON_END: Text1.Text = "RIBBON_END"
                Case STATUS_USUALLY_COOLING: Text1.Text = "COOLING"
        End Select
ElseIf stat And GROUP_SETTING Then    '設定エラーステータスグループ
        '設定エラー　（省略）
ElseIf stat And GROUP_HARDWARE Then   'ハードエラーステータスグループ
        'ハードエラー（省略）
ElseIf stat And GROUP_SYSTEM Then     'システムエラーステータスグループ
        'システムエラー（省略）
End If
```

反転ユニット、ステータスの取得（DP-DS80D）

【書式】　long GetDunitStatus(long lPortNum);

【引数】　lPortNum：　ポート番号

【戻り】　成功：　反転ユニットステータス
　　失敗：　STATUS_ERROR または CSPSTATUS_TIMEOUT

【説明】　反転ユニットのステータスを返します。

ステータスのビット位置は CspStat.h にマクロ定義されています。シンボルの意味は下記の通りです。

反転ユニットステータス

NON_ERROR	エラーなし
UCS_NONCONECT	反転ユニットが接続されていない
UCS_JAMMING_SUPPLY_SENS_ON	印画紙を給紙センサーを外れるまで搬送できずセンサーが ON（紙なし）にならなかった（給紙動作）
UCS_JAMMING_SUPPLY_SENS_OFF	印画紙を給紙センサーまで搬送できずセンサーが OFF（紙あり）にならなかった（給紙動作）
UCS_JAMMING_SLOT_SENS_ON	プリンター受け渡し口センサーを外れるまで紙を搬送できずセンサーが ON（紙なし）にならなかった（給紙・反転動作）
UCS_JAMMING_SLOT_SENS_OFF	プリンター受け渡し口センサーまで紙を搬送できずセンサーが OFF（紙あり）にならなかった（給紙・反転動作）
UCS_JAMMING_PASS_SENS_ON	ピンチパスセンサを外れるまで紙を搬送できずセンサーが ON（紙なし）にならなかった（反転・排紙動作）
UCS_JAMMING_PASS_SENS_OFF	ピンチパスセンサまで紙を搬送できずセンサーが OFF（紙あり）にならなかった（反転・排紙動作）
UCS_JAMMING_SHELL_SENS1_ON	シェルパスセンサ 1 から外れるまで紙を搬送できずセンサーが ON（紙なし）にならなかった（反転・排紙動作）
UCS_JAMMING_SHELL_SENS1_OFF	シェルパスセンサ 1 まで紙を搬送できずセンサーが OFF（紙あり）にならなかった（反転・排紙動作）
UCS_JAMMING_SHELL_SENS2_ON	シェルパスセンサ 2 から外れるまで紙を搬送できずセンサーが ON（紙なし）にならなかった（反転動作）
UCS_JAMMING_SHELL_SENS2_OFF	シェルパスセンサ 2 まで紙を搬送できずセンサーが OFF（紙あり）にならなかった（反転動作）
UCS_JAMMING_EJECT_SENS_ON	排紙センサーから外れるまで紙を搬送できずセンサーが ON（紙なし）にならなかった（排紙動作）
UCS_JAMMING_EJECT_SENS_OFF	排紙センサーまで紙を搬送できずセンサーが OFF（紙あり）にならなかった（排紙動作）
UCS_JAMMING_SLOT_FG_SENS	印画中にプリンター受け渡しFGセンサーが 検出できなかった（印画中）
UCS_JAMMING_SHELL_FG_SENS	紙をシェル内から外に搬送中にシェル内FGセンサーが 検出できなかった（反転動作）
UCS_JAMMING_PAPER_JAM	電源 ON 時に搬送パス・シェル内に紙が残っている（電源投入時）
UCS_REMAIN_SUPPLY_SENS	給紙センサーが OFF（紙あり）状態
UCS_REMAIN_SLOT_SENS	プリンター受け渡し口センサーが OFF（紙あり）状態
UCS_REMAIN_PASS_SENS	ピンチパスセンサーが OFF（紙あり）状態
UCS_REMAIN_SHELL_SENS1	シェルパスセンサー1が OFF（紙あり）状態
UCS_REMAIN_SHELL_SENS2	シェルパスセンサー2が OFF（紙あり）状態
UCS_REMAIN_EJECT_SENS	排紙センサーが OFF（紙あり）状態
UCS_CAPSTAN_TRAPZ_ERROR	キャプスタンの駆動制御が設定時間内に終了しない
UCS_SHELL_ROLLER_ERROR	シェルローラーの回転異常（遅い、もしくは止まっている）
UCS_PINCH_OPEN_ERROR	ピンチローラーを解放ポジションまで動かすことができなかった
UCS_PINCH_CLOSE_ERROR	ピンチローラーを圧着ポジションまで動かすことができなかった
UCS_PINCH_INIT_ERROR	ピンチローラーのイニシャライズ動作が完了できなかった
UCS_PINCH_POS_UNKNOWN	ピンチローラーのポジション不定（試作時のみ）
UCS_PASS_GUIDE_SUPPLY_ERROR	パスガイドを給紙ポジションまで動かすことができなかった
UCS_PASS_GUIDE_SHELL_ERROR	パスガイドを引き込みポジションまで動かすことができなかった
UCS_PASS_GUIDE_EJECT_ERROR	パスガイドを排紙ポジションまで動かすことができなかった
UCS_PASS_GUIDE_INIT_ERROR	パスガイドのイニシャライズ動作が完了できなかった
UCS_PASS_GUIDE_POS_UNKNOWN	パスガイドポジション不定（試作時のみ）

UCS_SIDE_GUIDE_HOME_ERROR	スキューガイドをHOMEポジションまで動かすことができなかった
UCS_SIDE_GUIDE_POS_ERROR	スキューガイドを所定のポジションまで動かすことができなかった
UCS_SIDE_GUIDE_INIT_ERROR	スキューガイドのイニシャライズ動作が完了できなかった
UCS_ACT_GUIDE_HOME_ERROR	可動ガイドをHOMEポジションまで動かすことができなかった
UCS_SHELL_ROTATE_HOME_ERROR	シェルをHOMEポジションまで動かすことができなかった
UCS_SHELL_ROTATE_REV_ERROR	シェルをREVポジションまで動かすことができなかった
UCS_LEVER_DOWN_ERROR	給紙レバーをDOWNポジションまで動かすことができなかった
UCS_LEVER_LOCK_ERROR	給紙レバーを LOCK ポジションまで動かすことができなかった
UCS_LEVER_UP_ERROR	給紙レバーをUPポジションまで動かすことができなかった
UCS_CUTTER_HOME_ERROR	カッターをHOMEポジションまで動かすことができなかった
UCS_CUTTER_AWAY_ERROR	カッターをAWAYポジションまで動かすことができなかった
UCS_CUTTER_INIT_ERROR	カッターのイニシャライズ動作が完了できなかった
UCS_CUTTER_POS_UNKNOWN	カッターポジション不定（試作時のみ）
UCS_TRAY_OUT	印画中にペーパートレイが抜けた
UCS_TOP_COVER_OPEN	印画中のメンテナンス用蓋が開いた

反転ユニット、カット紙情報取得（DP-DS80D）

【書式】　long GetCutPaper(long lPortNum, LPSTR p);

【引数】　lPortNum：　ポート番号
p：　受信バッファへのポインタ

【戻り】　成功：　バッファ p へ受信した文字数
失敗：　-1

【説明】　用紙サイズ（トレイガイド位置）、用紙状態を文字列バッファへ受信します。

反転ユニットステータス

4､3 桁目、 （nn00）　用紙サイズ （トレイガイド位置）	2､1 桁目 （00nn）用紙状態
0000 番　ユニット接続なし 0100 番　8 x 10.75 0200 番　8 x 12	0000 番　カット紙無 0001 番　保護シート 0002 番　カット紙有

コード例

用紙サイズ	用紙状態	メディアコード
8x10.75	有り	0102
8x10.75	保護シート	0101
8x10.75	無し	0100

反転ユニット、裏面印画キャンセルと用紙排出（DP-DS80D）

【書式】　BOOL SetCancelDuplexPrint(long lPortNum);

【引数】　lPortNum：　ポート番号

【戻り】　成功：　TRUE
失敗：　FLASE

【説明】　両面印画中、表面印画後に裏面印画をキャンセルして用紙を排出します。
本コマンドは、表面印画終了後、リボンエンドとなった場合等、プリンター内に待機している用紙の強制排出に使用します。

本コマンドとプリンター動作の関係は、プリンターの状態と裏面印画データの送信状態で異なります。
動作は下表を参照して下さい。

コマンドとプリンター動作の関係

プリンターの状態	裏面印画データの送信状態	
	裏面データ送信後	裏面データ送信前
表面印画終了後、 リボンエンドとなった場合	表面印画で用紙を排出 プリンター印画バッファークリア	表面印画で用紙を排出
上記以外	両面印画後、用紙を排出 コマンドは無効	表面印画で用紙を排出

【文例】

```
< Visual C >
SetCancelDuplexPrint( PortNo );

< Visual Basic >
SetCancelDuplexPrint(PortNo)
```

カウンタ値の取得

【書式】

```
long GetCounterL( long lPortNum );
long CvGetCounterL( long lPortNum );

long GetCounterA( long lPortNum );
long CvGetCounterA( long lPortNum );

long GetCounterB( long lPortNum );
long CvGetCounterB( long lPortNum );

long GetCounterP( long lPortNum );                (DS40/80,DS-RX1,CW02,DP-DS80D,DP-DS620,DP-DS820 機)

long GetCounterMatte( long lPortNum );            (DS40/80,DS-RX1,CW02,DP-DS80D,DP-DS620,DP-DS820 機)
long GetCounterM( long lPortNum );                (DS40/80,DS-RX1,CW02,DP-DS80D,DP-DS620,DP-DS820 機)
long GetCounterDuplex( long lProtNum );           (DP-DS80D 機)
```

【引数】

lPortNum：　ポート番号

【戻り】

成功：　カウンタの値
失敗：　-1

【説明】

GetCounterL(),CvGetCounterL()は、ライフカウンタの値を返します。
GetCounterA(),CvGetCounterA()は、カウンタ A の値を返します。
GetCounterB(),CvGetCounterB()は、カウンタ B の値を返します。
GetCounterP(long lPortNum)は、カウンタ P の値を返します。
GetCounterMatte()は、マットカウンタの値を返します。
GetCounterM()は、カウンタ M の値を返します。
GetCounterDuplex はカウンタ Duplex の値を返します。

■ライフカウンタ、カウンタ A、カウンタ B

・6 インチ幅サイズの場合､1 枚発行する毎に 1 カウントアップします。但し 6x9(A5W), 6x8(A5), 5x7(2L) サイズの場合は 1 枚発行する毎に 2 カウントアップします（PC/L サイズ換算)。マルチカットプリントの場合は 2 枚目の発行時に 2 カウントアップします。

・8 インチ幅サイズのスタンドアロンモデルの場合、1 枚発行する毎に 1 カウントアップします。マルチカットプリントの場合は、2 枚目（3 面割付は 3 枚目）発行時に 1 カウントアップします。

・8 インチ幅サイズのタワーモデルの場合は、印画サイズに応じて 1～3 カウントアップします（次ページ表参照)。

・カウンタ A は SetClearCounterA()関数で、カウンタ B は SetClearCounterB()関数でクリア可能です。

■カウンタ P

・電源投入時に 0 に初期化されます。SetCounterP()関数で任意の値に設定出来ます。

・1 画像排出毎に 1 ずつカウントアップします。

■マットカウンタ、カウンタ M

・オーバーコート仕上げをマットで印刷したときにカウントアップします（ライフカウンタ、カウンタ A/B もカウントアップします)。

・カウンタ M は SetClearCounterM()関数でクリア可能です。

・これらのカウンタは、マット印刷対応のプリンター(DS40/80 Ver.1.30 以降、DS40T/80T Ver.0.31 以降)で有効です。

・カウントアップ仕様は、上記ライフカウンタ、カウンタ A/B と同様です。

■カウンタ Duplex

・カウンタ Duplex は、カット紙で印画（反転ユニットを使用）した時にカウントアップします。
・カウンタ Duplex は、カット紙の片面を印画した時に 1 カウントアップ、両面を印刷した場合は 2 カウントアップします。

【文例】

```
< Visual C >
long counter;
if(( counter = GetCounterL( PortNo )) >= 0 ){
        // counter にライフカウンタ値
}
< Visual Basic >
Dim counter As Long
counter = GetCounterL(PortNo)
```

カウンタ L/A/B/P カウントアップの動作

■タワー
カウントアップするタイミングは、プリント画像がタワーのベルトソーターまたは上部トレイへ正常に搬送された後です。
エラーが発生したときはカウントアップされません。
カウンタ P は電源オフで 0 にクリアされます。

DS40T	プリントサイズ		Counter L/A/B	Counter P
シングルカット	5x3.5 (L)		+1	+1
	6x4 (PC)		+1	+1
	5x7 (2L)		+2	+1
	6x8 (A5)		+2	+1
	6x9 (A5W)		+2	+1
マルチカット	6x4x2	1 面目	---	+1
	(PC 2 画面割付)	2 面目	+2	+1

DS80T	プリントサイズ		Counter L/A/B	Counter P
シングルカット	8x10		+3	+1
	8x12		+3	+1
	8x4		+1	+1
	8x5		+1	+1
	8x6		+1	+1
	8x8		+2	+1
	A4_Length		+3	+1
マルチカット	8x4x2	1 面目	---	+1
		2 面目	+2	+1
	8x5x2	1 面目	---	+1
		2 面目	+2	+1
	8x6x2	1 面目	---	+1
		2 面目	+2	+1
	8x5+8x4	1 面目	---	+1
		2 面目	+2	+1
	8x6+8x4	1 面目	---	+1
		2 面目	+2	+1
	8x6+8x5	1 面目	---	+1
		2 面目	+2	+1
	8x8+8x4	1 面目	---	+1
		2 面目	+2	+1
	8x4x3	1 面目	---	+1
		2 面目	---	+1
		3 面目	+3	+1

■スタンドアロン
カウントアップするタイミングは、プリント画像が正常にカットされた後です。
エラーが発生したときはカウントアップされません。カウンタ P は電源オフで 0 にクリアされます。

DS40,DS-RX1, CW02,DP-DS620	プリントサイズ		カウンタ L/A/B	カウンタ P (*1)
シングルカット	5x3.5 (L)		+1	+1
	6x4 (PC)		+1	+1
	6x4.5 (*3)		+1	+1
	5x5 (*5)		+2	+1
	6x6 (*4)		+2	+1
	5x7 (2L)		+2	+1
	6x8 (A5)		+2	+1
	6x9 (A5W) *DS-RX1 を除く		+2	+1
マルチカット	6x4x2	1 面目	---	+1
	(PC 2 画面割付)	2 面目	+2	+1
	6x4.5x2 (*3)	1 面目	---	+1
	(6x4.5 2 画面割付)	2 面目	+2	+1
	5x3.5x2 (*2)	1 面目	---	+1
	(L 2 画面割付)	2 面目	+2	+1
2 インチカット	6x4	1 枚目	---	+1
*対応機種のみ		2 枚目	+1	+1
カッター制御コマンド	6x8	1 枚目	---	+1
参照		2 枚目	---	+1
		3 枚目	---	+1
		4 枚目	+2	+1

(*1)DS40 ファームウェアバージョン 1.04 以降
(*2)DS-RX1 ファームウェアバージョン 1.20 以降、及び DP-DS620
(*3) DP-DS620 ファームウェアバージョン 1.10 以降
(*4)DS40 ファームウェアバージョン 1.60 以降、DS-RX1 ファームウェアバージョン 2.06 以降、及び DP-DS620
(*5)DS-RX1 ファームウェアバージョン 2.06 以降、及び DP-DS620

DS80	プリントサイズ		カウンタ L/A/B	カウンタ P (*1)
シングルカット	8x10		+1	+1
	8x12		+1	+1
	8x4		+1	+1
	8x5		+1	+1
	8x6		+1	+1
	8x8		+1	+1
	A4_Length		+1	+1
マルチカット	8x4x2	1 面目	---	+1
		2 面目	+1	+1
	8x5x2	1 面目	---	+1
		2 面目	+1	+1
	8x6x2	1 面目	---	+1
		2 面目	+1	+1
	8x5+8x4	1 面目	---	+1
		2 面目	+1	+1
	8x6+8x4	1 面目	---	+1
		2 面目	+1	+1
	8x6+8x5	1 面目	---	+1
		2 面目	+1	+1
	8x8+8x4	1 面目	---	+1
		2 面目	+1	+1
	8x4x3	1 面目	---	+1
		2 面目	---	+1
		3 面目	+1	+1

(*1) DS80 ファームウェアバージョン 1.02 以降

DP-TC10	プリントサイズ	カウンタ L/A/B	カウンタ P
シングルカット	TC86x61	+1	+1
	TC127x61	+1	+1
2 画面連続印画	TC127x61	+2	+1

DP-DS820	プリントサイズ		カウンタ L/A/B	カウンタ P
シングルカット	8x4		+1	+1
	8x5		+1	+1
	8x6		+1	+1
	8x7		+1	+1
	8x8		+1	+1
	8x9		+1	+1
	8x10		+1	+1
	8x12		+1	+1
	A4_Length		+1	+1
	A5 Format		+1	+1
	A4x5		+1	+1
	A4x6		+1	+1
	A4x8		+1	+1
	A4x10		+1	+1
	A4 Format		+1	+1
マルチカット	8x4x2	1 面目	---	+1
		2 面目	+1	+1
	8x5x2	1 面目	---	+1
		2 面目	+1	+1
	8x6x2	1 面目	---	+1
		2 面目	+1	+1
	8x5+8x4	1 面目	---	+1
		2 面目	+1	+1
	8x6+8x4	1 面目	---	+1
		2 面目	+1	+1
	8x6+8x5	1 面目	---	+1
		2 面目	+1	+1
	8x8+8x4	1 面目	---	+1
		2 面目	+1	+1
	8x4x3	1 面目	---	+1
		2 面目	---	+1
		3 面目	+1	+1
	A5x2	1 面目	---	+1
		2 面目	+1	+1
	A4x5x2	1 面目	---	+1
		2 面目	+1	+1
2 インチカット	8x4	1 面目	---	+1
		2 面目	+1	+1
	8x6	1,2 面目	---	+1/面
	A4x6	3 面目	+1	+1
	8x8	1-3 面目	---	+1/面
	A4x8	4 面目	+1	+1
	8x10	1-4 面目	---	+1/面
	A4x10	5 面目	+1	+1
	8x12	1-5 面目	---	+1/面
		6 面目	+1	+1

CV、CW	プリントサイズ		Counter L/A/B
シングルカット	5x3.5 (L)		+1
	6x4 (PC)		+1
	5x7 (2L)		+2
	6x8 (A5)		+2
	6x9 (A5W)		+2
マルチカット	6x4x2	1 面目	---
	(PC 2 画面割付)	2 面目	+2

DP-QW410	プリントサイズ		カウンタ L/A/B	カウンタ P
シングルカット	4x3		+1	+1
	4x4		+1	+1
	4x4.5		+1	+1
	4x6		+1	+1
	4.5x3		+1	+1
	4.5x4		+1	+1
	4.5x4.5		+1	+1
	4.5x6		+1	+1
	4.5x8		+1	+1
マルチカット	4x3x2	1 枚目	---	+1
		2 枚目	+1	+1
	4.5x3x2	1 枚目	---	+1
		2 枚目	+1	+1
	4.5x4x2	1 枚目	---	+1
		2 枚目	+1	+1
2 インチカット	4x4, 4.5x4	1 枚目	---	+1
		2 枚目	+1	+1
	4x6, 4.5x6	1 枚目	---	+1
		2 枚目	---	+1
		3 枚目	+1	+1
	4.5x8	1 枚目	---	+1
		2 枚目	---	+1
		3 枚目	---	+1
		4 枚目	+1	+1

■両面
カウントアップするタイミングは、プリント画像が正常にカットされた後です。
エラーが発生したときはカウントアップされません。
カウンタ P は電源オフで 0 にクリアされます。

ロール紙を使用した場合

DP-DS80D	プリントサイズ		カウンタ L/A/B	カウンタ P	カウンタ Duplex
シングルカット	8x10		+1	+1	---
	8x12		+1	+1	---
	8x4		+1	+1	---
	8x5		+1	+1	---
	8x6		+1	+1	---
	8x8		+1	+1	---
	A4_Length		+1	+1	---
マルチカット	8x4x2	1 面目	---	+1	---
		2 面目	+1	+1	---
	8x5x2	1 面目	---	+1	---
		2 面目	+1	+1	---
	8x6x2	1 面目	---	+1	---
		2 面目	+1	+1	---
	8x5+8x4	1 面目	---	+1	---
		2 面目	+1	+1	---
	8x6+8x4	1 面目	---	+1	---
		2 面目	+1	+1	---
	8x6+8x5	1 面目	---	+1	---
		2 面目	+1	+1	---
	8x8+8x4	1 面目	---	+1	---
		2 面目	+1	+1	---
	8x4x3	1 面目	---	+1	---
		2 面目	---	+1	---
		3 面目	+1	+1	---

カット紙を使用した場合

DP-DS80D	プリントサイズ			カウンタ L/A/B	カウンタ P	カウンタ Duplex
シングルカット	8x10	片面		+1	+1	+1
		両面		+2	+1	+2
	8x12	片面		+1	+1	+1
		両面		+2	+1	+2
	8x4	片面		+1	+1	+1
		両面		+2	+1	+2
	8x5	片面		+1	+1	+1
		両面		+2	+1	+2
	8x6	片面		+1	+1	+1
		両面		+2	+1	+2
	8x8	片面		+1	+1	+1
		両面		+2	+1	+2
	8x10.5	片面		+1	+1	+1
		両面		+2	+1	+2
	8x10.75	片面		+1	+1	+1
		両面		+2	+1	+2
マルチカット	8x4x2	片面	1 面目	---	---	---
			2 面目	+1	+2	+1
		両面	1 面目	---	---	---
			2 面目	+2	+2	+2
	8x5x2	片面	1 面目	---	---	---
			2 面目	+1	+2	+1
		両面	1 面目	---	---	---
			2 面目	+2	+2	+2
	8x6x2	片面	1 面目	---	---	---
			2 面目	+1	+2	+1
		両面	1 面目	---	---	---
			2 面目	+2	+2	+2
	8x4x3	片面	1 面目	---	---	---
			2 面目	---	---	---
			3 面目	+1	+3	+1
		両面	1 面目	---	---	---
			2 面目	---	---	---
			3 面目	+2	+3	+2

拡張ライフカウンタ値の取得(DP-DS620)

【書式】 long GetCounterLifeEX(long lPortNum, DWORD dwMedia);

【引数】
lPortNum : ポート番号
dwMedia : 読み込み対象の指定

【戻り】
成功 : カウンタの値
失敗 : -1

【説明】 プリンターのライフカウンタの値を読み込みます。

読み込み対象の指定
COUNT_ALL : 全メディア合計のライフカウンタ
COUNT_SD : SD メディアで印刷したライフカウンタ
COUNT_PD : PD メディアで印刷したライフカウンタ
上記以外の値に設定した場合は、全メディア合計のライフカウンタの値が戻ります。

【文例】

```
< Visual C >
long counter;
if(( counter = GetCounterLifeEX ( PortNo, COUNT_ALL )) >= 0 ){
        // counter にライフカウンタ値
}

< Visual Basic >
Dim c As Integer
c = GetCounterLifeEX (PortNo, COUNT_ALL)
If c >= 0 Then Text1.Text = Str(c) Else Text1.Text = "ERROR!"
```

拡張マットカウンタ値の取得(DP-DS620)

【書式】 long GetCounterMatteEX(long lPortNum, DWORD dwMedia);

【引数】
lPortNum : ポート番号
dwMedia : 読み込み対象の指定

【戻り】
成功 : カウンタの値
失敗 : -1

【説明】 プリンターのマットカウンタの値を読み込みます。

読み込み対象の指定
COUNT_ALL : 全メディア合計のマットカウンタ
COUNT_SD : SD メディアで印刷したマットカウンタ
COUNT_PD : PD メディアで印刷したマットカウンタ
上記以外の値に設定した場合は、全メディア合計のマットカウンタの値が戻ります。

【文例】

```
< Visual C >
long counter;
if(( counter = GetCounterMatteEX ( PortNo, COUNT_ALL )) >= 0 ){
        // counter にマットカウンタ値
}

< Visual Basic >
Dim c As Integer
c = GetCounterMatteEX (PortNo, COUNT_ALL)
If c >= 0 Then Text1.Text = Str(c) Else Text1.Text = "ERROR!"
```

カウンタ値のクリア

【書式】

```
BOOL SetClearCounterA( long lPortNum );
BOOL CvSetClearCounterA( long lPortNum );

BOOL SetClearCounterB( long lPortNum );
BOOL CvSetClearCounterB( long lPortNum );

BOOL SetClearCounterM( long lPortNum );
```

【引数】

lPortNum：　ポート番号

【戻り】

成功：　TRUE
失敗：　FALSE

【説明】

SetClearCounterA(),CvSetClearCounterA()は、カウンタ A の値をクリアします。
SetClearCounterB(),CvSetClearCounterB()は、カウンタ B の値をクリアします。
SetClearCounterM()は、カウンタ M の値をクリアします。
カウンタ M は、マット印刷対応のプリンター(DS40/80 Ver.1.30 以降、DS40T/80T Ver.0.31 以降.)で有効です。

【文例】

```
< Visual C >
if( SetClearCounterA( PortNo )){
        // カウンタ A クリア成功
}

< Visual Basic >
If SetClearCounterA(PortNo) <> False Then
        'カウンタ A クリア成功
End If
```

カウンタ値（P）の設定

【書式】

```
BOOL SetCounterP( long lPortNum、long lCounter );
```

【引数】

lPortNum:　ポート番号
lCounter:　設定するカウンタ値

【戻り】

成功:　TRUE
失敗:　FALSE

【説明】

カウンタ P に任意の値をセットします。プリンターの電源切ると 0 に初期化されます。
負の値を指定した場合は、0 が設定されます。

【文例】

```
< Visual C >
if( SetCounterP( PortNo, 100)){
        // カウンタ値 P を設定
}

< Visual Basic >
If SetCounterP(PortNo, 100) <> False Then
        'カウンタ値 P を設定
End If
```

空きバッファ数の取得

【書式】
```
long GetFreeBuffer( long lPortNum );
long CvGetFreeBuffer( long lPortNum );
```

【引数】
lPortNum：　ポート番号

【戻り】
成功：　印画バッファの空き数
失敗：　-1

【説明】
印画バッファの空き数を返します。

【文例】
```
< Visual C >
long bn;
if(( bn = GetFreeBuffer( PortNo )) >= 0 ){
        // bn に空きバッファ数
}

< Visual Basic >
Dim bn As Long
bn = GetFreeBuffer(PortNo)
```

残り発行枚数の取得

【書式】
```
long GetPQTY( long lPortNum );
long CvGetPQTY( long lPortNum );
```

【引数】
lPortNum：　ポート番号

【戻り】
成功：　残り発行枚数
失敗：　-1

【説明】
残り発行枚数を返します。

【文例】
```
< Visual C >
long number;
if(( number = GetPQTY( PortNo )) >= 0 ){
        // number に残り発行枚数
}

< Visual Basic >
Dim number As Long
number = GetPQTY(PortNo)
```

メディア残り発行枚数の取得

【書式】 long GetMediaCounter(long lPortNum);
long CvGetMediaCounter(long lPortNum);

【引数】 lPortNum : ポート番号

【戻り】 成功 : メディア残り発行枚数
失敗 : -1

機種	メディアサイズ	印刷可能枚数	メディア発行枚数初期値
CV	5x3.5 (L)	400	500
	6x4 (PC)	400	450
	5x7 (2L)	230	280
	6x9 (A5W)	180	230
CW CW02	5x3.5 (L)	660	710
	6x4 (PC)	600	650
	5x7 (2L)	350	400
	6x8 (A5)	300	350
	6x9 (A5W)	280	330
DS40 DS40T	5x3.5 (L)	400	450
	6x4 (PC)	400	450
	5x7 (2L)	230	280
	6x8 (A5)	200	250
	6x9 (A5W)	180	230
DS-RX1	5x3.5 (L)	700	750
	6x4 (PC)	700	750
		350	400
	5x7 (2L)	350	400
	6x8 (A5)	350	400
DS80 DS80T DP-DS80D	8x10	130	180
	8x12	110	160
DP-TC10	TC86x61	600	650
	TC127x61	600	650
DP-DS620	5x3.5 (L) *2	420	420
	6x4 (PC)	400	400
	5x7 (2L)	230	230
	6x8 (A5)	200	200
	6x9 (A5W) *1	180	180
DP-DS820	8x10	130	130
	8x12	110	110
	A4	110	110
DP-QW410	4x6	150	150
	4.5x6 *3	150	150
	4.5x8	110	110

*1 DP-DS620 ファームウェアバージョン 1.10 以降
*2 DP-DS620 ファームウェアバージョン 1.60 以降
*3 DP-QW410 ファームウェアバージョン 1.07 以降

メディア交換等により用紙初期化が実行された場合、機種によってメディア残り発行枚数を消費するものがあります。消費の有無については下記を参照してください。

メディアの状態	DP-TC10(ver.1.20 のみ) DP-DS620/DP-DS820	DP-QW410	その他
未使用メディア	0	0	-1
使用途中のメディア	0	-1	-1

【説明】 メディア残り発行枚数を返します。
表の通り、残り印画可能枚数よりも発行枚数が多い初期値になっています。
（DP-QW410 はこの値が 0 でリボンエンドとなります。DP-DS620,DP-DS820 はこの値が 0 でリボンエンドとなりますが、その前にリボンエンドチェック動作が入ります。詳細は「(4) DP-DS620、DP-DS820　リボンエンドチェック動作」参照）

【文例】

```
< Visual C >
long number;
if(( number = GetMediaCounter( PortNo )) >= 0 ){
        // number にメディア残り発行枚数
}
< Visual Basic >
Dim number As Long
number = GetMediaCounter(PortNo)
```

ハーフサイズ換算メディア残り発行枚数の取得（DP-DS620, DP-DS820）

【書式】 long GetMediaCounterH(long lPortNum);

【引数】 lPortNum : ポート番号

【戻り】 成功 : 装着リボンのハーフサイズで換算したメディア残り発行枚数
失敗 : -1

機種	メディアサイズ	ハーフサイズ	ハーフサイズ換算メディア発行枚数初期値
DP-DS620	5x7 (2L)	5x3.5	460
	6x8 (A5)	6x4	400
	6x9 (A5W) *1	6x4.5	360
DP-DS820	8 x 10	8x5	260
	8 x 12	8x6	220
	A4	A5	220

*1 DP-DS620 ファームウェアバージョン 1.10 以降

【説明】 装着リボンのハーフサイズで換算した残り発行枚数を返します。

【文例】

```
< Visual C >
long number;
if(( number = GetMediaCounterH( PortNo )) >= 0 ){
        // number にメディア残り発行枚数
}

< Visual Basic >
Dim number As Long
number = GetMediaCounterH(PortNo)
```

メディアロット間色バラツキ補正値の取得

【書式】
```
long GetMediaColorOffset( long lPortNum );
long CvGetMediaColorOffset( long lPortNum );
```

【引数】
lPortNum：　ポート番号

【戻り】
成功：　メディアロット間色バラツキ補正値
失敗：　-1

【説明】
リボンのロット間での色バラツキ補正値を返します。

【文例】
```
< Visual C >
long offset;
if(( offset = GetMediaColorOffset( PortNo )) >= 0 ){
        // offset に色補正値
}

< Visual Basic >
Dim offset As Long
offset = GetMediaColorOffset(PortNo)
```

【例】
offset＝169082895（10 進）の場合、0x0A14000F（16 進）となり、各色の補正値は以下のようになります。

色	補正値
Yellow	10(0x0A)
Magenta	20(0x14)
Cyan	0(0x00)
Op	15(0x0F)

メディアタグ情報の取得

【書式】
```
long GetMediaLotNo( long lPortNum, LPSTR p );
long CvGetMediaLotNo( long lPortNum, LPSTR p );
```

【引数】
lPortNum：　ポート番号
p：　受信バッファへのポインタ

【戻り】
成功：　バッファ p へ受信した文字数
失敗：　-1

【説明】
メディアの RFID タグに書かれたユーザー情報を返します。

【文例】
```
< Visual C >
char rbuf[256];
if( GetMeidaLotNo( PortNo, (LPSTR)rbuf ) > 0 ){
        // 処理
}

< Visual Basic >
Dim rb As String * 255, Ver As String
Dim n As Long
n = GetMediaLotNo(PortNo, rb)
If n > 0 Then Ver = Left(rb, n)        '文字列を切出す場合
```

【メディアタグユーザー情報】
ML**********　ユーザー専用情報＜16Byte＞

プリンターシリアルナンバーの取得

【書式】
```
long GetSerialNo( long lPortNum, LPSTR p );
long CvGetSerialNo( long lPortNum, LPSTR p );
```

【引数】
lPortNum：　ポート番号
p：　受信バッファへのポインタ

【戻り】
成功：　バッファ p へ受信した文字数
失敗：　-1

【説明】
プリンターのシリアル No（製造番号）を取得します。

【文例】
< Visual C >
```
char rbuf[256];
if( GetSerialNo( PortNo, (LPSTR)rbuf ) > 0 ){
        // 処理
}
```

< Visual Basic >
```
Dim rb As String * 255, Ver As String
Dim n As Long
n = GetSerialNo(PortNo, rb)
If n > 0 Then Ver = Left(rb, n)        '文字列を切出す場合
```

【シリアル No　例】
DS4Xnnnnnnnn

ファームウェア書き換えモードへの切り替えコマンド

【書式】
```
BOOL SetFirmwUpdateMode( long lPortNum );
BOOL CvSetFirmwUpdateMode( long lPortNum );
```

【引数】
lPortNum：　ポート番号

【戻り】
成功：　TRUE
失敗：　FALSE

【説明】
プリンターをファームウェア書き換えモードに切り替えます。
詳細は「11. ファームウェアと制御データアップデート手順について」を参照して下さい。

【文例】
< Visual C >
```
if( SetFirmwUpdataMode(PortNo) == TRUE ){
        // 処理
}
```

< Visual Basic >
```
If ( SetFirmwUpdateMode(PortNo) = True ) Then
        '処理
End If
```

ファームウェア書き換えデータの送信コマンド

【書式】

```
BOOL SetFirmwDataWrite( long lPortNum, LPSTR lpData, DWORD dwDataLen );
BOOL CvSetFirmwDataWrite( long lPortNum, LPSTR lpData, DWORD dwDataLen );
```

【引数】

lPortNum :	ポート番号
lpData :	書き換えデータが格納されたバッファのポインタ
dwDataLen :	データの文字数

【戻り】

成功 :	TRUE
失敗 :	FALSE

【説明】

プリンターのファームウェア書き換えデータを転送します。
書き換えデータはファイルで提供されるモトローラ S フォーマットです。本コマンド使用時はファイルを読み込み、書き換えデータを一時格納するバッファ（10Mbyte 程度）が必要です。
書き換えが完了すると、プリンターは自動的に再起動して、書き換えモードは終了します。
詳細は「11. ファームウェアと制御データアップデート手順について」を参照して下さい。

【文例】

< Visual Basic >

```
Dim fd(10000000)
Dim c As Long, n As Long

C=0
Open fname For Binary Access Read As
#1 FileLength = LOF(1)
For n = 0 To FileLength - 1
        Get #1, , fd(c)
        c = c + 1
Next n
Close #1

SetFirmwDataWrite( PortNo, VarPtr(fd(0)), c )
```

反転ユニットファームウェア書き換えモードへの切り替えコマンド（DP-DS80D）

【書式】 BOOL SetDunitFirmwUpdateMode(long lPortNum, DWORD dwBaudRate);

【引数】
lPortNum： ポート番号
dwBaudRate: ボーレート設定
0: 38400 bps
1: 57600 bps
2: 115200 bps
3: 230400 bps

【戻り】
成功： TRUE
失敗： FALSE

【説明】
プリンターを反転ユニットファームウェア書き換えモードに切り替えます。
詳細は「11. ファームウェアと制御データアップデート手順について」を参照して下さい。
アップデートに失敗する場合は 2：115200 bps を試して下さい。

【文例】

```
< Visual C >
if( SetDunitFirmwUpdataMode(PortNo,3) == TRUE ){
        // 処理
}

< Visual Basic >
If ( SetDunitFirmwUpdateMode(PortNo,3) = True ) Then
        '処理
End If
```

反転ユニットファームウェア書き換えデータの送信コマンド（DP-DS80D）

【書式】 BOOL SetDunitFirmwDataWrite(long lPortNum, LPSTR lpData, DWORD dwDataLen);

【引数】
lPortNum： ポート番号
lpData： 書き換えデータが格納されたバッファのポインタ
dwDataLen： データの文字数

【戻り】
成功： TRUE
失敗： FALSE

【説明】
反転ユニットのファームウェア書き換えデータを転送します。
書き換えデータはファイルで提供されるバイナリーフォーマットです。本コマンド使用時はファイルを読み込み、書き換えデータを一時格納するバッファ（10Mbyte 程度）が必要です。
書き換えが完了すると、プリンターは自動的に再起動して、書き換えモードは終了します。
詳細は「11. ファームウェアと制御データアップデート手順について」を参照して下さい。

【文例】

```
< Visual Basic >
Dim fd(10000000)
Dim c As Long, n As Long

C=0
Open fname For Binary Access Read As
#1 FileLength = LOF(1)
For n = 0 To FileLength - 1
        Get #1, , fd(c)
        c = c + 1
Next n
Close #1

SetDunitFirmwDataWrite( PortNo, VarPtr(fd(0)), c )
```

反転ユニットファームウェア書換え状態の取得（DP-DS80D）

【書式】　long GetDunitUpdateStatus(long lPortNum, LPSTR p);

【引数】　lPortNum：　ポート番号
p：　受信バッファへのポインタ

【戻り】　成功：　バッファ p へ受信した文字数
失敗：　-1

【説明】　ファームウェア書換え情報を文字列バッファへ受信します。

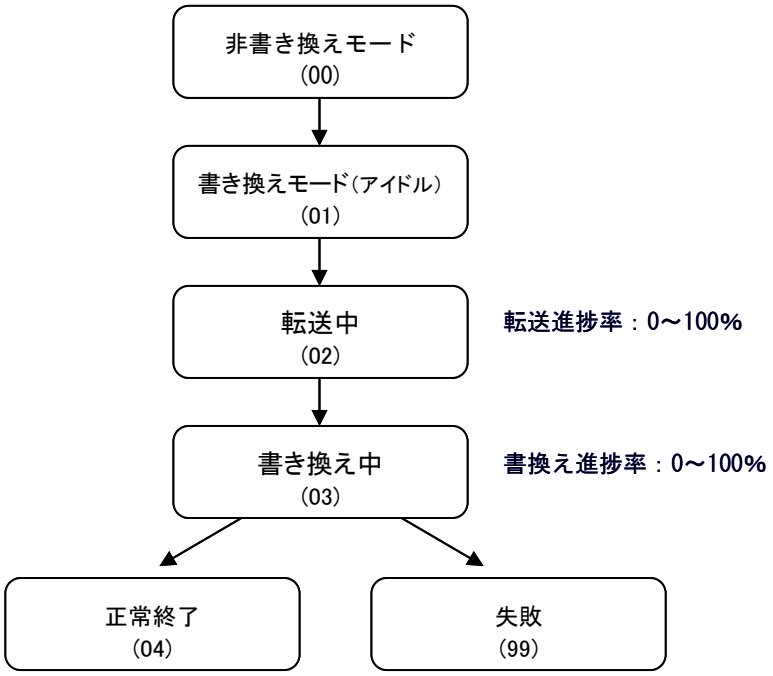
【文例】

```
< Visual C >
char rbuf[256];
if(GetDunitUpdateStatus( PortNo, (LPSTR)rbuf ) > 0 ){
        // 処理
}

< Visual Basic >
Dim rb As String * 255, Sen As String
Dim n As Long
n = GetDunitUpdateStatus(PortNo, rb)
If n > 0 Then Sen = Left(rb, n)        '文字列を切出す場合
```

【ファームウェア書換え情報】

STS-**;　書換えステータス
- 00: 非書き換えモード
- 01: 反転ユニットファームウェア書き換えモード　（アイドル）
- 02: 反転ユニットファームウェア転送中
- 03: 反転ユニットファームウェア書き換え中
- 04: 反転ユニットファームウェア書き換え 正常終了
- 99: 反転ユニットファームウェア書き換え 失敗

PRG-***;　進捗情報（000～100）
EXT-********;　拡張情報（00000000～99999999）



反転ユニットファームウェア書き換えモード ステータス遷移図

反転ユニットエラーログ作成要求（DP-DS80D）

【書式】 BOOL DunitErrorLogCreateReq(long lPortNum, DWORD dwBaudRate);

【引数】
lPortNum： ポート番号
dwBaudRate: ボーレート設定
0: 38400 bps
1: 57600 bps
2: 115200 bps

【戻り】
成功： TRUE
失敗： FALSE

【説明】 プリンターに反転ユニットのエラーログの作成を要求します。
詳細は「P69 反転ユニットエラーログデータ取得手順」を参照して下さい。

【文例】

```
< Visual C >
if( DunitErrorLogCreateReq(PortNo,2) == TRUE ){
        // 処理
}

< Visual Basic >
If ( DunitErrorLogCreateReq(PortNo,2) = True ) Then
        '処理
End If
```

反転ユニットエラーログデータ長の取得（DP-DS80D）

【書式】 long GetDunitErrorLogDataLength(long lPortNum);

【引数】
lPortNum： ポート番号

【戻り】
成功： データ長
失敗： -1

【説明】 反転ユニットのエラーログのデータ長を取得します。
詳細は「P69 反転ユニットエラーログデータ取得手順」を参照して下さい。

【文例】

```
< Visual Basic >
Dim length As Long

Length = GetDunitErrorLogDataLength( PortNo )
```

反転ユニットエラーログデータ受信（DP-DS80D）

【書式】 long GetDunitErrorLogData(long lPortNum, LPSTR lpData, DWORD dwBufSize);

【引数】
lPortNum： ポート番号
lpData： ログデータを受信するバッファのポインタ
dwBufSize： アプリケーションが確保した受信のバッファーサイズ

【戻り】
成功： データ長
失敗： -1

【説明】
反転ユニットのエラーログデータを受信します。
エラーログデータはCSV形式のテキストフォーマットです。本コマンド使用時は
エラーログデータを一時格納するバッファ（10Kbyte程度）が必要です。
受信が完了すると、プリンターはエラーログ取得モードを終了します。
詳細は「P69 反転ユニットエラーログデータ取得手順」を参照して下さい。

【文例】
```
< Visual Basic >
Dim fd(10000)
Dim c As Long, n As Long

GetDunitErrorLogData( PortNo, VarPtr(fd(0)), 10000 )
```

反転ユニットエラーログ作成状態の取得（DP-DS80D）

【書式】 long GetDunitErrorLogStatus(long lPortNum, LPSTR p, DWORD dwBufSize);

【引数】
lPortNum： ポート番号
p： 受信バッファへのポインタ
dwBufSize： アプリケーションが確保した受信のバッファーサイズ

【戻り】
成功： バッファ p へ受信した文字数
失敗： -1

【説明】
エラーログ作成状態情報を文字列バッファへ受信します。

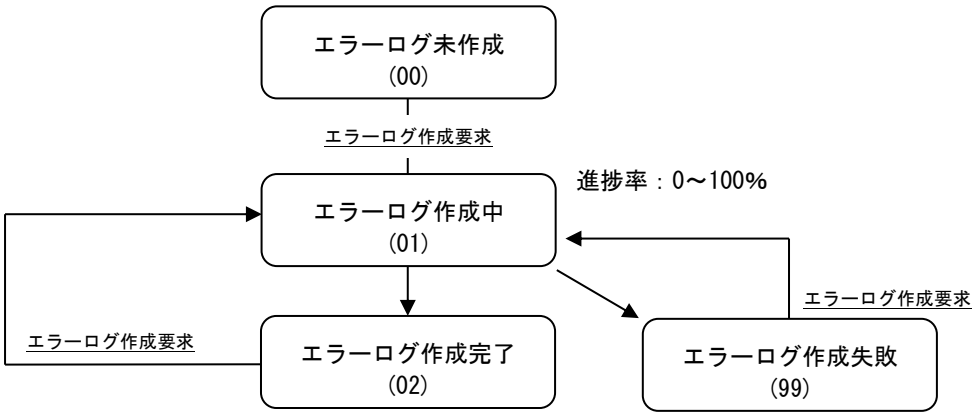
【文例】
```
< Visual C >
char rbuf[256];
if(GetDunitErrorLogStatus( PortNo, (LPSTR)rbuf,256 ) > 0 ){
        // 処理
}

< Visual Basic >
Dim rb As String * 255, Sen As String
Dim n As Long
n = GetDunitErrorLogStatus(PortNo, rb, 256)
If n > 0 Then Sen = Left(rb, n)        '文字列を切出す場合
```

【エラーログ作成状態情報】

STS-**;　　書換えステータス
- 00: 反転ユニットエラーログ未作成
- 01: 反転ユニットエラーログ作成中
- 02: 反転ユニットエラーログ作成完了
- 99: エラー

PRG-***;　　進捗情報（000～100）

EXT-********;　　拡張情報（00000000～99999999）
- 00005401: 反転ユニットログデータ転送準備エラー
- 00005402: 反転ユニットログ転送バイト数受信エラー
- 00005403: 反転ユニットログデータ受信エラー
- 00005404: 反転ユニットログデーターオーバーフロー
- 00005405: 反転ユニットログ転送終了受信エラー
- 00005406: 反転ユニットログ転送モード終了エラー
- 00005407: タイムアウトエラー
- 00005420: その他のエラー



反転ユニットエラーログモードステータス遷移図

反転ユニットエラーログデータ取得手順

お客様が API を使用してエラーログを取得する時は、次の手順で行って下さい。

API 使用例）		参照頁
1　DunitErrorLogCreateReq()	反転ユニットエラーログ作成要求	P67
2　GetDunitErrorLogStatus()	反転ユニットエラーログ作成状態の取得	P68
3　GetDunitErrorLogDataLength()	反転ユニットエラーログデータ長の取得	P67
4　GetDunitErrorLogData()	反転ユニットエラーログデータの受信	P68

色制御データクリアコマンド

【書式】
```
BOOL SetColorDataClear( long lPortNum );
BOOL CvSetColorDataClear( long lPortNum );
```

【引数】
lPortNum：　ポート番号

【戻り】
成功：　TRUE
失敗：　FALSE

【説明】
プリンターに書き込まれている色制御データをクリアします。
色制御データを書き換える場合には、最初にこのコマンドで既存の色制御データをクリアしてください。
詳細は「11. ファームウェアと制御データアップデート手順についてについて」を参照して下さい。

【文例】
```
< Visual C >
if(SetColorDataClear( PortNo )){
        //色制御データクリア成功
}

< Visual Basic >
If SetColorDataClear(PortNo) <> False Then
'色制御データクリア成功
EndIf
```

色制御データの送信コマンド

【書式】
```
BOOL SetColorDataWrite( long lPortNum, LPSTR lpData, DWORD dwDataLen  );
BOOL CvSetColorDataWrite( long lPortNum, LPSTR lpData, DWORD dwDataLen  );
```

【引数】
lPortNum：　ポート番号
lpData：　色制御データが格納されたバッファのポインタ
dwDataLen：　データの文字数

【戻り】
成功：　TRUE
失敗：　FALSE

【説明】
色制御データを転送します。
このコマンドで新しい色制御データを転送する前に、色制御データクリアコマンドで既存の色制御データをクリアしてください。

色制御データは独自フォーマットのバイナリファイルで提供されます。本コマンド使用時はファイルを読み込み、書き換えデータを一時格納するバッファ（10Mbyte 程度）が必要です。
詳細は「11. ファームウェアと制御データアップデート手順について」を参照して下さい。

【文例】
```
< Visual Basic >
Dim fd(10000000)
Dim c As Long, n As Long

C=0
Open fname For Binary Access Read As #1
FileLength = LOF(1)
For n = 0 To FileLength - 1
        Get #1, , fd(c)
        c = c + 1
Next n
Close #1

SetColorDataWrite( PortNo, VarPtr(fd(0)), c )
```

色制御データバージョン設定コマンド

【書式】
```
BOOL SetColorDataVersion( long lPortNum, LPSTR lpData, DWORD dwDataLen );
BOOL CvSetColorDataVersion( long lPortNum, LPSTR lpData, DWORD dwDataLen );
```

【引数】

lPortNum :	ポート番号
lpData :	色制御データバージョンが格納されたバッファのポインタ
dwDataLen :	データの文字数

【戻り】

成功 :	TRUE
失敗 :	FALSE

【説明】

色制御データのバージョンを設定します。
色制御データ送信コマンドで新しい色制御データを転送した後、このコマンドで色制御データのバージョンを設定してください。

設定する色制御データのバージョンは、提供される色制御データファイルのファイル名としてください。
詳細は「11. ファームウェアと制御データアップデート手順について」を参照して下さい。

【文例】

```
< Visual Basic >
Dim fname As String
'fname へバージョン情報文字列を格納
SetColorDataVersion( PortNo, fname, Len(fname) )
```

色制御データバージョン取得コマンド

【書式】
```
long GetColorDataVersion( long lPortNum, LPSTR p );
long CvGetColorDataVersion( long lPortNum, LPSTR p );
```

【引数】

lPortNum :	ポート番号
p :	受信バッファへのポインタ

【戻り】

成功 :	バッファ p へ受信した文字数
失敗 :	-1

【説明】

色制御データのバージョン情報を文字列バッファへ受信します。
詳細は「11. ファームウェアと制御データアップデート手順について」を参照して下さい。

【文例】

```
< Visual C >
char rbuf[256];
if(GetColorDataVersion( PortNo, (LPSTR)rbuf ) > 0 ){
        // 処理
}

< Visual Basic >
Dim rb As String * 255, Ver As String
Dim n As Long
n = GetColorDataVersion(PortNo, rb)
If n > 0 Then Ver = Left(rb, n)        '文字列を切出す場合
```

色制御データバージョン取得コマンド〈種別指定〉(DP-DS620, DP-DS820, DP-QW410)

【書式】 long GetColorDataVersionRes(long lPortNum, LPSTR p, long lType);

【引数】
lPortNum : ポート番号
p : 受信バッファへのポインタ
lType: 種別 設定値 : 0 - 4

【戻り】
成功 : バッファ p へ受信した文字数
失敗 : -1

【説明】
指定された種別の色制御データのバージョン情報を文字列バッファへ受信します。
詳細は「11. ファームウェアと制御データアップデート手順について」を参照して下さい。

機種毎の取得データと有効な設定値

取得値	設定値		
	DP-DS620	DP-DS820	DP-QW410
300dpi	0	0	0
600dpi	1	1	(非対象)
低速	2	2	4
高濃度	(非対象)	3	(非対象)

【文例】

```
< Visual C >
long lType;
char rbuf[256];
if(GetColorDataVersionRes( PortNo, (LPSTR)rbuf, lType ) > 0 ){
        // 処理
}
< Visual Basic >
Dim lType As Long
Dim rb As String * 255, Ver As String
Dim n As Long
n = GetColorDataVersionRes(  PortNo, rb, lType  )
If n > 0 Then Ver = Left(rb, n)        '文字列を切出す場合
```

色制御データバージョン取得コマンド<種別、メディア指定> (DP-DS620, DP-DS820, DP-QW410)

【書式】 long GetColorDataVersionResEX(long lPortNum, LPSTR p, long lType, DWORD dwMedia);

【引数】

lPortNum :	ポート番号	
p :	受信バッファへのポインタ	
lType:	種別	設定値 : 0 - 4
dwMedia	メディア指定	設定値 : 1(SD)、2(PD)、3(PP)

【戻り】

成功 :	バッファ p へ受信した文字数
失敗 :	-1

【説明】 色制御データのバージョン情報を文字列バッファへ受信します。
詳細は「11. ファームウェアと制御データアップデート手順について」を参照して下さい。

機種毎の取得データと有効な種別設定値

取得値	設定値		
	DP-DS620	DP-DS820	DP-QW410
300dpi	0	0	0
600dpi	1	1	(非対象)
低速	2	2	4
高濃度	(非対象)	3	(非対象)

メディア指定に‘1’が設定された場合は、SD メディアの色制御データを取得します。
メディア指定に‘2’が設定された場合は、PD メディアの色制御データを取得します。 *1
メディア指定に‘3’が設定された場合は、PP メディアの色制御データを取得します。 *2
*1 : DP-QW410、及び、DP-DS620(Ver.3.00 以降)
*2 : DP-DS820 のみ

【文例】

```
< Visual C >
char rbuf[256];
if(GetColorDataVersionResEX( PortNo, (LPSTR)rbuf, 1, 3 ) > 0 ){ // 600dpi のバージョンを取得
        // 処理
}
< VB.NET >
Dim s As String = New String("", 255)
Dim i As Integer
i = GetColorDataVersionResEX(PortNo, s, 0, 3) ' 300dpi のバージョンを取得
If i > 0 Then Text1.Text = VB.Left(s, i) Else Text1.Text = "ERROR!"
```

色制御データバージョンをテンポラリーへ設定（CW）

【書式】 BOOL SetColorDataVersionTmp(long lPortNum, LPSTR lpData, DWORD dwDataLen);

【引数】
lPortNum： ポート番号
lpData： 色制御データバージョンが格納されたバッファのポインタ
dwDataLen： データの文字数

【戻り】
成功： TRUE
失敗： FALSE

【説明】 プリンターの RAM（テンポラリー）の色制御データバージョンを設定します。
色制御データをテンポラリーへ転送する際、このコマンドで色制御データのバージョンを設定して下さい。
設定する色制御データのバージョンは、提供される色制御データファイルのファイル名として下さい。

【文例】 文例は SetColorDataWriteTmp を参照して下さい。

色制御データをテンポラリーへ送信（CW）

【書式】 BOOL SetColorDataWriteTmp(long lPortNum, LPSTR lpData, DWORD dwDataLen);

【引数】
lPortNum： ポート番号
lpData： 色制御データが格納されたバッファのポインタ
dwDataLen： データの文字数

【戻り】
成功： TRUE
失敗： FALSE

【説明】 色制御データをプリンターの RAM（テンポラリー）に送信します。色制御データを送信すると前のデータに上書きをします。

本コマンドはＣＷ機で色制御データ書き換えによる、プリンターの解像度切り替えを頻繁に行う時に使用します。テンポラリーはプリンターの電源 OFF で消去され、再起動時はデフォルト値で起動されます。（デフォルト値は SetColorDataWrite（）にて書き換えが可能です。）

色制御データは独自フォーマットのバイナリファイルで提供されます。本コマンド使用時は
ファイルを読み込み、書き換えデータを一時格納するバッファ（10Mbyte 程度）が必要です。

【文例】

```
< Visual Basic >
Dim fd(10000000)
Dim c As Long, n As Long

C=0
Open fname For Binary Access Read As #1

FileLength = LOF(1)
For n = 0 To FileLength - 1
        Get #1, , fd(c)
        c = c + 1
Next n
Close #1

SetColorDataVersionTmp( PortNo, fname, Len(fname) )
SetColorDataWriteTmp( PortNo, VarPtr(fd(0)), c )
```

色制御データチェックサム取得コマンド

【書式】
```
long GetColorDataChecksum( long lPortNum, LPSTR p );
long CvGetColorDataChecksum( long lPortNum, LPSTR p );
```

【引数】
lPortNum：　ポート番号
p：　受信バッファへのポインタ

【戻り】
成功：　バッファ p へ受信した文字数
失敗：　-1

【説明】
色制御データのチェックサムを文字列バッファへ受信します。
詳細は「11. ファームウェアと制御データアップデート手順について」を参照して下さい。

【文例】
```
< Visual C >
char rbuf[256];
if(GetColorDataChecksum( PortNo, (LPSTR)rbuf ) > 0 ){
        // 処理
}

< Visual Basic >
Dim rb As String * 255, Ver As String
Dim n As Long
n = GetColorDataChecksum(PortNo, rb)
If n > 0 Then Ver = Left(rb, n)        '文字列を切出す場合
```

色制御データチェックサム取得コマンド〈種別指定〉(DP-DS620, DP-DS820, DP-QW410)

【書式】
```
long GetColorDataChecksumRes( long lPortNum, LPSTR p, long lType );
```

【引数】
lPortNum：　ポート番号
p：　受信バッファへのポインタ
lType:　種別　設定値：0 - 4

【戻り】
成功：　バッファ p へ受信した文字数
失敗：　-1

【説明】
指定された種別の色制御データのチェックサムを文字列バッファへ受信します。
詳細は「11. ファームウェアと制御データアップデート手順について」を参照して下さい。

機種毎の取得データと有効な設定値

取得値	設定値		
	DP-DS620	DP-DS820	DP-QW410
300dpi	0	0	0
600dpi	1	1	(非対象)
低速	2	2	4
高濃度	(非対象)	3	(非対象)

【文例】
```
< Visual C >
long lType;
char rbuf[256];
if(GetColorDataChecksumRes( PortNo, (LPSTR)rbuf, lType ) > 0 ){
        // 処理
}

< Visual Basic >
Dim lType As Long
Dim rb As String * 255, Ver As String
Dim n As Long
n = GetColorDataChecksumRes(PortNo, rb, lType)
If n > 0 Then Ver = Left(rb, n)        '文字列を切出す場合
```

色制御データチェックサム取得コマンド〈種別、メディア指定〉(DP-DS620, DP-DS820, DP-QW410)

【書式】 long GetColorDataChecksumResEX(long lPortNum, LPSTR p, long lType, DWORD dwMedia);

【引数】
lPortNum： ポート番号
p： 受信バッファへのポインタ
lType: 種別 設定値：0 - 4
dwMedia メディア指定 設定値：1(SD) 、2(PD)、3(PP)

【戻り】
成功： バッファ p へ受信した文字数
失敗： -1

【説明】
指定された種別の色制御データのチェックサムを文字列バッファへ受信します。
詳細は「11. ファームウェアと制御データアップデート手順について」を参照して下さい。

機種毎の取得データと有効な設定値

取得値	設定値		
	DP-DS620	DP-DS820	DP-QW410
300dpi	0	0	0
600dpi	1	1	(非対象)
低速	2	2	4
高濃度	(非対象)	3	(非対象)

メディア指定に‘1’が設定された場合は、SD メディアのチェックサムを取得します。
メディア指定に‘2’が設定された場合は、PD メディアのチェックサムを取得します。 *1
メディア指定に‘3’が設定された場合は、PP メディアのチェックサムを取得します。 *2
*1：DP-QW410、及び、DP-DS620(Ver.3.00 以降)
*2：DP-DS820 のみ

【文例】

```
< Visual C >
char rbuf[256];
if(GetColorDataChecksumResEX( PortNo, (LPSTR)rbuf, 1, 3) > 0 ){
        // 処理
}

< Visual Basic >
Dim rb As String * 255, Ver As String
Dim n As Long
n = GetColorDataChecksumResEX(PortNo, rb, 1, 3)
If n > 0 Then Ver = Left(rb, n)        '文字列を切出す場合
```

カッター制御コマンド

【書式】 BOOL SetCutterMode(long lPortNum, DWORD ctMode);

【引数】
lPortNum : ポート番号
ctMode : カッターのモード選択

【戻り】
成功 : TRUE
失敗 : FALSE

【説明】
カッター動作を指定します。
カッターの動作指定は CspStat.h にマクロ定義されています。
シンボルの意味は下記の通りです。

CUTTER_MODE_STANDARD	(0)	通常のカット動作
CUTTER_MODE_NONSCRAP	(1)	カットくず無し指定
CUTTER_MODE_SINGLECUT	(2)	シングルカット指定(DP-TC10)
CUTTER_MODE_NONSCRAP_1	(100)	2 面付、両方カットくず無し指定 ※4
CUTTER_MODE_NONSCRAP_2	(101)	2 面付、1 枚目カットくず無し指定 ※4
CUTTER_MODE_NONSCRAP_3	(102)	2 面付、2 枚目カットくず無し指定 ※4
CUTTER_MODE_2INCHCUT	(120)	2 インチカット指定 ※1
CUTTER_MODE_L2IMAGEPRINT	(130)	L カードサイズ 2 画面連続印画指定(DP-TC10 のみ) ※2
CUTTER_MODE_PANORAMA	(1000)	パノラマ印画動作(画像間余白あり) ※3

注) カッター制御コマンドは、画像データの送信前に動作指定します。コマンドは 1 画像に対して 1 回有効です。プリンターは印画⇒指定カット動作後、通常のカット動作に戻ります。
2 インチカット指定は以下の用紙サイズに対して有効です。
DS40/DS-RX1 : 6x4、6x8
DP-DS620 : 6x4、6x8、6x4x2
DP-DS820 : 8x4、8x6、8x8、8x10、8x12、A4x6、A4x8、A4x10
DP-QW410 : 4x4、4x6、4.5x4、4.5x6、4.5x8、4.5x4x2
パノラマ印刷は以下の用紙サイズに対応しています。
DP-DS620 : 6x16(6x8 2 枚印画)、6x24(6x8 3 枚印画)
DP-DS820 : 8x20(8x10 2 枚印画)、8x24(8x12 2 枚印画)、A4x2(A4 2 枚印画)、8x30(8x10 3 枚印画)、8x24(8x12 3 枚印画)、A4x3(A4 3 枚印画)

※1 下表の機種・ファームウェアバージョンで有効

機種	DS40	DS-RX1	DP-DS620	DP-DS820	DP-QW410
ファームウェア Ver.	1.40 以上	1.10 以上	○	○	○

※2 DP-TC10 の L カードサイズ 2 画面連続印画の詳細については、
「13. 補足資料 (2) DP-TC10 L カードサイズ 2 画面連続印画について」を参照してください。

※3 DP-DS620 ファームウェアバージョン 01.20 以降対応 又は DP-DS820
パノラマ印画については、
「13. 補足資料 (5) 余白ありパノラマ印画」を参照してください。

※4 DP-DS620 ファームウェアバージョン 04.00 以降対応
2 面付系カットくず無し指定パラメータ(CUTTER_MODE_NONSCRAP_1, CUTTER_MODE_NONSCRAP_2, CUTTER_MODE_NONSCRAP_3)を選択した場合のカットくず無し動作は、下表のようになります。

指定値	2 面付		1 面付
	1 枚目印画物	2 枚目印画物	
CUTTER_MODE_NONSCRAP_1	カットくず無し動作	カットくず無し動作	カットくず無し動作
CUTTER_MODE_NONSCRAP_2	カットくず無し動作	通常動作	カットくず無し動作
CUTTER_MODE_NONSCRAP_3	通常動作	カットくず無し動作	カットくず無し動作

【文例】

< Visual C >
```
SetCutterMode(PortNo, (DWORD)CUTTER_MODE_NONSCRAP);
```

< Visual Basic >
```
SetCutterMode(PortNo, CUTTER_MODE_NONSCRAP)
```

メディア ID 設定情報取得

【書式】 long GetMediaIdSetInfo(long lPortNum);

【引数】 lPortNum : ポート番号

【戻り】 成功 : メディア ID 設定情報
失敗 : -1

機種	戻り値	メディア ID 設定内容
CW, CWO2	0	ID 未設定
	1	CW 標準 ID
	2	CW-PMI ID
その他	0	

【説明】 メディア ID 設定情報を返します。

【文例】

```
< Visual C >
long idval;
if(( idval = GetMediaIdSetInfo( PortNo )) >= 0 ){
        // idval にメディア ID 設定情報
}

< Visual Basic >
Dim idval As Long
idval = GetMediaIdSetInfo(PortNo)
```

一枚画像保持印画　制御コマンド (DS40)

【書式】 BOOL SetRetentionMode(long lPortNum, DWORD rtMode);

【引数】 lPortNum : ポート番号
rtMode : 一枚画像保持印画モードの選択

【戻り】 成功 : TRUE
失敗 : FALSE

【説明】 プリンター (DS) の一枚画像保持印画の動作を制御します。
一枚画像保持印画の動作指定は CspStat.h にマクロ定義されています。
シンボルの意味は下記の通りです。

RETENTION_MODE_OFF (0) 通常動作 (デフォルト)
RETENTION_MODE_ON (1) 一枚保持動作

注) 一枚画像保持は本コマンドで通常動作に戻すか、本体の電源が切られるまで動作します。

【文例】

```
< Visual C >
SetRetentionMode(PortNo, (DWORD)RETENTION_MODE_ON);

< Visual Basic >
SetRetentionMode(PortNo, RETENTION_MODE_ON)
```

オーバーコート仕上げ制御コマンド

【書式】　BOOL SetOvercoatFinish(long lPortNum, DWORD ovcoat);

【引数】　lPortNum：　ポート番号
ovcoat：　オーバーコート仕上げの種類選択

【戻り】　成功：　TRUE
失敗：　FALSE

【説明】　プリンターのオーバーコート仕上げを制御します。
オーバーコート仕上げの動作指定は CspStat.h にマクロ定義されています。
シンボルの意味は下記の通りです。

OVERCOAT_FINISH_GLOSSY　(0)　光沢（デフォルト）
OVERCOAT_FINISH_MATTE1　(1)　マット１

OVERCOAT_FINISH_PMATTE1　(11)　部分マット１（ 部分マット ：　） HR3000 推奨 ）

OVERCOAT_FINISH_FINEMATTE　(21)　ファインマット
OVERCOAT_FINISH_LUSTER　(22)　ラスター

OVERCOAT_FINISH_PMATTE11　(101)　部分マット(マット)
OVERCOAT_FINISH_PMATTE12　(121)　部分マット(ファインマット)
OVERCOAT_FINISH_PMATTE13　(122)　部分マット(ラスター)

注）　オーバーコート仕上げコマンドは、画像データ送信前に動作指定します。コマンドは 1 画像に対して 1 回有効です。プリンターは、指定オーバーコート動作後、光沢仕上げに戻ります。
プリンターの初期状態は従来の光沢仕上げですので、本コマンドを実行しなければ、従来通り光沢での印画が行えます。
ウォーターマークの場合、ウォーターマークを入れる位置や文字領域の大きさによっては、オーバーコートにシワが生じたり、リボンの貼り付き・破断等の不具合が生じる可能性があります。
印刷動作を十分に確認された上でご使用ください。
（透かし文字推奨：MS ゴシック、10.5pt、最大 2 行の領域を 2 つまで）

本コマンドは DS40/80, DS40T/80T のファームウェア対応バージョン、DS-RX1、CW02、DP-DS620、DP-DS820、DP-QW410、DP-DS80D 及び DP-TC10 で有効です。これ以外の機種が接続された場合は FALSE を返します。本コマンドはプリンターのバージョン確認を最初に行い、Matte 未対応バージョンの場合は FALSE を返します。

機種	Version	戻り値
CV (MPⅢ)/CW	-	FALSE
DS40/80	対応	TRUE
DS40T/80T	未対応	FALSE
その他	対応	TRUE

機種毎のオーバーコート仕上げ指定の対応状況については、以下の「オーバーコート仕上げ」の表を参照してください。

・オーバーコート仕上げ<全体タイプ>
○：対応　　－：非対応　　数値：対応ファームウェアバージョン（これ以降で動作）

オーバーコート仕上げ	機種							
	DS40T DS80T	DS40 DS80	CW2 DS-RX1	DP-DS80D	DP-DS620	DP-DS820	DP-QW410	その他
OVERCOAT_FINISH_GLOSSY	○	○	○	○	○	○	○	○
OVERCOAT_FINISH_MATTE1	0.31	1.30	○	○	○	○	○	-
OVERCOAT_FINISH_FINEMATTE	-	-	-	-	1.20	○	-	-
OVERCOAT_FINISH_LUSTER	-	-	-	-	1.30	○	-	-

・オーバーコート仕上げ<部分マットタイプ> ※1
○：対応　　ー：非対応　　数値：対応ファームウェアバージョン（これ以降で動作）

オーバーコート仕上げ	機種				
	DP-TC10	DP-DS620	DP-DS820	DP-QW410	その他
OVERCOAT_FINISH_PMATTE1	○	-	-	-	-
OVERCOAT_FINISH_PMATTE11	-	1.20	○	○	-
OVERCOAT_FINISH_PMATTE12	-	1.52	○	-	-
OVERCOAT_FINISH_PMATTE13	-	1.30	○	-	-

※1 部分マット印画の詳細については、
「13. 補足資料（3）部分マット印画について」を参照してください。

【文例】

```
< Visual C >
SetOvercoatFinish(PortNo, (DWORD) OVERCOAT_FINISH_MATTE);

< Visual Basic >
SetOvercoatFinish(PortNo, OVERCOAT_FINISH_MATTE)
```

印画リトライ制御コマンド

【書式】　BOOL SetRetryControl(long lPortNum, DWORD retry);

【引数】　lPortNum：　ポート番号
retry：　リトライ印刷無効/有効の選択
PRINT_RETRY_OFF　(0)　リトライ印刷【無効】
PRINT_RETRY_ON　(1)　リトライ印刷【有効】

【戻り】　成功：　TRUE
失敗：　FALSE

【説明】　プリンターにエラーが発生したときのプリンターバッファ内の印画データを、エラー解除後にリトライ印刷する機能を設定します。
リトライ制御の動作指定は CspStat.h にマクロ定義されています。
リトライ制御コマンドは DS40/80(Ver.1.30 以降)、及び DS-RX1, CW02, DP-DS80D, DP-DS620, DP-DS820, DP-QW410 で有効です。

注）リトライ制御コマンドは、画像データ送信前に動作指定します。コマンドは 1 画像に対して 1 回有効です。プリンターの初期状態は従来のリトライ無効ですので、本コマンドを実行しなければ、従来通りの動作を行います。動作の詳細は次の通りです。

両面機 DP-DS80D のカット紙使用時はリトライ機能を使用できません。

設定値：	「無効(Disable)」(初期値)	「有効(Enable)」
機能：	プリンタエラー時のプリンタバッファ内の印画データをクリアします。エラー解除後、クリアされた印画データを再送信して下さい。	プリンタエラー時のプリンタバッファ内の印画データを保持します。 エラー解除後、保持しているデータから印刷処理を再開します。 但し、電源をOFFした場合、データは保持されません。

各エラー毎のリトライ印刷動作に関して

エラー内容	リトライ印刷「無効」設定時	リトライ印刷「有効」設定時
ペーパーエンド	エラー解除後、印刷されなかった画像データは再送信してください。	エラー解除後、プリンタバッファ内に保持しているデータから印画処理を再開します。 用紙初期化動作が行われる場合は、用紙初期化動作後に印刷されます。
リボンエンド		
カバーオープン		
屑ボックスエラー		
ペーパージャム	エラー解除後、印刷されなかった画像データは再送信してください。	DP-DS820、DP-QW410： 印画リトライ機能が「有効」に設定されていても、印画データはクリアされます。 エラー解除後、印刷されなかった画像データは再送信してください。 その他の機種： エラー解除後、プリンタバッファ内に保持しているデータから印画処理を再開します。 用紙初期化動作が行われる場合は、用紙初期化動作後に印刷されます。
リボンエラー		
用紙設定ミス		
データエラー	エラー解除にはプリンタの電源を一度OFFにする必要があるため、印画リトライ機能の設定に関係なくプリンタバッファ内の印画データはクリアされます。 プリンタ再起動後、印刷されなかった画像データは再送信してください。	エラー解除にはプリンタの電源を一度OFFにする必要があるため、印画リトライ機能が「有効」に設定されていても、プリンタ電源OFFにより印画データはクリアされます。 プリンタ再起動後、印刷されなかった画像データは再送信してください。
ヘッド電圧異常		
ヘッド位置エラー		
電源ファン停止		
カッターエラー		
ヘッド温度異常		
メディア温度異常		
リボンテンションエラー		
RFIDモジュールエラー		
モータ温度異常		
USB給電電圧異常		
システムエラー		

メディア種別の取得（DP-DS620, DP-DS820）

【書式】 long GetRfidMediaClass(long lPortNum, LPSTR p);

【引数】 lPortNum： ポート番号
p： 受信バッファへのポインタ

【戻り】 成功： バッファ p へ受信した文字数
失敗： -1

【説明】 RF-ID タグに記録されているメディア種別データを返します。（4 桁の ASCII 文字）

・メディア種別情報

取得文字列	詳細
（なし）	デフォルトメディア
0001	Digital(SD)メディア
0002	Premium Digital(PD) メディア
0003	Pure Premium(PP)メディア
1004	メタリック紙
1023	シルバーパール紙

DP-DS620、DP-DS820、DP-QW410 以外の機種：未設定。（文字列なし）
※デフォルトメディアは機種によって異なります。

【文例】

< Visual C >

```
char rbuf[256];
if(GetRfidMediaClass( PortNo, (LPSTR)rbuf ) > 0 ){
        // 処理
}
```

< VB.NET >

```
Dim s As String = New String("", 255)
Dim i As Integer
i = GetRfidMediaClass(PortNo, s)
If i > 0 Then Text1.Text = VB.Left(s, i) Else Text1.Text = "ERROR!"
```

RF-ID タグ予備データの取得 (DS-RX1, DP-TC10, DP-DS620, DP-DS820, DP-QW410)

【書式】 long GetRfidReserveData(long lPortNum, LPSTR p, DWORD dwPage);

【引数】
lPortNum : ポート番号
p: 受信バッファへのポインタ
dwPage: 予備データのページ番号

【戻り】
成功 : バッファ p へ受信した文字数
失敗 : -1

【説明】 RF-ID タグに記録されている予備データを返します。(4 桁の ASCII 文字)

【文例】

```
< Visual C >
char rbuf[256];
if(GetRfidReserveData( PortNo, (LPSTR)rbuf ,0x01 ) > 0 ){ // ページ 1 を指定
        // 処理
}

< VB.NET >
Dim s As String = New String("", 255)
Dim i As Integer
i = GetRfidReserveData(PortNo, s, &h2 )  'ページ 2 を指定
If i > 0 Then Text1.Text = VB.Left(s, i) Else Text1.Text = "ERROR!"
```

メディア初期枚数の取得（DS-RX1, DP-TC10, DS40, DP-DS620, DP-DS820, DP-QW410）

【書式】 long GetInitialMediaCount(long lPortNum);

【引数】 lPortNum : ポート番号

【戻り】 成功 : メディア初期枚数
失敗 : -1

【説明】 メディアの初期枚数を返します。
メディア初期枚数は実際の枚数よりも+50 枚多い値になっています。
関数使用時は+50 の枚数を考慮した上で使用して下さい。(DP-DS620、DP-DS820、DP-QW410 を除く)

【文例】

```
< Visual C >
long number;
if(( number = GetInitialMediaCount( PortNo )) >= 0 ){
        // number にメディア初期枚数
}

< VB.NET >
Dim number As Integer
number = GetInitialMediaCount(PortNo)
```

反転ユニット、ファームウェアバージョン情報の取得（DP-DS80D）

【書式】 long GetVersionDuplex(long lPortNum, LPSTR p);

【引数】 lPortNum : ポート番号
p : 受信バッファへのポインタ

【戻り】 成功 : バッファ p へ受信した文字数
失敗 : -1

【説明】 反転ユニットのバージョン情報を文字列バッファへ受信します。
両面機以外のプリンターの場合、戻り値が-1 となります。

【文例】

```
< Visual C >
char rbuf[256];
if ( GetVersionDuplex( PortNo, (LPSTR)rbuf ) > 0 ){
        // 処理
}

< Visual Basic >
Dim rb As String * 255, Ver As String
Dim n As Long
n = GetVersionDuplex(PortNo, rb)
If n > 0 Then Ver = Left(rb, n)        '文字列を切出す場合
```

【バージョン情報】

*.** 反転ユニットのバージョン情報
△△△△△△ 反転ユニット未接続又はバージョン情報取得未対応の場合、バージョン情報にスペース△（0x20）が入ります。

フルカッターセットアップ（DS40, DP-DS620, DP-DS820, DP-QW410）

【書式】 long SetFullCutterSetUp(long lPortNum,
long lCutSize1, long lCutSize2, long lCutSize3, long lCutSize4);

【引数】
lPortNum: ポート番号
lCutSize1: 画像１カットサイズ設定　設定値：0、20～
lCutSize2: 画像２カットサイズ設定　設定値：0、20～
lCutSize3: 画像３カットサイズ設定　設定値：0、20～
lCutSize4: 画像４カットサイズ設定　設定値：0、20～

【戻り】
成功: 1
失敗: 0

【説明】
下表の用紙サイズのときに、１枚から最大カット枚数まで、指定した各サイズにカットします。
カットサイズは、合計したサイズが最大サイズを超えない範囲で、２インチから最大サイズまで0.1インチ単位で設定できます。
画像1カットサイズ設定に‘0’を設定した場合は、通常サイズで印画します。その場合、画像2～4のカットサイズ設定は無視されます。

カット枚数が1枚の場合は、画像2～4カットサイズ設定に‘0’を設定して下さい。
カット枚数が2枚の場合は、画像3～4カットサイズ設定に‘0’を設定して下さい。
カット枚数が3枚の場合は、画像4カットサイズ設定に‘0’を設定して下さい。

各カットサイズ設定で、上記設定範囲外の値を設定した場合、“0”が戻ります。

■用紙サイズとカットサイズ設定範囲

用紙サイズ	最大カット枚数	最大サイズ	画像	カットサイズ設定範囲	
				最小	最大
6x4(PC) (*1) 4x4 (*4) 4.5x4 (*5)	2 枚	4 インチ	1	20	40
			2	20	20
4x4.5 (*5) 4.5x4.5 (*4)	2 枚	4.5 インチ	1	20	45
			2	20	25
4x6 (*4) 4.5x6 (*4)	3 枚	6 インチ	1	20	60
			2	20	40
			3	20	20
5x7(2L) (*1) 8x7	3 枚	7 インチ	1	20	70
			2	20	50
			3	20	30
6x8(A5) (*2) 4.5x8 (*4)	4 枚	8 インチ	1	20	80
			2	20	60
			3	20	40
			4	20	20
6x9(A5W) (*1) 8x9 (*3)	4 枚	9 インチ	1	20	90
			2	20	70
			3	20	50
			4	20	30
8x10 (*3)	4 枚	10 インチ	1	20	100
			2	20	80
			3	20	60
			4	20	40
A4 (*3)	4 枚	11.7 インチ	1	20	117
			2	20	97
			3	20	77
			4	20	57
8x12 (*3)	4 枚	12 インチ	1	20	120
			2	20	100
			3	20	80
			4	20	60

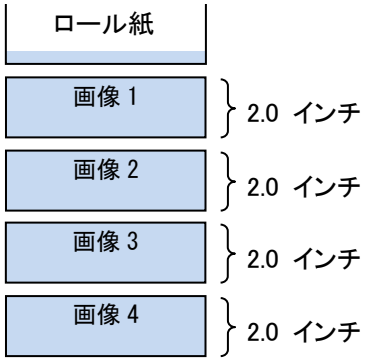
*1 DP-DS620 ファームウェアバージョン 01.20 以降対応
*2 DS40 ファームウェアバージョン 01.60 以降、および、DP-DS620 対応
*3 DP-DS820 対応
*4 DP-QW410 対応
*5 DP-QW410 ファームウェアバージョン 01.09 以降対応

【注意】　カッター制御コマンドと本コマンドの両方を指定した場合は、本コマンドの設定を優先します。
本コマンドで設定したカットサイズの合計が、プリンターに転送された画像サイズより大きい場合、
本コマンドは無効となり、プリンターに転送した用紙サイズ(通常サイズ)で印画します。

【文例】

```
< Visual C >
if((SetFullCutterSetUp( PortNo, 20, 20, 20, 20 )) == 0 ){
        //Error
}

< Visual Basic >
Dim Result As Long
Result = SetFullCutterSetUp (PortNo, 20, 20, 20, 20)
If Result = 0 Then GoTo Error
```

＊上記の文例を使用した場合の印刷結果は以下になります。



フルカッターセットアップ拡張設定（DS40, DP-DS620, DP-DS820, DP-QW410）

【書式】
```
long SetFullCutterSetUpEX(long lPortNum,
                          long lCutNum, long lDustSize, long lCutSize1, ... );
```

【引数】

lPortNum:	ポート番号
lCutNum:	画像カットサイズの設定数　設定値：1～6
lDustSize:	中間層カットサイズ設定　設定値：0、12～22（*1）
lCutSize1:	画像 1 カットサイズ設定　設定値：0、20～
lCutSize2:	画像 2 カットサイズ設定　設定値：0、20～
lCutSize3:	画像 3 カットサイズ設定　設定値：0、20～
lCutSize4:	画像 4 カットサイズ設定　設定値：0、20～
lCutSize5:	画像 5 カットサイズ設定　設定値：0、20～
lCutSize6:	画像 6 カットサイズ設定　設定値：0、20～

【戻り】

成功:	1
失敗:	0

【説明】

中間層カットサイズの設定が可能なフルカッターセットアップの拡張コマンドです。
画像カットサイズの設定数には、中間層カットサイズを除いた画像カットサイズの個数を設定してください。設定範囲は 1～6 です。
中間層カットサイズは 0.12～0.22 インチの範囲（*1）で、0.01 インチ単位で設定できます。
中間層カットサイズを 0 以外の有効な範囲で設定する場合、下記の式のカットサイズ合計が用紙の最大サイズを超えないように設定してください。

カットサイズ合計 = (各画像のカットサイズ合計)+((中間層カットサイズ/10)x(画像枚数-1))

例)用紙サイズ 6x9 インチ(90)、画像 1～4 を各 2 インチ(設定値 20)で 4 枚、中間層を 0.22 インチ(設定値 22)に設定した場合の計算

(20+20+20+20)+((22/10)x(4-1)) = 86.6 ≦ 90

中間層カットサイズを 0 にした場合の各画像の設定可能範囲は、「フルカッターセットアップ」の表、用紙サイズとカットサイズ設定範囲を参照してください。

各カットサイズ設定で、上記設定範囲外の値を設定した場合、“0”が戻ります。

本コマンドは下表の機種・ファームウェアバージョンで有効

機種	DP-DS620	DS40	DP-DS820	DP-QW410
ファームウェア Ver.	01.20 以上	01.60 以上	○	○

用紙サイズとカットサイズ設定範囲については、フルカッターセットアップの表「用紙サイズとカットサイズ設定範囲」を参照してください。

(*1) DP-QW410 は、0（中間層なし）又は 22 のみ設定可能。

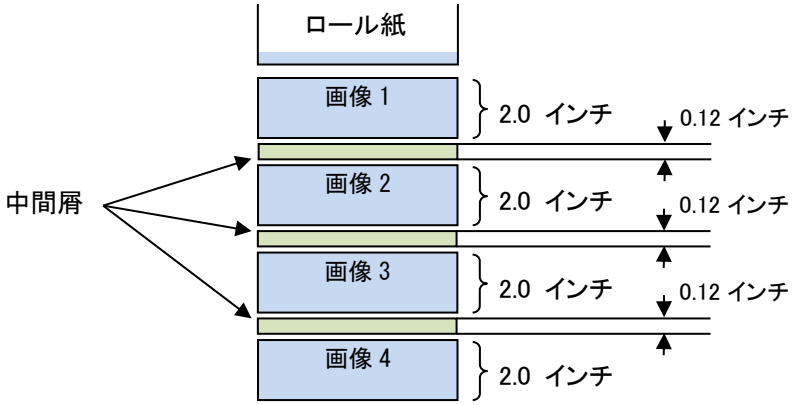
【注意】

カッター制御コマンドと本コマンドの両方を指定した場合は、本コマンドの設定を優先します。
本コマンドで設定したカットサイズの合計が、プリンターに転送された画像サイズより大きい場合、本コマンドは無効となり、プリンターに転送した用紙サイズ(通常サイズ)で印画します。

【文例】

```
< Visual C >
if((SetFullCutterSetUpEX( PortNo, 4, 12, 20, 20, 20, 20 )) == 0 ){
        //Error
}

< Visual Basic >
Dim Result As Long
Result = SetFullCutterSetUpEX (PortNo, 4, 12, 20, 20, 20, 20)
If Result = 0 Then GoTo Error
```

＊上記の文例を使用した場合の印刷結果は以下になります。



スタンバイモード移行時間設定 (DP-DS620, DP-DS820)

【書式】 long SetStandbyTime(long lPortNum、long lTime);

【引数】
lPortNum: ポート番号
lTime: スタンバイモード移行時間 設定値：0、1～99

【戻り】
成功: 1
失敗: 0

【説明】
スタンバイモードに移行するまでの時間を１分単位で設定します。
設定されている時間だけアイドル状態が継続すると、スタンバイモードに移行します。
この設定はプリンターに記憶されます（プリンターの電源を切っても有効）。

移行時間に‘0’が設定された場合は、スタンバイモードに入りません。
移行時間に上記設定範囲外の値を設定した場合、“0”が戻ります。

【注意】
デフォルトの10分より短い時間を設定する場合、低消費電力スタンバイモード移行と復帰を10分以下の短い周期で繰り返すような使用は、プリンター内の電源の寿命が短くなる可能性があるため、避けてください。
（電源のリレー寿命は、10分で低消費電力スタンバイモードに移行し10分後に復帰の繰り返しを、1日 12時間で月20日稼働としたとき、11年程度と推定されます。低消費電力スタンバイモード移行と復帰の周期を短くすると、その時間に比例して寿命は短くなります。）

【文例】

```
< Visual C >
if((SetStandbyTime( PortNo, 0 )) == 0 ){
        //Error
}

< Visual Basic >
Dim Result As Long
Result = SetStandbyTime( PortNo, 99 )
If Result = 0 Then GoTo Error
```

スタンバイモード移行時間設定値の取得 (DP-DS620, DP-DS820)

【書式】 long GetStandbyTime(long lPortNum);

【引数】
lPortNum： ポート番号

【戻り】
成功： スタンバイモード移行時間
失敗： -1

【説明】
スタンバイモード移行時間を返します。
本設定の初期値は、“10 分” となります。

【文例】

```
< Visual C >
long time;
if(( time = GetStandbyTime( PortNo )) > 0 ){
        // time にスタンバイモード移行時間
}

< VB.NET >
Dim time As Integer
time = GetStandbyTime( PortNo )
```

メディアエンド保持モード設定 (DP-DS620, DP-DS820, DP-QW410)

【書式】 long SetEndKeepMode(long lPortNum、long lMode);

【引数】
lPortNum: ポート番号
lMode: 保持モード設定　設定値：0、1

【戻り】
成功: 1
失敗: 0

【説明】
メディアエンド発生後、そのままの状態で、カバー開閉や電源を再投入したときに、メディアエンドを保持するか、保持しないかの動作モードを設定します。
メディアエンド（リボンエンド、ペーパーエンド）とメディアエラー（リボンエラー、ペーパージャム）が対象となります。
この設定はプリンターに記憶されます（プリンターの電源を切っても有効）。

モード設定に‘0’が設定された場合は、メディアエンド非保持モードになります。
モード設定に‘1’が設定された場合は、メディアエンド保持モードになります。(工場出荷時設定値)
モード設定に上記設定範囲外の値を設定した場合、“0”が戻ります。

【文例】
< Visual C >
```
if((SetEndKeepMode( PortNo, 0 )) == 0 ){
        //Error
}
```

< Visual Basic >
```
Dim Result As Long
Result = SetEndKeepMode( PortNo, 1 )
If Result = 0 Then GoTo Error
```

メディアエンド保持モード設定値の取得 (DP-DS620, DP-DS820, DP-QW410)

【書式】 long GetEndKeepMode(long lPortNum);

【引数】
lPortNum： ポート番号

【戻り】
成功： 保持モード設定
失敗： -1

【説明】
メディアエンド保持モード設定値を返します。
本設定の工場出荷時設定値は、“メディアエンド保持モード (1)” となります。

【文例】
< Visual C >
```
long mode;
if((mode = GetEndKeepMode( PortNo )) > 0 ){
        // mode にモード設定
}
```

< VB.NET >
```
Dim mode As Integer
mode = GetEndKeepMode( PortNo )
```

USB iSerialNumber 有効／無効 設定 (DS40, DS-RX1, DP-DS620, DP-DS820, DP-QW410)

【書式】 long SetUSBSerialEnable(long lPortNum, long dwEnable);

【引数】
lPortNum : ポート番号
dwEnable : 有効／無効設定　設定値：0、1

【戻り】
成功 : 1
失敗 : 0

【説明】
USB iSerialNumber の有効／無効を設定します。
この設定はプリンターに記憶されます（プリンターの電源を切っても有効）。
この設定変更は、プリンターの電源が再投入された際に有効となります。

有効／無効設定に‘0’が設定された場合は、USB iSerialNumber 無効になります。（デフォルト）
有効／無効設定に‘1’が設定された場合は、USB iSerialNumber 有効になります。
また、本設定を有効にした場合のシリアル番号は、製品固有情報である製品シリアル番号となります。

【文例】
< Visual C >

```
if( SetUSBSerialEnable(PortNo, 1) == 1 ){      // USB シリアルを使用する設定
        // 設定成功
}
```

< VB.NET >

```
If SetUSBSerialEnable(PortNo, 0) = 1 Then      ' USB シリアルを使用しない設定
        ' 設定成功
End If
```

USB iSerialNumber 設定値の取得 (DS40, DS-RX1, DP-DS620, DP-DS820, DP-QW410)

【書式】 long GetUSBSerialEnable(long lPortNum);

【引数】
lPortNum : ポート番号

【戻り】
USB iSerialNumber 有効: 1
USB iSerialNumber 無効: 0
失敗 : -1

【説明】
USB iSerialNumber 有効／無効設定を取得します。
本設定の初期値は、“USB iSerialNumber 無効 (0)” となります。

【文例】
< Visual C >

```
if( GetUSBSerialEnable( PortNo ) == 1 ){
        // 使用する
}
```

< VB.NET >

```
If GetUSBSerialEnable( PortNo ) = 0 Then
        ' 使用しない
End If
```

Rewind モード設定 (DP-DS620, DP-DS820)

【書式】 long SetRewindMode(long lPortNum, int iMode);

【引数】
lPortNum : ポート番号
iMode : モード設定　設定値：0、1

【戻り】
成功 : 1
失敗 : -1

【説明】
Rewind モードを設定します。
モード設定に‘0’が設定された場合は、Rewind モードが解除されます。(デフォルト)
モード設定に‘1’が設定された場合は、Rewind モードになります。

Rewind モードに設定されている場合、複数画面割付用紙サイズを指定して 1 画面分のイメージデータのみ送信して印刷を実行すると、用紙サイズは以下の巻き戻し印刷用の用紙サイズに置換されて印刷されます。

指定した用紙サイズ	置換された用紙サイズ
CSP_Lx2	CSP_L_REWIND
CSP_PCx2	CSP_PC_REWIND
CSP_6x4P5x2	CSP_6x4P5_REWIND
CSP_8x4x2	CSP_8x4_REWIND
CSP_8x5x2	CSP_8x5_REWIND
CSP_8x6x2	CSP_8x6_REWIND
CSP_8x4x3	CSP_8x4_REWIND
CSP_A5x2	CSP_A5_REWIND
CSP_A4x5x2	CSP_A4x5_REWIND

※ただし以下の複数画面割付用紙サイズは Rewind モードの対象外です。
CSP_8x5_8x4, CSP_8x6_8x4, CSP_8x6_8x5, CSP_8x8_8x4

【文例】

< Visual C >
```
if(SetRewindMode( PortNo, 1 ) == 1 ){          // Rewind モードに設定する
        // 設定成功
}
```

< VB.NET >
```
If SetRewindMode( PortNo, 0 ) = 1 Then         ' Rewind モードを解除する
        ' 設定成功
End If
```

Rewind モードの取得 (DP-DS620, DP-DS820)

【書式】 long GetRewindMode(long lPortNum);

【引数】 lPortNum : ポート番号

【戻り】 Rewind モード設定中: 1
Rewind モード解除 : 0
失敗 : -1

【説明】 Rewind モードを取得します。
本設定の初期値は、“Rewind モード解除 (0)” となります。

【文例】

```
< Visual C >
if(GetRewindMode( PortNo ) == 1 ){
        // Rewind モード設定中
}

< VB.NET >
If GetRewindMode( PortNo ) = 0 Then
        ' Rewind モード解除
End If
```

パノラマ印画開始可能チェック (DP-DS620, DP-DS820)

【書式】 DWORD GetPanoramaPrintable(long lPortNum);

【引数】 lPortNum : ポート番号

【戻り】 成功 : パノラマ印画開始可能かどうかのプリンターステータス
失敗 : STATUS_COMM_ERROR

【説明】 パノラマ印画開始可能かどうか、プリンターの状態を返します。

本コマンドは下表の機種・ファームウェアバージョンで有効。

機種	DP-DS620	DP-DS820
ファームウェア Ver.	01.20 以降	○

各ステータスの値は、CspStat.h にマクロ定義されています。
シンボルの意味は以下の通りです。

戻り値	値[HEX]	状態
STATUS_PRINTABLE	0x00000000	パノラマ印画開始可能
STATUS_HIGHTEMP_HEAD	0x00000001	ヘッド高温
STATUS_HIGHHUMIDITY *1	0x00000100	高湿
STATUS_LOWTEMP_MEDIA *2	0x00000020	メディア低温
STATUS_OTHER_STATE	0x00010000	その他の状態(印刷中、クーリング中、エラー等) プリンターステータスの取得コマンドで内容が確認できます。
STATUS_COMM_ERROR	0xFFFFFFFF	通信エラー

*1: DP-DS620 ファームウェアバージョン 01.40 以降、及び、DP-DS820 01.01 以降対応
*2: DP-DS620 ファームウェアバージョン 01.40 以降、及び、DP-DS820 01.03 以降対応

【注意】

・ヘッド高温状態の場合は、パノラマ印画可能状態になるのを待たずに、パノラマ印画データを送信してください。(通常印画と同様にクーリングを開始し、ヘッド温度低下後に印画が開始されます。)

・湿度が高い状況での連続パノラマ印画は、画質が低下する可能性があるため、推奨いたしません。

・メディア温度が低い状況での連続パノラマ印画は、行わないでください。(画像によってはジャム等の不具合が発生する可能性があります。)

(余白ありパノラマ印画の詳細については“13. 補足資料 (5) 余白ありパノラマ印画”を参照して下さい。)
(連続パノラマ印画については連続パノラマ SDK を参照してください。)

【文例】

```
< Visual C >
DWORD stat;
stat = GetPanoramaPrintable( PortNo );
if(stat == STATUS_LOWTEMP_MEDIA){
        // 連続パノラマ印画を実行しないでください
}else if(stat == STATUS_HIGHHUMIDITY){
        // 連続パノラマ印画の実行は推奨いたしません
}else if((stat == STATUS_PRINTABLE) || (stat == STATUS_HIGHTEMP_HEAD)){
        // パノラマ印画を開始してください
}else{
        // プリンターステータスの確認等
}

< VB.NET >
Dim stat As UInt32
stat = GetPanoramaPrintable( PortNo )
If stat = STATUS_LOWTEMP_MEDIA Then
        ' 連続パノラマ印画を実行しないでください
Else If stat = STATUS_HIGHHUMIDITY Then
        ' 連続パノラマ印画の実行は推奨いたしません
Else If (stat = STATUS_PRINTABLE) Or (stat = STATUS_HIGHTEMP_HEAD) Then
        ' パノラマ印画を開始してください
Else
        ' プリンターステータスの確認等
End If
```

メディアオフセット枚数の取得 (DS40, DS-RX1)

【書式】 long GetMediaCountOffset(long lPortNum);

【引数】 lPortNum : ポート番号

【戻り】 成功 : メディアオフセット枚数
失敗 : -1

【説明】 メディアのオフセット枚数(実際の枚数に対してあらかじめ加算されている値)を返します。

【文例】

```
< Visual C >
Long qtr_offset;
qtr_offset = GetMediaCountOffset( PortNo );

< VB.NET >
Dim qtr_offset As Integer
qtr_offset = GetMediaCountOffset( PortNo )
```

連続パノラマ印画設定 (DP-DS620, DP-DS820)

【書式】 BOOL SetContPanorama(long lPortNum, DWORD dwContinuous, DWORD dwOverlap);

【引数】
lPortNum : ポート番号
dwContinuous: 連続パノラマ印画指定
dwOverlap: 重ね合わせ幅 0-200(推奨値 200)

【戻り】
成功 : TRUE
失敗 : FALSE

【説明】 連続パノラマ印画動作を設定します。

本コマンドは下表の機種・ファームウェアバージョンで有効。

機種	DP-DS620	DP-DS820
ファームウェア Ver.	01.31 以降	○

連続パノラマ印画指定の値は、CspStat.h にマクロ定義されています。
シンボルの意味は以下の通りです。

CONT_PANORAMA_PRINT : 連続パノラマ印画指定
CONT_PANORAMA_LAST : 連続パノラマ印画最終画像、または通常印画(連続パノラマキャンセル)

重ね合わせ幅は、0.00～2.00 インチの範囲で、0.01 単位で設定できます。

【注意】 連続パノラマ印画指定は 1,2 ページ目のみ設定が有効になります。3 ページ目に連続パノラマ指定を行った場合は、設定が無視され用紙のカットを行います。

重ね合わせ幅は、連続パノラマの 1 枚目の印画で設定した値が 2 ページ目以降の印画にも適用され、2 ページ目以降に異なる値を設定した場合は無効となります。
また、重ね合わせ幅を推奨値よりも狭くした場合、重ね合わせ部分の印画品質が低下します。

【文例】

< Visual C >

```
if( SetContPanorama( PortNo, CONT_PANORAMA_MODE, 200 ) ){        // 連続パノラマ印画設定
        // 設定成功
}
```

< VB.NET >

```
If SetContPanorama( PortNo, CONT_PANORAMA_MODE, 200 ) Then        ' 連続パノラマ印画設定
        ' 設定成功
End If
```

プリント速度指定コマンド (DP-DS820, DP-QW410)

【書式】 BOOL SetPrintSpeed(long lPortNum, long lPrintSpeed);

【引数】 lPortNum: ポート番号
lPrintSpeed : 印画速度

0: 通常速度 (デフォルト)
1: 解像度600dpi印画速度 *1
2: 低速印画速度(DP-DS820)、もしくは、高画質印画速度(DP-QW410)
3: 高濃度印画速度 *1

*1:DP-DS820のみ

【戻り】 成功: TRUE
失敗: FALSE

【説明】 指定したプリント速度で印刷を行います。
印画速度を下げて高画質化を目指したい場合に指定します。

【注意】 本コマンドは1画像毎に指定します。
プリンターは1画像印刷後、デフォルト設定に戻ります。

【文例】

```
< Visual C >
if( SetPrintSpeed (PortNo, 3) ){      // 高濃度印画速度
        // 設定成功
}

< VB.NET >
If SetPrintSpeed (PortNo, 0) Then     //  通常印画速度
        ' 設定成功
End If
```

ガンマテーブルの設定 (DP-DS820)

【書式】 BOOL SetGammaTable(long lPortNum, LPBYTE lpGammaTable, long lDataLen);

【引数】
lPortNum : ポート番号
lpGammaTable : ガンマテーブルデータの先頭アドレス。
lDataLen : ガンマテーブルデータのサイズ。(Byte 単位)

【戻り】
成功 : TRUE
失敗 : FALSE

【説明】
ガンマテーブルデータを設定します。
ガンマテーブルデータの形式については、下表をご参照ください。

種別	項目	説明
ヘッダー (8byte)	Data unit size	0002h 固定
	Row	0100h 固定
	Plane	0004h 固定
	Checksum	xxxxh(拡張データとガンマテーブルデータのサム値)
拡張データ (8byte)	Mode	0000h 固定
	Rsv1	0000h 固定
	Rev2	0000h 固定
	Rev	0000h 固定
ガンマテーブルデータ (2048byte)	Y[0]…Y[255] M[0]…M[255] C[0]…C[255] OP[0]…OP[255]	0000h～FFFFh

【注意】
ガンマテーブルデータは、プリンターの状態が電源 OFF、もしくは、スタンバイモードに移行すると失われるため、印刷前に送信する必要があります。

【文例】

```
< Visual C >
BYTE tableData[2064];
…
if(SetGammaTable( PortNo, tableData, 2064 )){
        // 成功
}

< VB.NET >
Dim tableData(2064) As Byte
…
If SetGammaTable( PortNo, tableData, 2064 ) Then
        ' 成功
End If
```

ガンマテーブルチェックサム取得 (DP-DS820)

【書式】 long GetGammaTableChecksum(long lPortNum, LPSTR p);

【引数】 lPortNum : ポート番号
lpRcvBuf : 受信バッファのポインタ。

【戻り】 成功 : バッファ p へ受信した文字数
失敗 : -1

【説明】 ガンマテーブルのチェックサム値を取得します。
但し、プリンターにガンマテーブルデータが設定されていない場合は、４文字のスペース文字列 "□□□□"（CR<0Dh>終端、４バイトパディング）を受信します。

【文例】

```
< Visual C >
char sum[256];
if(GetGammmaTableChecksum(PortNo, sum ) > 0 ){
        // 成功
}

< VB.NET >
Dim sum As String = New String( "" , 256)
If GetGammmaTableChecksum(PortNo, sum ) > 0 Then
        ' 成功
End If
```

データテーブルのクリア (DP-DS820)

【書式】 BOOL SetDataTableClear(long lPortNum);

【引数】 lPortNum : ポート番号

【戻り】 成功 : TRUE
失敗 : FALSE

【説明】 プリンターに設定したガンマテーブルの値をクリアします。

【文例】

```
< Visual C >
if(SetDataTableClear( PortNo )){
        // 成功
}

< VB.NET >
If SetDataTableClear( PortNo ) Then
        ' 成功
End If
```

デカール制御設定

【書式】 BOOL SetDecurlCtrl(long lPortNum, WORD wMode);

【引数】
lPortNum : ポート番号
wMode : デカール動作設定
0:デカール動作しない
1:デカール動作する
2:自動制御 (デフォルト)

【戻り】
成功 : TRUE
失敗 : FALSE

【説明】
用紙のデカール動作を設定します。
デカール制御設定は、画像データ送信前に設定します。
デカール制御設定は1画像に対して1回有効です。
プリンターは1画像毎にデフォルト設定に戻ります。

【文例】

```
< Visual C >
if(SetDecurlCtrl(PortNo, 1 )){
        // 成功
}

< VB.NET >
If SetDecurlCtrl(PortNo, 1 ) Then
        ' 成功
End If
```

現在時刻通知

【書式】 BOOL SetSysTime(long lPortNum, LPSTR lpSystemTime);

【引数】
lPortNum : ポート番号
lpSystemTime : 現在時刻 (フォーマット : yyyymmddHHMMSS)
範囲 : 20000101000000(2000年1月1日0時0分0秒)～
99991231235959(9999年12月31日23時59分59秒)

【戻り】
成功 : TRUE
失敗 : FALSE

【説明】
プリンターに現在時刻を通知します。
プリンターは時計機能を搭載していないため、プリンターの電源起動後は初期値(2000年1月1日0時0分0秒)が設定されます。
範囲外の値を設定した場合、各設定値の最小又は最大の値で設定されます。
この現在時刻は、デカール制御設定が「自動制御」の場合に使用されます。
SendImageData()を実行する前に、設定してください。

lpSystemTime 入力例 :
現在時刻が2019年4月26日8時30分12秒の場合 => 20190426083012

【文例】

```
< Visual C >
if(SetSysTime( PortNo, "20190426083012")){
        // 成功
}

< VB.NET >
If SetSysTime( PortNo, "20190426083012" ) Then
        ' 成功
End If
```

対応メディア情報の取得(DP-DS620)

【書式】 long GetSupportedMediaInfo(long lPortNum, LPSTR p);

【引数】 lPortNum : ポート番号
p : 受信バッファへのポインタ

【戻り】 成功 : バッファ p へ受信した文字数
失敗 : -1

【説明】 プリンターが対応しているメディアの情報を取得します。

■DP-DS620
設定値は、対応メディア情報として以下の 8 桁の 10 進数 ASCII 数字が設定されます。

設定値	対応メディア		
	Luxury(LX)メディア	Digital(SD)メディア	Premium Digital(PD)メディア
00000101	○	×	○
00000110	○	○	×
00000111	○	○	○

【文例】

< Visual C >

```
char rbuf[256];
if(GetSupportedMediaInfo(PortNo, (LPSTR)rbuf) > 0 ){
        // 成功
}
```

< VB.NET >

```
Dim s As String = New String("", 255)
Dim i As Integer
i = GetSupportedMediaInfo(PortNo, s)
If i > 0 Then
        Text1.Text = VB.Left(s, i)
Else
        Text1.Text = "ERROR!"
End If
```

汎用設定系コマンド

【書式】
```
BOOL SetCommand( long lPortNum, LPSTR lpCmd, DWORD dwCmdLen );
BOOL CvSetCommand( long lPortNum, LPSTR lpCmd, DWORD dwCmdLen );
```

【引数】

lPortNum：	ポート番号
lpCmd：	コマンドが格納されたバッファへのポインタ
dwCmdLen：	コマンドの文字数

【戻り】

成功：	TRUE
失敗：	FALSE

【説明】 プリンターへコマンドを送信します。

汎用取得系コマンド

【書式】
```
long GetCommandEX( long lPortNum,
                   LPSTR lpCmd,
                   DWORD dwCmdLen,
                   LPSTR lpRetBuff,
                   DWORD dwRetBuffSize
                   );
long CvGetCommandEX( long lPortNum,
                   LPSTR lpCmd,
                   DWORD dwCmdLen,
                   LPSTR lpRetBuff,
                   DWORD dwRetBuffSize
                   );
```

【引数】

lPortNum：	ポート番号
lpCmd：	コマンドが格納されたバッファへのポインタ
dwCmdLen：	コマンド文字数
lpRetBuff：	受信データを格納するバッファへのポインタ
dwRetBuffSize：	受信バッファの許容サイズ

【戻り】

成功：	受信バイト数（lpRetBuff へ受信データ）
失敗：	-1

【説明】 プリンターへコマンドを送信後、受信データをバッファへ格納します。

ミニラボ タワーエラーステータスの取得

【書式】 long GetMiniLabTowerStatus(long lPortNum);

【引数】 lPortNum： ポート番号

【戻り】 成功： タワーエラーステータス
失敗： STATUS_ERROR

【説明】 タワーのステータスを返します。
ステータスのビット位置は CspStat.h にマクロ定義されています。
シンボルの意味は下記の通りです。

TGROUP_CXM0	[シュータ/ソータエラー] グループを示すビット
TGROUP_CXM1	[シュータ/ソータエラー] グループを示すビット
TGROUP_CXM2	[シュータ/ソータエラー] グループを示すビット
TGROUP_CXM3	[シュータ/ソータエラー] グループを示すビット
TGROUP_CXM4	[シュータ/ソータエラー] グループを示すビット
/* tower normal */	
STATUS_CXM_TOWER_NORMAL	タワー通常動作
/*ソータ関係*/	
STATUS_CXM_SRT_SENSON	ソータユニット 用紙位置 MAX 通知（警告）
STATUS_CXM_SRT_MAXERR	ソータユニット 用紙位置 MAX 通知（エラー）
/*マルチソータ関係*/	
STATUS_CXM_SRT_SYSERR	ソータユニット システムエラー
STATUS_CXM_SRT_MOTERR	ソータユニット モーター動作エラー
STATUS_CXM_SRT_TIMEOUT	ソータユニット ソータ送りタイムアウト
/* シュータ カバー、ドア関係 */	
STATUS_CXM_SHT_COPENA	シュータ カバーオープン A
STATUS_CXM_SHT_COPENB	シュータ カバーオープン B
STATUS_CXM_SHT_COPENC	シュータ カバーオープン C
STATUS_CXM_SHT_COPEND	シュータ カバーオープン D
STATUS_CXM_SHT_DOPENA	シュータ ドアオープン A
STATUS_CXM_SHT_DOPENB	シュータ ドアオープン B
STATUS_CXM_SHT_CORNER_OPEN	コーナーユニットオープン
/*裏面印画ユニットドア関係*/	
STATUS_CXM_BP_DOPENA	裏面印刷ユニット ドアオープン A
STATUS_CXM_BP_DOPENB	裏面印刷ユニット ドアオープン B
STATUS_CXM_BP_DOPENC	裏面印刷ユニット ドアオープン C
STATUS_CXM_BP_DOPEND	裏面印刷ユニット ドアオープン D
STATUS_CXM_BP_DOPENE	裏面印刷ユニット ドアオープン E
/* シュータ用紙関係 */	
STATUS_CXM_SHT_PJAM_A	シュータ 用紙ジャム A 位置
STATUS_CXM_SHT_PJAM_B	シュータ 用紙ジャム B 位置
STATUS_CXM_SHT_PJAM_CORNER	シュータ 用紙ジャム コーナー
/*裏面印画ユニット用紙関係*/	
STATUS_CXM_BP_PJAM_A	裏面印刷ユニット 用紙ジャム A
STATUS_CXM_BP_PJAM_ABC	裏面印刷ユニット 用紙ジャム A or B or C

Dai Nippon Printing Co., Ltd.

```
/* その他*/
STATUS_CXM_SHT_SYSERR          シュータ　システムエラー
STATUS_CXM_RS422_NONC          シュータ　RS422 通信未接続エラー
STATUS_CXM_BP_WDP              裏面印刷ユニット　ワイヤードットプリンター

STATUS_CXM_KIOSK_MODERR        キオスクモードエラー　注１）
STATUS_CXM_BP_MODERR           ソータモードエラー　　注１）
```

注）キオスクモードとソータモードはタワー本体のスイッチで切り替えます。
キオスク、ソータモードエラーはそれぞれの動作モードの時の、BP ユニットの接続状態で判断します。

タワースイッチ（動作モード）	BP ユニットの接続状態	エラー
キオスクモード	有り	キオスクモードエラー
ソータモード	無し	ソータモードエラー

ミニラボ 印画データグループ No.　シリアル No. 初期値設定

【書式】　BOOL SetMiniLabGroupNoIni(long lPortNum, long lGrNo, long lSrNo, long lOption);

【引数】

lPortNum：	ポート番号		
lGrNo：	グループ No.初期値	1～10000	通常動作
		0x55AA	バルク転送モード
lSrNo：	シリアル No.初期値	1～10000	通常動作
		0x55AA	バルク転送モード
lOption:	動作指定オプション	0：	指定なし
		1：	ソータ動作

【戻り】

成功：	TRUE
失敗：	FALSE

【説明】

タワーコントローラのグループ No.とシリアル No.の初期値設定（ポインタ変更）を行います。
ソータの動作は次の条件で動作します。

- 動作指定オプションに１（ソータ動作）を指定し、新しいグループの画像が排出された時。
- ソータの一山が一定量に達した場合。

バルク転送モード指定時、タワーは受け付けた GrNo. SrNo.の順番に関係なく、印刷と排出準備が整った画像から排出します。この場合のソータ動作は GrNo. SrNo.によって切り替わりません。

注）本コマンドは印画終了後の順次実行コマンドとして処理されます。
本コマンドは必ずグループの最終画像を印画したプリンターへ送信して下さい。

例）プリンター３台接続した場合の送信方法

① Gr.1 Sr1 画像送信　⇒　プリンター 1
② Gr.1 Sr2 画像送信　⇒　プリンター 2
③【初期値設定】SetMinilabGroupNoIni(Gr.2 Sr.1,0) ⇒　プリンター 2
④ Gr.2 Sr1 画像送信　⇒　プリンター 3

コントローラは②の印画終了後、③で GrNo. SrNo.の初期値設定（ポインタ変更）を行います。
その後、④の新しい GrNo.を受け付けます。

【文例】

```
< Visual C >
long lGrNo = 1;                 // グループ No　初期化
long lSrNo = 1;                 // シリアル No　初期化
char text[82];
char text2[41];
char len;

//  ①グループ No 1　シリアル No 1
SetMiniLabGroupNo(PortNo1, lGrNo);                      // グループ No.送信
SetMiniLabSerealNo(PortNo1, lSrNo, SERIAL_NUM_SET);     // シリアル No.送信
strcpy( text, "Group No.1 -¥r" );                       // １行目データ
strcpy( text2, "Serial No.1¥r");                        // ２行目データ
strcat( text, text2 );                                  // １行目と２行目のデータを合成
len = strlen(text);                                     // 文字列長さ
SetMiniLabBackPrintData( PortNo1, (LPSTR)text, len);    // 裏面印刷データ送信

//  ここで印画データを送信

//  ②グループ No 1　シリアル No 2（グループ No 1　終了）
lSrNo += 1;                                             // シリアル No　更新
SetMiniLabGroupNo(PortNo2, lSrNo);                      // グループ No.送信
SetMiniLabSerealNo(PortNo2, lSrNo, SERIAL_NUM_SET);     // シリアル No.送信
strcpy( text, "Group No.1 -¥r" );                       // １行目データ
strcpy( text2, "Serial No.2¥r");                        // ２行目データ
strcat( text, text2 );                                  // １行目と２行目のデータを合成
len = strlen(text);                                     // 文字列長さ
SetMiniLabBackPrintData( PortNo2, (LPSTR)text, len);    // 裏面印刷データ送信

//  ここで印画データを送信

lGrNo += 1;                                             // グループ No　更新
lSrNo = 1;                                              // シリアル No　初期化
// ③グループ、シリアル No.初期化
SetMinilabGroupNoIni(PortNo2, lGrNo, lSrNo, 1) ;        // Sorter 制御

//  ④グループ No 2　シリアル No 1
SetMiniLabGroupNo(PortNo3, lGrNo);                      // グループ No.送信
SetMiniLabSerealNo(PortNo3, lSrNo, SERIAL_NUM_SET);     // 最終シリアル No.送信
strcpy( text, "Group No.2 -¥r" );                       // １行目データ
strcpy( text2, "Serial No.1¥r");                        // ２行目データ
strcat( text, text2 );                                  // １行目と２行目のデータを合成
len = strlen(text);                                     // 文字列長さ
SetMiniLabBackPrintData( PortNo3, (LPSTR)text, len);    // 裏面印刷データ送信

//  ここで印画データを送信
```

< Visual Basic >

```
Dim lGrNo As Long
Dim lSrNo As Long
Dim bpText As String * 255

lGrNo = 1                                                    'グループ No.を初期化
lSrNo = 1                                                    'シリアル No.を初期化

'   ①グループ No 1   シリアル No 1
SetMiniLabGroupNo(PortNo1, lGrNo)                            'グループ No.送信
SetMiniLabSerealNo(PortNo1, lSrNo, SERIAL_NUM_SET)           'シリアル No.送信
bpText = "Group No.1 -" & Chr(&HD) & "Sereal No.1" & Chr(&HD)
SetMiniLabBackPrintData( PortNo1, bpText, 24)                '裏面印刷データ送信

'   ここで印画データを送信

'   ②グループ No 1   シリアル No 2   (グループ No 1   終了)
lSrNo = lSrNo + 1                                            'シリアル No   更新
SetMiniLabGroupNo(PortNo2, lSrNo)                            'グループ No.送信
SetMiniLabSerealNo(PortNo2, lSrNo, SERIAL_NUM_SET)           'シリアル No.送信
bpText = "Group No.1 -" & Chr$(&HD) & "Sereal No.2" & Chr$(&HD)
SetMiniLabBackPrintData( PortNo2, bpText, 24)                '裏面印刷データ送信

'   ここで印画データを送信

lGrNo = lGrNo + 1                                            'グループ No   更新
lSrNo = 1                                                    'シリアル No   初期化
' ③グループ、シリアル No.初期化
SetMinilabGroupNoIni(PortNo2, lGrNo, lSrNo, 1)               'Sorter 制御

'   ④グループ No 2   シリアル No 1
SetMiniLabGroupNo(PortNo3, lGrNo)
SetMiniLabSerealNo(PortNo3, lSrNo, SERIAL_NUM_SET)           'シリアル No.送信
bpText = "Group No.2 -" & Chr$(&HD) & "Sereal No.1" & Chr$(&HD)
SetMiniLabBackPrintData( PortNo3, bpText, 24)

'   ここで印画データを送信
```

ミニラボ 印画データグループ No. 設定

【書式】 BOOL SetMiniLabGroupNo(long lPortNum, long lGrNo);

【引数】
lPortNum : ポート番号
lGrNo : 印画データのグループ No.(1～10000)
No.0 は特殊番号

【戻り】
成功 : TRUE
失敗 : FALSE

【説明】
印画データのグループ No. を設定します。
CXM はグループ No. により印画物をソートします。
No.0 は順番に関係なく、シュータが空いていれば排出される番号です。

【文例】 次頁を参照して下さい。

ミニラボ 印画データシリアル No. 設定

【書式】 BOOL SetMiniLabSerialNo(long lPortNum, long lSrNo, long lFinal);

【引数】
lPortNum : ポート番号
lSrNo : 印画データのシリアル No (1～10000)
No.0 は特殊番号
lFinal : 印画データシリアル No の終了通知

【戻り】
成功 : TRUE
失敗 : FALSE

【説明】
印画データのシリアル No. を設定します。
シリアル No. は 1 画像に対して、1 回指定します。
グループの最終データのときは、必ず lFinal に SERIAL_NUM_FINAL を指定して下さい。
動作指定は CspStat.h にマクロ定義されています。

SERIAL_NUM_SET 通常のシリアル No 指定時
SERIAL_NUM_FINAL シリアル No の最後

No.0 はシリアルの順番に関係なく、シュータが空いていれば排出される番号です。
No.0 を設定した場合、タワー内のシリアル番号はインクリメントされません。

例）タワー内のシリアル番号につて
■通常の順番制御

	タワー内シリアル番号
No.1 を設定	⇒ 2
No.2 を設定	⇒ 3
No.3 を設定	⇒ 4

■No.0 を使用した場合

	タワー内シリアル番号
No.1 を設定	⇒ 2
No.2 を設定	⇒ 3
No.0 を設定	⇒ 3 のままインクリメントされない。

【文例】 次頁を参照して下さい。

ミニラボ 裏面印刷データ 設定

【書式】 BOOL SetMiniLabBackprintData(long lPortNum, LPSTR lpText, long lTextLen)

【引数】
lPortNum : ポート番号
lpText : 裏面印刷データテキスト文字列（
データの終わりには CR(0xD)を入れて下さい。
lTextlen: 文字数（1-82）
CR を含めた文字数をセットしてください。

【戻り】
成功 : TRUE
失敗 : FALSE

【説明】
裏面に印画するテキストデータを設定します。
2 行のデータを設定する場合は、[1 行目テキスト] + CR + [2 行目テキスト] + CR としてください。
1 行の最大印字数は 40 文字です。

【文例】
< Visual C >

```
long lGrNo = 1;                 // グループ No 初期化
long lSrNo = 1;                 // シリアル No 初期化
char text[82];
char text2[41];
char len;

//  グループ No 1  シリアル No 1
SetMiniLabGroupNo(PortNo, lGrNo);                       // グループ No.送信
SetMiniLabSerealNo(PortNo, lSrNo, SERIAL_NUM_SET);      // シリアル No.送信
strcpy( text, "Group No.1 -¥r" );                       // １行目データ
strcpy( text2, "Serial No.1¥r");                        // ２行目データ
strcat( text, text2 );                                  // １行目と２行目のデータを合成
len = strlen(text);                                     // 文字列長さ
SetMiniLabBackPrintData( PortNo, (LPSTR)text, len);     // 裏面印刷データ送信

//ここで印画データを送信

//  グループ No 1  シリアル No 2（グループ No 1  終了）
lSrNo += 1;                                             //シリアル No 更新
SetMiniLabGroupNo(PortNo, lSrNo);                       // グループ No.送信
SetMiniLabSerealNo(PortNo, lSrNo, SERIAL_NUM_FINAL);    // 最終シリアル No.送信
strcpy( text, "Group No.1 -¥r" );                       // １行目データ
strcpy( text2, "Serial No.2¥r");                        // ２行目データ
strcat( text, text2 );                                  // １行目と２行目のデータを合成
len = strlen(text);                                     // 文字列長さ
SetMiniLabBackPrintData( PortNo, (LPSTR)text, len);     // 裏面印刷データ送信

//ここで印画データを送信

//  グループ No 2  シリアル No 1
lGrNo += 1;                                             // グループ No 更新
lSrNo = 1;                                              // シリアル No 初期化
SetMiniLabGroupNo(PortNo, lGrNo);                       // グループ No.送信
SetMiniLabSerealNo(PortNo, lSrNo, SERIAL_NUM_FINAL);    // 最終シリアル No.送信
strcpy( text, "Group No.2 -¥r" );                       // １行目データ
strcpy( text2, "Serial No.1¥r");                        // ２行目データ
strcat( text, text2 );                                  // １行目と２行目のデータを合成
len = strlen(text);                                     // 文字列長さ
SetMiniLabBackPrintData( PortNo, (LPSTR)text, len);     // 裏面印刷データ送信

//ここで印画データを送信
```

```
< Visual Basic >
Dim lGrNo As Long
Dim lSrNo As Long
Dim bpText As String * 255
lGrNo = 1                                                    'グループ No.を初期化
lSrNo = 1                                                    'シリアル No.を初期化

'グループ No 1  シリアル No 1
SetMiniLabGroupNo(PortNo, lGrNo)                             'グループ No.送信
SetMiniLabSerealNo(PortNo, lSrNo, SERIAL_NUM_SET)            'シリアル No.送信
bpText = "Group No.1 -" & Chr(&HD) & "Sereal No.1" & Chr(&HD)
SetMiniLabBackPrintData( PortNo, bpText, 24)                 '裏面印刷データ送信

'ここで印画データを送信

'グループ No 1  シリアル No 2  (グループ No 1  終了)
lSrNo = lSrNo + 1                                            'シリアル No  更新
SetMiniLabGroupNo(PortNo, lSrNo)                             'グループ No.送信
SetMiniLabSerealNo(PortNo, lSrNo, SERIAL_NUM_FINAL)          '最終シリアル No.送信
bpText = "Group No.1 -" & Chr$(&HD) & "Sereal No.2" & Chr$(&HD)
SetMiniLabBackPrintData( PortNo, bpText, 24)                 '裏面印刷データ送信

'ここで印画データを送信

'グループ No 2  シリアル No 1
lGrNo = lGrNo + 1                                            'グループ No  更新
lSrNo = 1                                                    'シリアル No  初期化
SetMiniLabGroupNo(PortNo, lGrNo)
SetMiniLabSerealNo(PortNo, lSrNo, SERIAL_NUM_FINAL)          '最終シリアル No.送信
bpText = "Group No.2 -" & Chr$(&HD) & "Sereal No.1" & Chr$(&HD)
SetMiniLabBackPrintData( PortNo, bpText, 24)

'ここで印画データを送信
```

印画中止コマンド

【書式】 BOOL StopPrint(long lPortNum);

【引数】 lPortNum : ポート番号

【戻り】 成功 : TRUE
失敗 : FALSE

【説明】 プリンターはこのコマンドを受信すると、印画中の場合には最後まで印画し、シュータへ排出できる場合は排出動作を行います。排出が出来ない場合は、印画物を細かくカットして破棄します。
また印画バッファにデータが有る場合にはバッファをクリアします。

【文例】

```
< Visual C >
StopPrint(PortNo);

< Visual Basic >
StopPrint(PortNo)
```

用紙排出先指定

【書式】 BOOL SetMiniLabOutputDest(long lPortNum, long lDest);

【引数】 lPortNum : ポート番号
lDest : 用紙排出先

【戻り】 成功 : TRUE
失敗 : FALSE

【説明】 ミニラボ使用時の用紙排出先の指定をします。
用紙の排出先指定は CspStat.h にマクロ定義されています。
シンボルの意味は下記の通りです。

OUTPUT_DEST_AUTO	(0)	用紙サイズにより自動選択（デフォルト）
OUTPUT_DEST_SORTER	(1)	ソータへ排出
OUTPUT_DEST_TRAY	(2)	フロントトレイに排出

【文例】

```
< Visual C >
SetMiniLabOutputDest(PortNo, OUTPUT_DEST_SORTER);

< Visual Basic >
SetMiniLabOutputDest(PortNo, OUTPUT_DEST_SORTER)
```

ソータ用紙 MAX 位置検出センサーON/OFF　設定

【書式】　BOOL SetMiniLabSrtSensor(long lPortNum, long lSens);

【引数】
lPortNum :　ポート番号
lSens :　用紙検出機能の切り替え

【戻り】
成功 :　TRUE
失敗 :　FALSE

【説明】
ソータの用紙ＭＡＸ位置を検出するセンサーの機能を設定します。
用紙位置検出指定は CspStat.h にマクロ定義されています。
シンボルの意味は下記の通りです。

SORTER_MAX_OFF　用紙位置検出を無視して排出
SORTER_MAX_ON　用紙位置検出する。(デフォルト)

【文例】

```
< Visual C >
SetMiniLabSrtSensor(PortNo, SORTER_MAX_ON);

< Visual Basic >
SetMiniLabSrtSensor(PortNo, SORTER_MAX_OFF)
```

ソータ用紙 MAX 位置検出センサーON/OFF　取得

【書式】　long GetMiniLabSrtSensor(long lPortNum);

【引数】
lPortNum :　ポート番号

【戻り】
成功 :　用紙検出機能の現在の状態
失敗 :　-1

【説明】
ソータの用紙ＭＡＸ位置を検出するセンサーの状態を取得します。
用紙位置検出指定は CspStat.h にマクロ定義されています。
シンボルの意味は下記の通りです。

SORTER_MAX_OFF　用紙位置検出を無視して排出
SORTER_MAX_ON　用紙位置検出する。(デフォルト)

【文例】

```
< Visual C >
long lSens;
lSens = GetMiniLabSrtSensor(PortNo, SORTER_MAX_ON);

< Visual Basic >
Dim lSens As Long
lSens = GetMiniLabSrtSensor(PortNo, SORTER_MAX_OFF)
```

ミニラボ グループ No. 取得

【書式】 long GetMiniLabGroupNo(long lPortNum);

【引数】 lPortNum : ポート番号

【戻り】 成功 : グループ No (1～10000)
失敗 : -1

【説明】 タワーコントローラに設定されている、現在のグループ No. を取得します。

注意) タワー動作中は常時グループ No が変化しております。
アプリケーションと同期を取りたい場合は、タワー停止中に取得して下さい。

【文例】

```
< Visual C >
long lGNo;
if((lGNo = GetMiniLabGroupNo(PortNo)) >= 0 ){
        // lGNo  グループ No.
}

< Visual Basic >
Dim lGNo As Long
lGNo = GetMiniLabGroupNo(PortNo)
```

ミニラボ シリアル No. 取得

【書式】 long GetMiniLabSerialNo(long lPortNum);

【引数】 lPortNum : ポート番号

【戻り】 成功 : シリアル No (1～10000)
失敗 : -1

【説明】 タワーコントローラに設定されている、現在のシリアル No. を取得します。

注意) タワー動作中は常時シリアル No が変化しております。
アプリケーションと同期を取りたい場合は、タワー停止中に取得して下さい。

【文例】

```
< Visual C >
long lSNo;
if((lSNo = GetMiniLabSerialNo(PortNo)) >= 0 ){
        // lSNo  シリアル No.
}

< Visual Basic >
Dim lSNo As Long
lSNo = GetMiniLabSerialNo(PortNo)
```

プリンター位置の取得

【書式】 long GetMiniLabPrinterAdd(long lPortNum);

【引数】 lPortNum : ポート番号

【戻り】 成功 : 未接続 (0)
プリンター位置 (1 - 4)
失敗 : -1

【説明】 プリンターがタワーにセットされている位置を取得します。
No.1 が最上段です。

【文例】

```
< Visual C >
long lAddr;
if((lAddr = GetMiniLabPrinterAdd(PortNo)) >= 0 ){
        // lAddr : printer address
}

< Visual Basic >
Dim lAddr As Long
lAddr = GetMiniLabPrinterAdd(PortNo)
```

13. 補足資料

(1)DP-TC10 シングルカットモードについて

① 動作概要

・印画先端部分の 3mm は次の印画と重なる部分とし、OP の印画を行いません。

・印画物をカットするときは、印画先端から 2mm の位置でカットします。

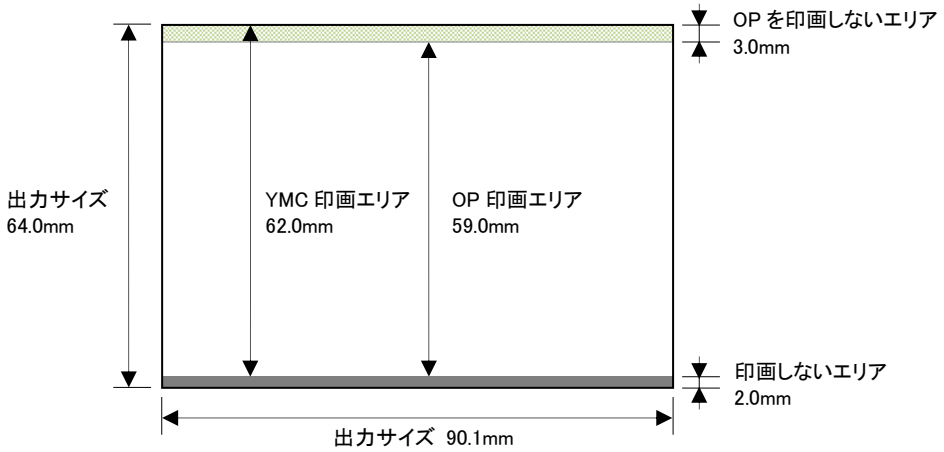
・印画終端部分は、前回の印画の残り 2mm の部分に 1mm 重なるように、用紙の終端部から 1mm の位置まで印画を行います。

・プリンターは縦 64mm の画像データを受信し、下部 2mm(300dpi:24dot、600dpi:48dot)のデータは印画しません。

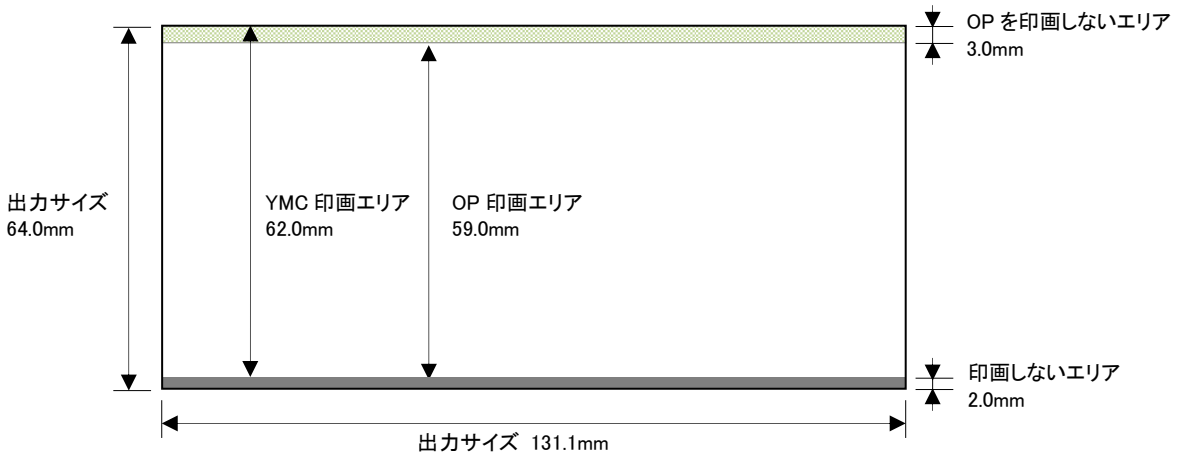
② プリントエリアサイズ

用紙種類	出力画像データサイズ(横 x 縦)	出力画像ピクセルサイズ		印刷サイズ
		解像度 300x600dpi	解像度 300x300dpi	
TC86x61	90.1 x 64.0mm	1064 x 1512 pixel	1064 x 756 pixel	90.1 x 62mm
TC127x61	131.1 x 64.0mm	1548 x 1512 pixel	1548 x 756 pixel	131.1 x 62mm

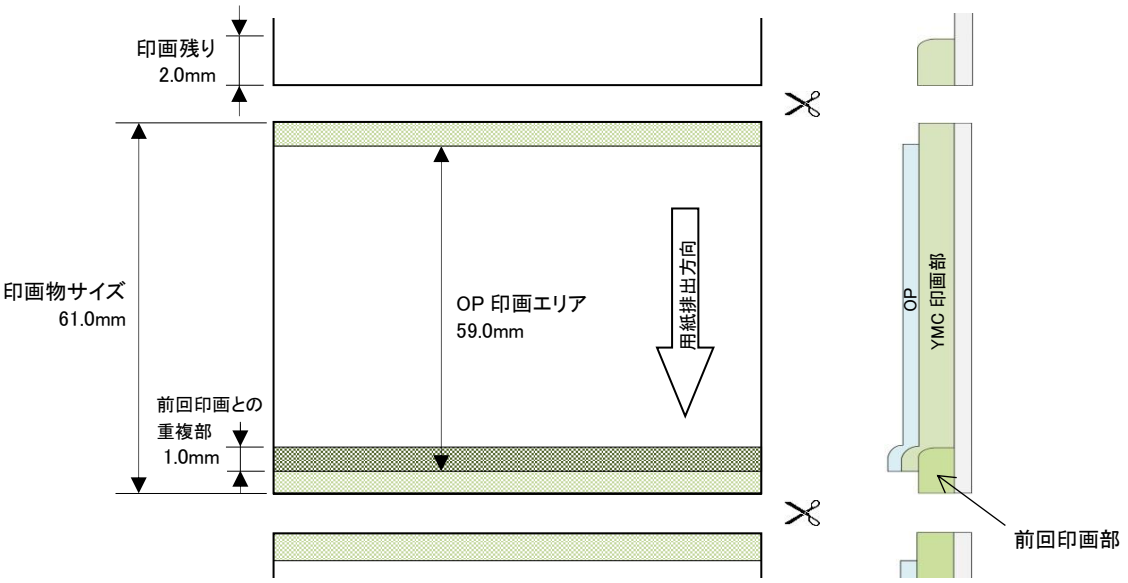
■ カードサイズ(TC86x61)



■ L 幅サイズ(TC127x61)



③ 印画結果



(2)DP-TC10 L カードサイズ 2 画面連続印画について

① 動作概要

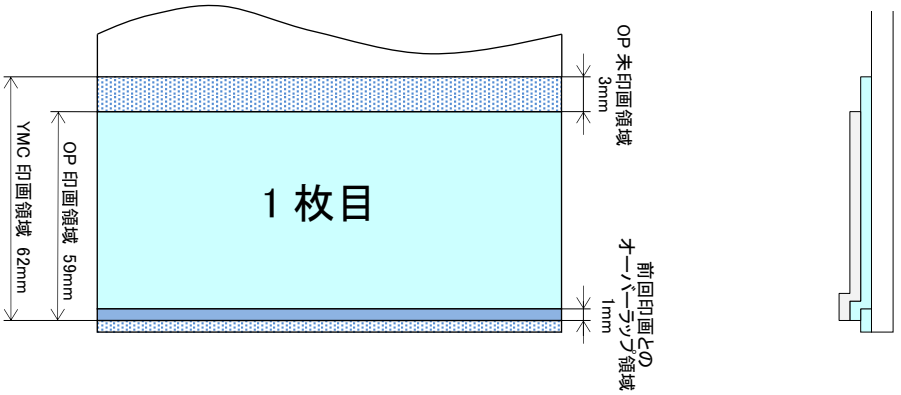
・ホストからのコマンド指示により、L カードサイズ 2 画面連続印画を行います。

・ホストから L カードサイズ 2 画面連続印画のコマンドを送信後、2 枚分の L カードサイズの印画データを送信することで、プリンターは 2 画面連続印画を行います。

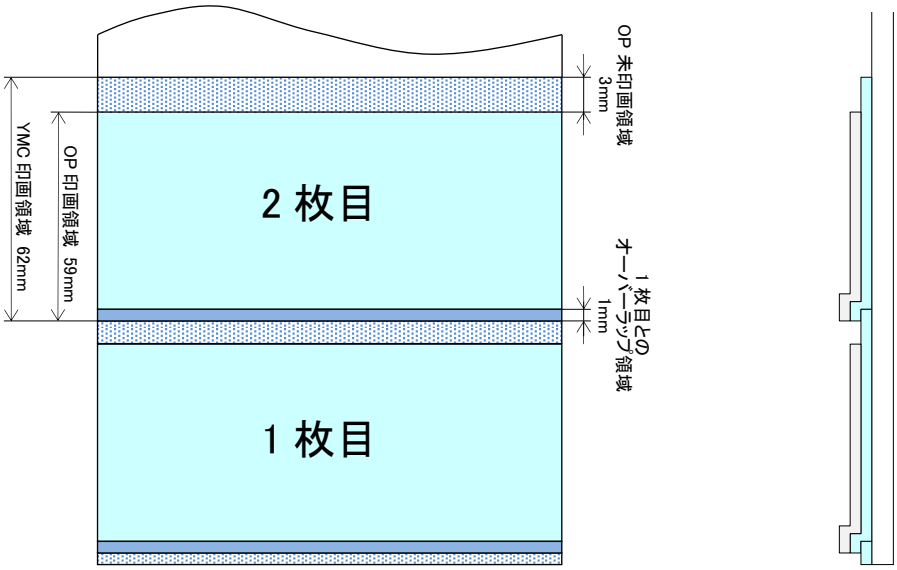
・2 画面連続印画が行えるのは L カードサイズ(127x61mm)で、カードサイズ(86x61mm)では 2 画面連続印画は行いません。

② プリントエリアサイズ

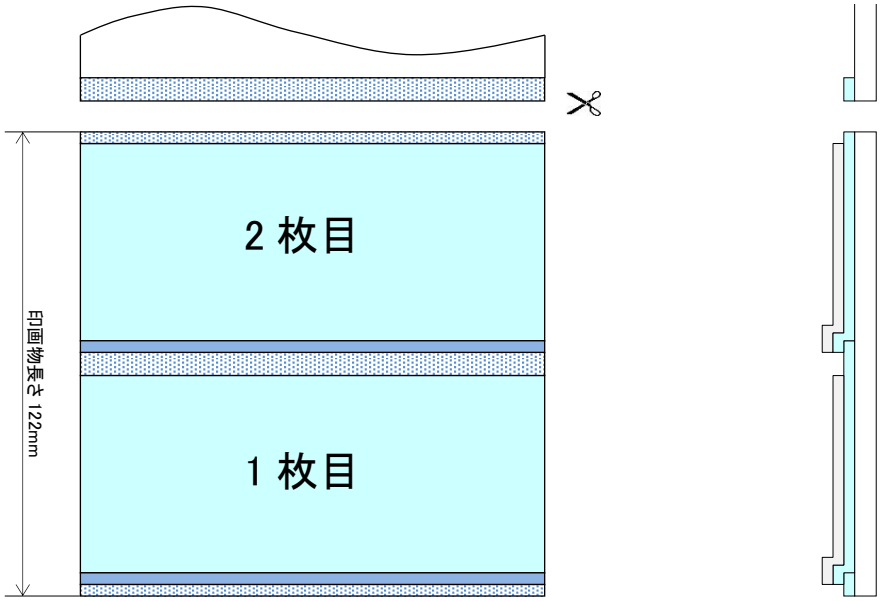
(Ⅰ)1枚目の印画



（Ⅱ）2枚目の印画



（Ⅲ）印画物カット



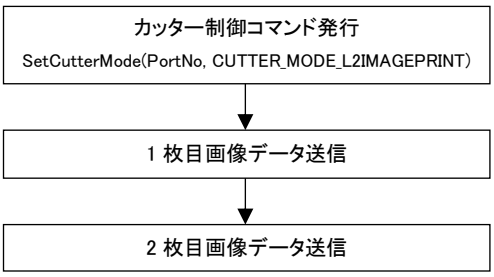
③ 2 画面連続印画方法

・2 画面連続印画の指定は、ダイレクト DLL のカッター制御コマンドで指定することが出来ます。

■ 指定方法

（Ⅰ） 2 画面連続印画を行う場合、1 枚目のデータを送信する前に、カッター制御コマンドで「CUTTER_MODE_L2IMAGEPRINT」(130)を指定してください。

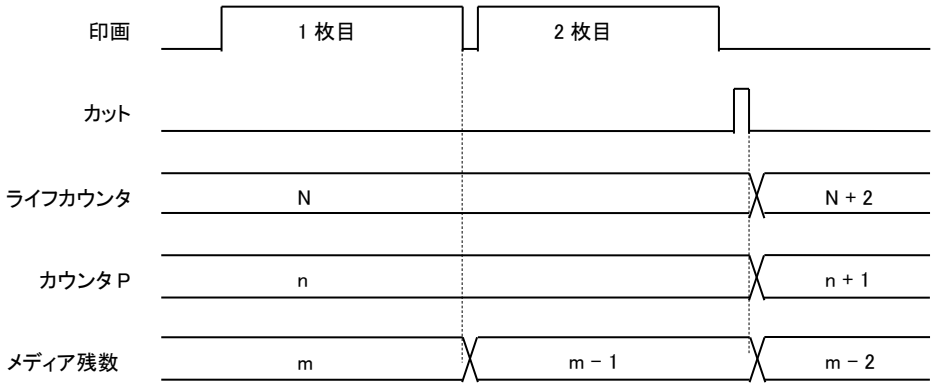
（Ⅱ） コマンド発行後に印画データ 2 枚分を送信して下さい。



④ ライフカウンタ動作

・2 画面連続印画の場合、ライフカウンタは 2 枚目を印画して、印画物のカットが完了した時点で+2 します。

・カウンタ P は、2 枚目を印画して印画物のカットが完了した時点で+1 します。



⑤ 注意事項

・2 画面連続印画を行うときは、メディアの残数をプリンターから取得して、2 枚分の印画が出来ることを確認してから実行してください。（メディア残数は、プリンターのカバー開閉などで実際の枚数と違ってくる可能性がありますので、注意が必要です。）

・2 画面連続印画の 1 枚目の印画後にリボンエンドとなった場合には、プリンターは 1 枚目をカットして停止します（プリンターのステータスはリボンエンドになります）。

・2 画面連続印画の 2 枚目のデータが送信されてこない場合には、約 20 秒で印画物をカットして停止します（プリンターのステータスはアイドルになります）。

(3) 部分マット印画について

① 部分マット概要

・ホストから部分マットモードの指示コマンドを送信することで、プリンターは部分マットモードになり、コマンド送信後に部分マットのパターンを送信し、その後に印画画像データを送信することで、部分マットの印画を行います。

・部分マットモードの指定は、DP-TC10 の場合、部分マットの濃さを 5 段階で指定できます。DP-DS620 の場合、マットパターンをマットとファインマットから選択できます。

・部分マットモードは、1 枚印画を行うと通常のモードに戻ります。部分マットの印画を行うときは、1 枚毎にコマンドと部分マットのパターンを送信して下さい。

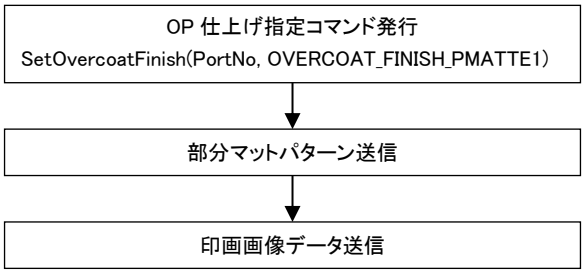
・部分マットパターンのデータ送信は、白黒で作成したパターンを、通常の印画データ送信と同じ処理でプリンターへ送信して下さい。

② 部分マットパターンについて

（Ⅰ）部分マットの印画を行う場合、まずオーバーコート仕上げ制御コマンド SetOvercoatFinish( )で DP-TC10 の場合 OVERCOAT_FINISH_PMATTE1、DP-DS620 の場合 OVERCOAT_FINISH_PMATTE11～13 のいずれかを指定してください。

（Ⅱ）部分マットのパターンを通常の印画データと同じように、SendImageData( )でプリンターへ送信して下さい。

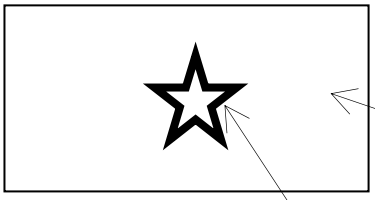
（Ⅲ）部分マットパターンに続いて印画データを送信してください



③ 部分マット指定方法

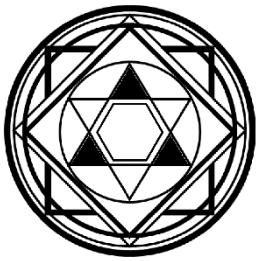
部分マットのパターンは白黒の画像で作成して下さい。

（ダイレクト DLL へ渡す画像フォーマットは RGB24bit カラービットマップフォーマットにしてください）



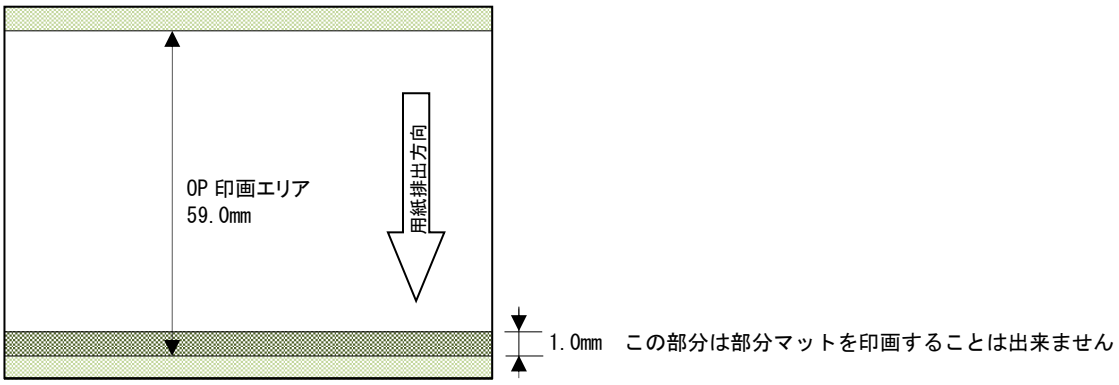
④ 弊社確認パターン

本部分マットモードは、下図の様な図形で、印画率 8～10%を想定しています。



⑤ 制約事項

部分マットパターン印画時は、OP の印画終端部分に部分マットパターンを印画することは出来ません。(DP-TC10)



(4)DP-DS620、DP-DS820　リボンエンドチェック動作

これまでの機種のリボンは、各色パネル先頭に黒線が設けてあり、OP 印画後のリボン巻き取り時に、次のイエロー(Y)パネルの黒線が検出できるか否かによってリボンエンドを判断します。リボンのエンド部は、次の Y パネル黒線をリボンセンサーで検出できない位置関係で製造されています(OP から Y までのピッチは長めに設計されている)。

一方、DP-DS620,DP-DS820 のリボンは、各色のパネル検出用の黒線がなく(パネルの色を検出)、更にパネルピッチを短縮した設計となっており、リボンのエンド部も、次の Y パネルがリボンセンサーで検出できる位置関係にあります。その3ため、DP-DS620、DP-DS820 では、リボンの RF-ID タグに記録されているリボン残り枚数で判断します。残り枚数が 0 でリボンエンドとなります。

通常は、実際のリボン残り枚数と、リボン RF-ID タグに記録されているリボン残り枚数は一致していますが、印画途中でジャムなどが発生し、実際のリボンの枚数が RF-ID タグの枚数より 1 枚少なくなった場合、実際のリボンが最後の 1 枚であっても、次の Y パネルを検出した場合、RF-ID タグの枚数は 1 のため、リボンエンドになりません。

これを避けるため、印画後の RF-ID タグのリボン枚数が以下の条件に限り、リボンエンドをチェックする動作を行います。

＜リボンエンドチェック条件＞

	RF-ID タグのリボン枚数(OP 印画後)	
メディアサイズ	関数:CvGetMediaCounter() / GetMediaCounter()	関数:GetMediaCounterH()
5x3.5 (L)	2 または 1	-
6x4 (PC)	2 または 1	-
5x7 (2L)	2 または 1	4 または 2 (5x3.5 換算枚数)
6x8 (A5)	2 または 1	4 または 2 (6x4 換算枚数)
6x9 (A5W) *1	2 または 1	4 または 2 (6x4.5 換算枚数)
8x10	2 または 1	4 または 2 (8x5 換算枚数)
8x12	2 または 1	4 または 2 (8x6 換算枚数)
A4	2 または 1	4 または 2 (A5 換算枚数)

*1 DP-DS620 ファームウェアバージョン 1.10 以降

OP 印画後の RF-ID タグのリボン枚数が上記の値のときに、下記のリボンエンドチェック動作が入ります。

＜リボンエンドチェック動作＞

OP 印画後、次の Y パネルをリボンセンサーで検出するまでリボンを巻き上げます。

リボンセンサーが Y パネルを検出した後も、リボンエンドチェックとして、更に規定量(約 20mm)リボン巻き上げを継続します。なお、このときのリボン巻き上げは、リボン供給軸とリボンフィルムとの接着部の剥がれを防止するために、低速で行います。

規定量巻き上げることができら、リボンを巻き戻して Y パネル位置出しを行い、動作を継続します。

規定量巻き上げられなかった場合(巻き上げ動作が停止)、リボンエンドで動作を停止します。

(5)余白ありパノラマ印画

① 概要

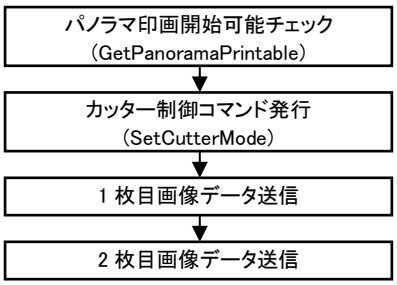
・ ホストからのコマンド指示により、パノラマ印画を行います。

・ プリンターは、カッター制御コマンド(パノラマ印画動作)を受信すると、印画終了後に印画物をカットせず、次に受信したデータを続けて印画します。2 枚目の画像データ送信前に再度パノラマ印画指示コマンドを送信すると、2 枚目印画後もカットせずに次に受信したデータを続けて印画します。

② パノラマ印画処理フロー

1. パノラマ印画を行う場合、パノラマ印画開始可能チェック関数でプリンターがパノラマ印画を実行可能か確認することが出来ます。(省略可能。ヘッド温度が高い場合は、パノラマ印画可能状態になるのを待たずに、パノラマ印画データを送信してください。)
2. 画像データを送信する前にカッター制御コマンドでパノラマ印画動作「CUTTER_MODE_PANORAMA」を指定してください。
3. コマンド発行後に 6x8、8x10、8x12、A4 サイズの印画データを送信してください。

■6x16、8x20、8x24、A4x2 印画の場合

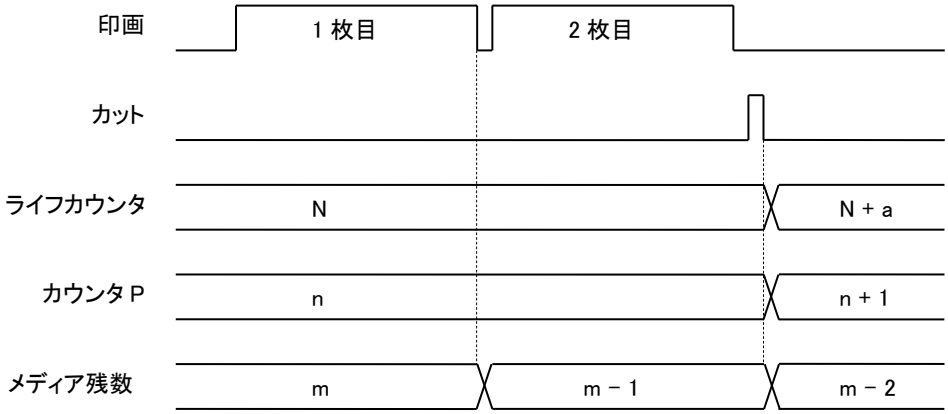


■6x24、8x30、8x36、A4x3 印画の場合

③ ライフカウンタ動作

・ パノラマ印画の場合、印画が終了して印画物をカットした時点でカウントアップします。
(6x16 の場合:+4、6x24 の場合:+6、8x20、8x24、A4x2 の場合:+2、8x30、8x36、A4x3 の場合:+3)

・ カウンタ P は、印画が終了して印画物をカットした時点で+1 します。



DP-DS620 : a=4
DP-DS820 : a=2

④ 注意事項

- ヘッド温度が高い場合は、プリンターは温度が下がるのを待って印画を開始します。
- パノラマ印画を行うときは、メディアの残数をプリンターから取得して、パノラマ印画を行う分のメディアが残っていることを確認してから実行してください。
- パノラマ印画の途中でリボンエンドとなった場合には、プリンターは印画物をカットして停止します。プリンターのステータスはリボンエンドになります。
- パノラマ印画の途中で次の印画データが送信されてこない場合には、プリンターは次のデータが送信されてくるまで待機します。その場合、デカール部での跡と、用紙をグリップしている跡が印画物に付く可能性がありますので、ご注意ください。
- パノラマ印画の途中で、次の印画データが送信されてこない状態が約 60 秒続いた場合には、印画物をカットして停止し、プリンターのステータスはアイドルになります。
- パノラマ印画中は、印画リトライ制御が無効になります。
- パノラマ印画中に、フルカッターセットアップコマンド、またはカット屑なしコマンド、2 インチカットコマンドを受信した場合には、パノラマ印画中の印画物をカットして、通常動作に戻ります。
- パノラマ印画中に 6x8 サイズ(DP-DS820:8x10, 8x12, A4)以外のデータを受信した場合には、パノラマ印画中の印画物をカットして、通常動作に戻ります。
- パノラマ印画の最中にプリンターの出口から出ている用紙には触れないでください。
- 最大で約 36 インチの長さで排出しますので、プリンターの設置場所に注意してください。