

# API

1. コマンド	2
1-1. 実行コマンド文字列	2
1-2. 実行オプション	2
1-3. データソース	2
2. API	5
2-1. 実行プログラムの起動 (EXE) - Windows 環境	5
2-2. 関数の実行 (DLL) - Windows 環境	6
2-2-1. Cast	6
2-2-2. Screen	7
2-2-3. Print	9
2-2-4. PrintStage	11
2-2-5. PrintStage Web	12
2-2-6. Expage	13
2-3. 関数の実行 (.so) - Linux/UNIX 環境	15
2-3-1. Cast	15
2-3-2. Screen	16
2-3-3. Print	17
2-3-4. PrintStage Web	17
2-3-5. Expage	18
3. API の戻り値	20
4. 実行オプション	21
4-1. Print/PrintStage 製品	21
4-1-1. Windows 版 /UNIX 版共通	21
4-1-2. Windows 版のみ	22
4-1-3. オプション補足説明	22
4-1-4. 印刷オプション使用例	25
4-2. PrintStage Web 製品	25
4-2-1. Windows 版 /UNIX 版共通	25
4-2-2. オプション補足説明	25
4-3. Cast 製品	26
4-3-1. Windows 版 /UNIX 版共通	26
4-3-2. Windows 版のみ	27
4-3-3. オプション補足説明	27
4-3-4. 実行オプション使用例	30
4-4. Screen 製品	30
4-4-1. 各種オプション説明	30
4-4-2. Windows 版のみ	32
4-4-3. オプション補足説明	32
4-4-3. 実行オプション使用例	33
4-5. Expage 製品	34
4-5-1. 各種オプション説明	34
4-5-2. Windows 版のみ	34
4-5-3. 実行オプション使用例	34

## 1. コマンド

### 1-1. 実行コマンド文字列

ランタイムを起動するには、実行プログラムや実行 API に引数として実行コマンド文字列を渡す必要があります。実行コマンド文字列は以下のような構成となります。

**実行コマンド文字列 = 実行オプション + データソース (ファイルや DB)**

### 1-2. 実行オプション

実行オプションは、[-][オプション文字列][パラメータ]の形でさまざまな設定を組み合わせ指定します。実行オプションの間は半角スペースで区切ります。

※実行オプションの詳細については「4. 実行オプション」で説明します。

-D 帳票資源ファイルが存在する作業ディレクトリ名

-s 実行ジョブファイル名

-o 出力ファイル名

[Windows]

(例) -D:C:\ProgramData\Infotec\CreateForm\11\work\sample\04\_名簿・一覧 -ssheet.sty  
-osheet\_p1.pdf sheet\_p1.csv

[Linux/UNIX]

(例) -D/opt/createv11/sample/04\_meibo -ssheet.sty -osheet\_p1.pdf sheet\_p1.csv

<< 注意 >>

Windows 環境において作業ディレクトリ名や、出力ファイル名などファイルのパスに半角スペースが含まれる場合は、パスをダブルクォーテーションで囲ってください。

(例) -D"C:\Program Files\Infotec\CreateForm\11\work\workdir" -ssheet.sty  
-osheet\_p1.pdf "C:\Program Files\Infotec\CreateForm\11\workdir\sheet\_p1.csv"

### 1-3. データソース

#### テキストファイル指定

CSV、TXT、XML といったデータファイルは、実行オプションの最後にファイル名を指定します。ランタイム実行時にデータファイルがカレントディレクトリにない場合には、ファイル名をフルパスで指定します。

#### DB データソースの指定

データベースをデータソースとして使用する場合、データファイルの代わりにデータソース名、ユーザ名、パスワードを指定します。指定方法は、以下のとおりです。

```
driver:dsn(#id):usr;pwd(:;sql)
```

(DB 種別 [ コロン ] データソース名 #id [ セミコロン ] ユーザ名 [ セミコロン ] パスワード [ セミコロン ] SQL ファイル名)

driver :

DB 種別 : データベースの種別を指定します。

ODBC 接続は [odbc]、Oracle 接続は [oracle] を指定します。

dsn :

データソース : データソース名を指定します。同一データソース名で別のデータソースとみなしたい場合は、データソース名の後に “#” を付加し、続けて任意の ID 文字列を付加します。同一データソース名で同一 ID の場合は、同一データソースとして扱われます。また、ID を省略した場合も同一データソース名は同一データソースとして扱われます。

usr :

ユーザ名 : ユーザ名を指定します。

pwd :

パスワード : パスワードを指定します。

sql :

SQL ファイル名 : 実行したい SQL を記述したファイルを指定します。(省略可能)

SQL ファイル名を省略した場合、Datamap ファイルに記述されている SQL 文が使用されます。

データファイル以外のコマンドラインオプションは、通常どおり使用できます。

[Windows]

(例) C:\cast -DProgramData\Infotec\CreateForm\11\work\sample -ssample.sty -ooutput.pdf odbc:test#1;usr;pwd:C:\ProgramData\Infotec\CreateForm\11\work\sample\sql.sql

[Linux/UNIX]

(例) ccast -D/opt/createv11/sample -ssample.sty -ooutput.pdf oracle:test#1;usr;pwd:/opt/createv11/sample/sql.sql

<< 注意 >>

Linux/UNIX 環境において DB データソースを利用してランタイムを実行する際、文字化けが発生する可能性があります。

※ Linux/UNIX 環境における DB データソースの利用についてはマネージャのメニュー [ヘルプ]-[オンラインマニュアル] から「1. インストール」-「1.6 導入環境の設定」-「データベースを使用するときの設定」をご覧ください。

### Oracle データベースの使用について

Create!Form ランタイムの実行時に Oracle データベースを使用するには、ランタイム実行環境に Oracle Database Client が導入されている必要があります。

### マルチデータソース機能利用時のデータソース指定

マルチデータソース機能を利用する場合は、複数のデータソースを指定します。  
その際は、利用データソースの順番通りに半角スペースで区切って列挙します。

(例) TextA.txt TextB.csv TextC.csv

データソースをマージして指定する場合は、データソースの順番通りにカンマで区切って列挙します。

(例) TextA.txt TextB.csv,TextC.csv

この場合、TextB.csv と TextC.csv がマージ出力されます。

※マルチデータソースに関してはマネージャのメニュー [ヘルプ]-[オンラインマニュアル] から「3. 機能リファレンス」-「3.5 改ページとフォームの切り替え」-「マルチデータソース」をご覧ください。

## 2. API

### 2-1. 実行プログラムの起動 (EXE) - Windows 環境

各ランタイム製品は実行プログラムからのランタイム実行に対応しています。  
以下の実行プログラムに実行オプションとデータソースなどの文字列を作成して Windows のプログラム実行関数を使用することで、簡単にランタイムの実行を行うことができます。

```
Cast      : CCast.exe
Screen    : CScreen.exe
Print     : CPrint.exe
PrintStage : CPrintST.exe
ExpPage   : CExpPage.exe
```

ここでは Cast ランタイムの実行プログラム "CCast.exe" を例に説明します。

#### ■ C プログラムで Windows の API "WinExec()" を使用した場合

```
WinExec( "CCast -DC:¥¥ProgramData¥¥Infotec¥¥CreateForm¥¥11¥¥work¥¥workdir -ssheet.
sty -osheet.pdf C:¥¥ProgramData¥¥Infotec¥¥CreateForm¥¥11¥¥work¥¥workdir¥¥sheet_
p1.csv", SW_SHOW );
```

#### ■ Visual Basic のプログラム実行関数 "Shell()" を使用した場合

```
Shell( "CCast -DC:¥ProgramData¥Infotec¥CreateForm¥11¥work¥workdir -ssheet.sty
-osheet.pdf C:¥ProgramData¥Infotec¥CreateForm¥11¥work¥workdir¥sheet_p1.csv", 1 )
```

#### 戻り値

正常終了の場合は 1 が返されます。

警告・エラーの場合は 1 以外のエラー値が返されます。

※ WinExec()、Shell() の詳細や実行時の状態設定などはそれぞれのマニュアルをご覧ください。なお、呼び出しを行うアプリケーションの言語によっては、上記と同じ関数を提供していない場合があります。その場合は、それぞれの環境において相当するプログラム実行関数を使用してください。

#### << 注意 >>

これらの実行プログラムからのランタイム実行では非同期的に処理が実行される場合があります。これはプログラム実行関数の終了と、プログラム実行関数から起動したプログラムの終了が同じでないということです。印刷ジョブの順序などが重要な場合など、同期処理による印刷実行を行う場合は後述の関数の実行をご覧ください。

## 2-2. 関数の実行 (DLL) - Windows 環境

各ランタイム製品は関数の実行によるランタイム実行に対応しています。  
以下に各ランタイムの関数実行で使用するファイルを説明します。

### 2-2-1. Cast

Cast ランタイムの関数実行には次のファイルを利用します。

- CCast.h
- CCast.lib
- CCast.dll

CCast.dll はプログラムフォルダ直下に、その他はプログラムフォルダの下の "lib" フォルダに用意されています。

※ "CCast.h" および "CCast.lib" はアプリケーション開発時に使用し、"CCast.dll" は実行時に使用されます。

以下に関数の詳細を記述します。ここに記載した関数説明とプロトタイプ宣言は "CCast.h" に記述されています。

CCast  
CCastV  
CCastS

#### 書式 (C/C++)

```
int CCast ( HWND hWnd, UINT uCmdShow, LPSTR lpTitle, LPSTR lpCommand )
int CCastV ( HWND hWnd, UINT uCmdShow, LPSTR lpTitle, LPSTR lpCommandV, ... )
int CCastS ( LPSTR lpCommand )
```

#### 引数

hWnd

HWND: オーナーアプリケーションのウィンドウハンドルを指定します。この引数が NULL の場合、処理進行ウィンドウは親ウィンドウを持ちません。

uCmdShow

UINT: 処理進行ウィンドウとエラー確認ダイアログの表示方法を以下の値を組み合わせて設定します。

値	意味
CPW_SHOW	処理進行ウィンドウを中央に表示します。
CPW_HIDE	処理進行ウィンドウを表示しません。
CPW_MINIMIZE	処理進行ウィンドウを左下に最小化して表示します。

ERW_SHOW	エラー発生時に確認ダイアログを表示します。
ERW_HIDE	エラーが発生しても確認ダイアログを表示しません。

### lpTitle

LPSTR: 処理進行ウィンドウに表示するタイトルを示す NULL で終わる文字列のポインタを指定します。長さ 0 の文字列を指定した場合、処理中のテキストファイル名が表示されます。表示される文字列は 32 バイトまでで、それ以上の長さの文字列を指定すると 32 バイトに切り詰められます。

### lpCommand

LPSTR: ランタイムの実行コマンド文字列を示す NULL で終わる文字列のポインタを指定します。

### lpCommandV

LPSTR: ランタイムの実行コマンド文字列を示す NULL で終わる文字列のポインタを指定します。可変個数で指定できます。

## 戻り値

int: 正常終了の場合は 1 が返されます。  
エラーの場合は 1 以外のエラー値が返されます。

## 説明

Cast ランタイムの出力処理を実行します。

CCast 関数はオプションを一つの文字列にして実行します。CCastV 関数は各オプションを一つの引数として与える可変引数の関数です。CCastS 関数は実行コマンド文字列のみで実行できる簡易版です。

CCastS( szCommand ) は、CCast( NULL, CPW\_HIDE | ERW\_HIDE, "", szCommand ) と指定することと同等です。

lpCommand にオプションとテキストファイル名を組み合わせた実行コマンド文字列を指定します。

オプションごとに一つの引数を用いて可変個数のオプションを指定する場合には CCastV 関数を用います。オプションの数によって指定する lpCommandV の個数は変わります。オプションの指定順序は問いませんが、最後の lpCommandV は必ずテキストファイル名を指定するようにします。

## 2-2-2. Screen

Screen ランタイムの関数実行には次のファイルを利用します。

- CScreen.h
- CScreen.lib
- CScreen.dll

CScreen.dll はプログラムフォルダ直下に、その他はプログラムフォルダの下の "lib" フォルダに用意されています。

※ "CScreen.h" および "CScreen.lib" はアプリケーション開発時に使用し、"CScreen.dll" は実行時に使用されます。

以下に関数の詳細を記述します。ここに記載した関数説明とプロトタイプ宣言は "CScreen.h" に記述されています。

CScreen

CScreenV

CScreenS

---

**書式 (C/C++)**

```
int CScreen ( HWND hWnd, UINT uCmdShow, LPSTR lpTitle, LPSTR lpCommand )
int CScreenV ( HWND hWnd, UINT uCmdShow, LPSTR lpTitle, LPSTR lpCommandV, ... )
int CScreenS ( LPSTR lpCommand )
```

---

**引数****hWnd**

HWND: オーナーアプリケーションのウィンドウハンドルを指定します。この引数が NULL の場合、処理進行ウィンドウは親ウィンドウを持ちません。

**uCmdShow**

UINT: 処理進行ウィンドウとエラー確認ダイアログの表示方法を以下の値を組み合わせて設定します。

値	意味
CPW_SHOW	処理進行ウィンドウを中央に表示します。
CPW_HIDE	処理進行ウィンドウを表示しません。
CPW_MINIMIZE	処理進行ウィンドウを左下に最小化して表示します。
ERW_SHOW	エラー発生時にエラー確認ダイアログを表示します。
ERW_HIDE	エラーが発生してもエラー確認ダイアログを表示しません。

**lpTitle**

LPSTR: 処理進行ウィンドウに表示するタイトルを示す NULL で終わる文字列のポインタを指定します。長さ 0 の文字列を指定した場合、処理中のテキストファイル名が表示されます。表示される文字列は 32 バイトまでで、それ以上の長さの文字列を指定すると 32 バイトに切り詰められます。

**lpCommand**

LPSTR: ランタイムの実行コマンド文字列を示す NULL で終わる文字列のポインタを指定します。

**lpCommandV**

LPSTR: ランタイムの実行コマンド文字列を示す NULL で終わる文字列のポインタを指定します。可変個数で指定できます。

**戻り値**

int: 正常終了の場合は 1 が返されます。  
エラーの場合は 1 以外のエラー値が返されます。

## 説明

Screen ランタイムの出力処理を実行します。

GScreen 関数はオプションを一つの文字列にして実行します。GScreenV 関数は各オプションを一つの引数として与える可変引数の関数です。GScreenS 関数は実行コマンド文字列のみで実行できる簡易版です。

GScreenS( szCommand ) は、GScreen( NULL, CPW\_SHOW | ERW\_SHOW , "", szCommand ) と指定することと同等です。

lpCommand にオプションとテキストファイル名を組み合わせた実行コマンド文字列を指定します。

オプションごとに一つの引数を用いて可変個数のオプションを指定する場合には GScreenV 関数を用います。オプションの数によって指定する lpCommandV の個数は変わります。オプションの指定順序は問いませんが、最後の lpCommandV は必ずテキストファイル名を指定するようにします。

## 2-2-3. Print

Print ランタイムの関数実行には次のファイルを利用します。

- CPrint.h
- CPrint.lib
- CPrint.dll

CPrint.dll はプログラムフォルダ直下に、その他はプログラムフォルダの下の "lib" フォルダに用意されています。

※ "CPrint.h" および "CPrint.lib" はアプリケーション開発時に使用し、"CPrint.dll" は実行時に使用されます。

以下に関数の詳細を記述します。ここに記載した関数説明とプロトタイプ宣言は "CPrint.h" に記述されています。

CPrint  
CPrintV  
CPrintS

---

### 書式 (C/C++)

```
int CPrint ( HWND hWnd, UINT uCmdShow, LPSTR lpTitle, LPSTR lpCommand )  
int CPrintV ( HWND hWnd, UINT uCmdShow, LPSTR lpTitle, LPSTR lpCommandV, ... )  
int CPrintS ( LPSTR lpCommand )
```

---

### 引数

hWnd

HWND: オーナーアプリケーションのウィンドウハンドルを指定します。この引数が NULL の場合、処理進行ウィンドウは親ウィンドウを持ちません。

**uCmdShow**

UINT: 処理進行ウィンドウとエラー確認ダイアログの表示方法を以下の値を組み合わせて設定します。

値	意味
CPW_SHOW	処理進行ウィンドウを中央に表示します。
CPW_HIDE	処理進行ウィンドウを表示しません。
CPW_MINIMIZE	処理進行ウィンドウを左下に最小化して表示します。
ERW_SHOW	エラー発生時にエラー確認ダイアログを表示します。
ERW_HIDE	エラーが発生してもエラー確認ダイアログを表示しません。

**lpTitle**

LPSTR: 処理進行ウィンドウに表示するタイトルを示す NULL で終わる文字列のポインタを指定します。長さ 0 の文字列を指定した場合、処理中のテキストファイル名が表示されます。表示される文字列は 32 バイトまでで、それ以上の長さの文字列を指定すると 32 バイトに切り詰められます。

**lpCommand**

LPSTR: ランタイムの実行コマンド文字列を示す NULL で終わる文字列のポインタを指定します。

**lpCommandV**

LPSTR: ランタイムの実行コマンド文字列を示す NULL で終わる文字列のポインタを指定します。可変個数で指定できます。

**戻り値**

int: 正常終了の場合は 1 が返されます。  
エラーの場合は 1 以外のエラー値が返されます。

**説明**

Print ランタイムの印刷処理を実行します。

CPrint 関数はオプションを一つの文字列にして実行します。CPrintV 関数は各オプションを一つの引数として与える可変引数の関数です。CPrintS 関数は実行コマンド文字列のみで実行できる簡易版です。

CPrintS( szCommand ) は、CPrint( NULL, CPW\_SHOW | ERW\_SHOW , "", szCommand ) と指定することと同等です。

lpCommand にオプションとテキストファイル名を組み合わせた実行コマンド文字列を指定します。

オプションごとに一つの引数を用いて可変個数のオプションを指定する場合には CPrintV 関数を用います。オプションの数によって指定する lpCommandV の個数は変わります。オプションの指定順序は問いませんが、最後の lpCommandV は必ずテキストファイル名を指定するようにします。

## 2-2-4. PrintStage

PrintStage ランタイムの関数実行には次のファイルを利用します。

- ・ CPrintST. h
- ・ CPrintST. lib
- ・ CPrintST. dll

CPrintST. dll はプログラムフォルダ直下に、その他はプログラムフォルダの下の "lib" フォルダに用意されています。

※ "CPrintST. h" および "CPrintST. lib" はアプリケーション開発時に使用し、"CPrintST. dll" は実行時に使用されます。

以下に関数の詳細を記述します。ここに記載した関数説明とプロトタイプ宣言は "CPrintST. h" に記述されています。

CPrintST  
CPrintSTV  
CPrintSTS

### 書式 (C/C++)

```
int CPrintST ( HWND hWnd, UINT uCmdShow, LPSTR lpTitle, LPSTR lpCommand )
int CPrintSTV ( HWND hWnd, UINT uCmdShow, LPSTR lpTitle, LPSTR lpCommandV, ... )
int CPrintSTS ( LPSTR lpCommand )
```

### 引数

hWnd

HWND: オーナーアプリケーションのウィンドウハンドルを指定します。この引数が NULL の場合、処理進行ウィンドウは親ウィンドウを持ちません。

uCmdShow

UINT: 処理進行ウィンドウとエラー確認ダイアログの表示方法を以下の値を組み合わせで設定します。

値	意味
CPW_SHOW	処理進行ウィンドウを中央に表示します。
CPW_HIDE	処理進行ウィンドウを表示しません。
CPW_MINIMIZE	処理進行ウィンドウを左下に最小化して表示します。

ERW_SHOW	エラー発生時にエラー確認ダイアログを表示します。
ERW_HIDE	エラーが発生してもエラー確認ダイアログを表示しません。

lpTitle

LPSTR: 処理進行ウィンドウに表示するタイトルを示す NULL で終わる文字列のポインタを指定します。長さ 0 の文字列を指定した場合、処理中のテキストファイル名が表示されます。表示される文字列は 32 バイトまでで、それ以上の長さの文字列を指定すると 32 バイトに切り詰められます。

**lpCommand**

LPSTR: ランタイムの実行コマンド文字列を示す NULL で終わる文字列のポインタを指定します。

**lpCommandV**

LPSTR: ランタイムの実行コマンド文字列を示す NULL で終わる文字列のポインタを指定します。可変個数で指定できます。

**戻り値**

int: 正常終了の場合は 1 が返されます。  
エラーの場合は 1 以外のエラー値が返されます。

**説明**

PrintStage ランタイムの印刷処理を実行します。

CPrintST 関数はオプションを一つの文字列にして実行します。CPrintSTV 関数は各オプションを一つの引数として与える可変引数の関数です。CPrintSTS 関数は実行コマンド文字列のみで実行できる簡易版です。

CPrintSTS( szCommand ) は、CPrintST( NULL, CPW\_SHOW | ERW\_SHOW , "", szCommand ) と指定することと同等です。

lpCommand にオプションとテキストファイル名を組み合わせた実行コマンド文字列を指定します。

オプションごとに一つの引数を用いて可変個数のオプションを指定する場合には CPrintSTV 関数を用います。オプションの数によって指定する lpCommandV の個数は変わります。オプションの指定順序は問いませんが、最後の lpCommandV は必ずテキストファイル名を指定するようにします。

**2-2-5. PrintStage Web**

PrintStage Web ランタイムの関数実行には次のファイルを利用します。

- ・ CPrintST.h
- ・ CPrintST.lib
- ・ CPrintST.dll

CPrintST.dll はプログラムフォルダ直下に、その他はプログラムフォルダの下の "lib" フォルダに用意されています。

※ "CPrintST.h" および "CPrintST.lib" はアプリケーション開発時に使用し、"CPrintST.dll" は実行時に使用されます。

以下に関数の詳細を記述します。ここに記載した関数説明とプロトタイプ宣言は "CPrintST.h" に記述されています。

**CPCompress****書式 (C/C++)**

int CPCompress ( HWND hWnd, UINT uCmdShow, LPSTR lpTitle, LPSTR lpCommand )

## 引数

### hWnd

HWND: オーナーアプリケーションのウィンドウハンドルを指定します。この引数が NULL の場合、処理進行ウィンドウは親ウィンドウを持ちません。

### uCmdShow

UINT: 処理進行ウィンドウとエラー確認ダイアログの表示方法を以下の値を組み合わせて設定します。

値	意味
CPW_SHOW	処理進行ウィンドウを中央に表示します。
CPW_HIDE	処理進行ウィンドウを表示しません。
CPW_MINIMIZE	処理進行ウィンドウを左下に最小化して表示します。
ERW_SHOW	エラー発生時にエラー確認ダイアログを表示します。
ERW_HIDE	エラーが発生してもエラー確認ダイアログを表示しません。

### lpTitle

LPSTR: 処理進行ウィンドウに表示するタイトルを示す NULL で終わる文字列のポインタを指定します。長さ 0 の文字列を指定した場合、処理中のテキストファイル名が表示されます。表示される文字列は 32 バイトまでで、それ以上の長さの文字列を指定すると 32 バイトに切り詰められます。

### lpCommand

LPSTR: ランタイムの実行コマンド文字列を示す NULL で終わる文字列のポインタを指定します。

## 戻り値

int: 正常終了の場合は 1 が返されます。  
エラーの場合は 1 以外のエラー値が返されます。

## 説明

PrintStage Web ランタイムの ccd ファイル生成を行います。

## 2-2-6. Expage

Expage ランタイムの関数実行には次のファイルを利用します。

- CExpage. h
- CExpage. lib
- CExpage. dll

CExpage. dll はプログラムフォルダ直下に、その他はプログラムフォルダの下の "lib" フォルダに用意されています。

※ "CExpage. h" および "CExpage. lib" はアプリケーション開発時に使用し、"CExpage. dll" は実行時に使用されます。

以下に関数の詳細を記述します。ここに記載した関数説明とプロトタイプ宣言は“CExpage.h”に記述されています。

CExpage  
CExpageV  
CExpageS

### 書式 (C/C++)

```
int CExpage ( HWND hWnd, UINT uCmdShow, LPSTR lpTitle, LPSTR lpCommand )
int CExpageV ( HWND hWnd, UINT uCmdShow, LPSTR lpTitle, LPSTR lpCommandV, ... )
int CExpageS ( LPSTR lpCommand )
```

### 引数

hWnd

HWND: オーナーアプリケーションのウィンドウハンドルを指定します。この引数が NULL の場合、処理進行ウィンドウは親ウィンドウを持ちません。

uCmdShow

UINT: 処理進行ウィンドウとエラー確認ダイアログの表示方法を以下の値を組み合わせて設定します。

値	意味
CPW_SHOW	処理進行ウィンドウを中央に表示します。
CPW_HIDE	処理進行ウィンドウを表示しません。
CPW_MINIMIZE	処理進行ウィンドウを左下に最小化して表示します。

ERW_SHOW	エラー発生時にエラー確認ダイアログを表示します。
ERW_HIDE	エラーが発生してもエラー確認ダイアログを表示しません。

lpTitle

LPSTR: 処理進行ウィンドウに表示するタイトルを示す NULL で終わる文字列のポインタを指定します。長さ 0 の文字列を指定した場合、処理中のテキストファイル名が表示されます。表示される文字列は 32 バイトまでで、それ以上の長さの文字列を指定すると 32 バイトに切り詰められます。

lpCommand

LPSTR: ランタイムの実行コマンド文字列を示す NULL で終わる文字列のポインタを指定します。

lpCommandV

LPSTR: ランタイムの実行コマンド文字列を示す NULL で終わる文字列のポインタを指定します。可変個数で指定できます。

### 戻り値

int: 正常終了の場合は 1 が返されます。  
エラーの場合は 1 以外のエラー値が返されます。

## 説明

Expage ランタイムの出力処理を実行します。

CExpage 関数はオプションを一つの文字列にして実行します。CExpageV 関数は各オプションを一つの引数として与える可変引数の関数です。CExpageS 関数は実行コマンド文字列のみで実行できる簡易版です。

CExpageS( szCommand ) は、CExpage( NULL, CPW\_SHOW | ERW\_SHOW , "", szCommand ) と指定することと同等です。

lpCommand にオプションとテキストファイル名を組み合わせた実行コマンド文字列を指定します。

オプションごとに一つの引数を用いて可変個数のオプションを指定する場合には CExpageV 関数を用います。オプションの数によって指定する lpCommandV の個数は変わります。オプションの指定順序は問いませんが、最後の lpCommandV は必ずテキストファイル名を指定するようにします。

## 2-3. 関数の実行 (.so) - Linux/UNIX 環境

各ランタイム製品は関数の実行によるランタイム実行に対応しています。

以下に各ランタイムの関数実行で使用するファイルを説明します。

### 2-3-1. Cast

Cast ランタイムの関数実行には以下のファイルを使用します。

- ・ ccast.h
- ・ libccast.so

ccast.h はプログラムフォルダの下の "lib" フォルダに用意されています。

libccast.so はプログラムフォルダの下の "csys" フォルダに用意されています。

以下に関数の詳細を記述します。ここに記載した関数説明とプロトタイプ宣言は "ccast.h" に記述されています。

### 書式

```
int CCast ( char *pszCommand )
```

### 引数

pszCommand

char \*:Cast のオプション文字列を示す NULL で終わる文字列のポインタを指定します。

### 戻り値

int: 正常終了の場合は 0、警告の場合は 1、エラーの場合は 2 が返されます。

ただし、環境変数 CREATE\_RCODE に "yes" を指定した場合は、正常終了以外の場合に、警告もしくはエラーコードの負数を返します。

※エラーコード一覧に関してはマネージャのメニュー [ヘルプ]-[オンラインマニュアル] から「6. エラーコード」-[6.2 Linux/UNIX] をご覧ください。

**説明**

Cast ランタイムの出力処理を実行します。

CCast 関数はオプションを一つの文字列にして実行します。

pszCommand にオプションとテキストファイル名を組み合わせたオプション文字列を指定します。オプションの指定順序は問いませんが、オプション文字列の最後は必ずテキストファイル名を指定するようにします。

**2-3-2. Screen**

Screen ランタイムの関数実行には以下のファイルを使用します。

- ・ cscreen. h
- ・ libcscreen. so

cscreen. h はプログラムフォルダの下の "lib" フォルダに用意されています。

libcscreen. so はプログラムフォルダの下の "csys" フォルダに用意されています。

以下に関数の詳細を記述します。ここに記載した関数説明とプロトタイプ宣言は "cscreen. h" に記述されています。

**書式**

```
int CScreen ( char *pszCommand )
```

**引数**

pszCommand

char \*:Screen のオプション文字列を示す NULL で終わる文字列のポインタを指定します。

**戻り値**

int: 正常終了の場合は 0、警告の場合は 1、エラーの場合は 2 が返されます。

ただし、環境変数 CREATE\_RCODE に "yes" を指定した場合は、正常終了以外の場合に、警告もしくはエラーコードの負数を返します。

※エラーコード一覧に関してはマネージャのメニュー [ヘルプ]-[オンラインマニュアル] から「6. エラーコード」-[6.2 Linux/UNIX] をご覧ください。

**説明**

Screen ランタイムの出力処理を実行します。

CScreen 関数はオプションを一つの文字列にして実行します。

pszCommand にオプションとテキストファイル名を組み合わせたオプション文字列を指定します。オプションの指定順序は問いませんが、オプション文字列の最後は必ずテキストファイル名を指定するようにします。

### 2-3-3. Print

Print ランタイムの関数実行には以下のファイルを利用します。

- ・ cprint.h
- ・ libcprint.so

cprint.h はプログラムフォルダの下の "lib" フォルダに用意されています。

libcprint.so はプログラムフォルダの下の "bin" フォルダに用意されています。

以下に関数の詳細を記述します。ここに記載した関数説明とプロトタイプ宣言は "cprint.h" に記述されています。

---

#### 書式

```
int CPrint ( char *pszCommand )
```

---

#### 引数

pszCommand

char \*:Print のオプション文字列を示す NULL で終わる文字列のポインタを指定します。

#### 戻り値

int: 正常終了の場合は 0、警告の場合は 1、エラーの場合は 2 が返されます。

ただし、環境変数 CREATE\_RCODE に "yes" を指定した場合は、正常終了以外の場合に、警告もしくはエラーコードの負数を返します。

※エラーコード一覧に関してはマネージャのメニュー [ヘルプ]-[オンラインマニュアル] から「6. エラーコード」-[6.2 Linux/UNIX] をご覧ください。

#### 説明

Print ランタイムの出力処理を実行します。

CPrint 関数はオプションを一つの文字列にして実行します。

pszCommand にオプションとテキストファイル名を組み合わせたオプション文字列を指定します。オプションの指定順序は問いませんが、オプション文字列の最後は必ずテキストファイル名を指定するようにします。

### 2-3-4. PrintStage Web

PrintStage Web ランタイムの関数実行には以下のファイルを利用します。

- ・ cprintst.h
- ・ libcprintst.so

cprintst.h はプログラムフォルダの下の "lib" フォルダに用意されています。

libcprintst.so はプログラムフォルダの下の "bin" フォルダに用意されています。

以下に関数の詳細を記述します。ここに記載した関数説明とプロトタイプ宣言は“cprintst.h”に記述されています。

---

### 書式

```
int CPrintST ( char *pszCommand )
```

---

### 引数

pszCommand

char \*:PrintStage Web のオプション文字列を示す NULL で終わる文字列のポインタを指定します。

### 戻り値

int: 正常終了の場合は0、警告の場合は1、エラーの場合は2が返されます。

ただし、環境変数 CREATE\_RCICODE に“yes”を指定した場合は、正常終了以外の場合に、警告もしくはエラーコードの負数を返します。

※エラーコード一覧に関してはマネージャのメニュー [ヘルプ]-[オンラインマニュアル] から「6. エラーコード」-[6.2 Linux/UNIX] をご覧ください。

### 説明

PrintStage Web ランタイムの ccd ファイル生成処理を行います。

CPrintST 関数はオプションを一つの文字列にして実行します。

pszCommand にオプションとテキストファイル名を組み合わせたオプション文字列を指定します。オプションの指定順序は問いませんが、オプション文字列の最後は必ずテキストファイル名を指定するようにします。

## 2-3-5. Expage

Expage ランタイムの関数実行には以下のファイルを使用します。

- cexpage.h
- libcexpage.so

cexpage.h はプログラムフォルダの下の“lib”フォルダに用意されています。

libcexpage.so はプログラムフォルダの下の“csys”フォルダに用意されています。

以下に関数の詳細を記述します。ここに記載した関数説明とプロトタイプ宣言は“cexpage.h”に記述されています。

---

### 書式

```
int CExpage ( char *pszCommand )
```

---

### 引数

pszCommand

char \*:Expage のオプション文字列を示す NULL で終わる文字列のポインタを指定します。

## 戻り値

int: 正常終了の場合は0、警告の場合は1、エラーの場合は2が返されます。

ただし、環境変数 CREATE\_RCODE に “yes” を指定した場合は、正常終了以外の場合に、警告もしくはエラーコードの負数を返します。

※エラーコード一覧に関してはマネージャのメニュー [ヘルプ]-[オンラインマニュアル] から「6. エラーコード」-[6.2 Linux/UNIX] をご覧ください。

## 説明

Expage ランタイムの出力処理を実行します。

CExpage 関数はオプションを一つの文字列にして実行します。

pszCommand にオプションとテキストファイル名を組み合わせたオプション文字列を指定します。オプションの指定順序は問いませんが、オプション文字列の最後は必ずテキストファイル名を指定するようにします。

### 3. API の戻り値

ランタイム製品の実行結果（正常 / 警告 / エラー）を戻り値で判定することができます。  
各実行環境における実行種別ごとの戻り値は下表の通りです。

#### ■ Windows 環境

実行種別	正常	警告	エラー
EXE	1	-XXXX	-XXXX
DLL	1	-XXXX	-XXXX
Java	0	-XXXX	-XXXX
.Net	1	-XXXX	-XXXX

※「XXXX」は警告やエラーの発生を示すメッセージ番号で、戻り値としては負数となります。  
メッセージ番号については、マネージャのメニュー [ヘルプ]-[オンラインマニュアル] から「6. エラーコード」-「6.1 Windows ランタイム」をご覧ください。

#### ■ UNIX / Linux 環境

実行種別	正常	警告	エラー
コマンド	0	1	2
so	0	1 (※)	2 (※)
Java	0	0 (※)	-2 (※)

※ UNIX/Linux 環境では、環境変数「CREATE\_RCODE」に“yes”を設定した場合、Windows 環境と同様に警告・エラーコードを戻り値として取得することができます。  
(設定例)

```
$ export CREATE_RCODE=yes
```

※ UNIX/Linux 環境の警告・エラーコードについては、マネージャのメニュー [ヘルプ]-[オンラインマニュアル] から「6. エラーコード」-「6.2 Linux/UNIX」をご覧ください。

## 4. 実行オプション

Create!Form ランタイム製品の実行オプションについて説明します。

### 4-1. Print/PrintStage 製品

#### 4-1-1. Windows 版 / UNIX 版共通

オプション	説明
D	: 作業ディレクトリの指定 帳票資源ファイルの場所（ディレクトリ）を指定します。
l	: ライン数 / 頁の指定 テキスト印刷で改ページコードがない場合に使用します。
nc	: コピー枚数の指定 指定しない場合は“-nc1”の指定と同じです。
o	: 出力ファイルの指定 出力をファイルに書き出す場合に使用します。
q	: “.qdf”ファイル（辞書定義ファイルまたはキュー定義ファイルの指定 指定しない場合は“DEFAULT”の指定と同じです。
s	: ジョブファイル(.sty)の指定 指定しない場合は標準ジョブファイル(default.sty)の指定と同じです。
St	: 給紙トレイの指定
So	: 排紙トレイの指定 ※ Print のみ
Ss	: 拡大 / 縮小の指定
Sd	: 両面印刷の指定 Tumble を指定する場合は Sdt を、両面印刷リセットを指定する場合は Sdr を指定します。Tumble と 両面印刷リセットの両方を指定する場合は Sdtr を指定します。
Sm	: 位置の指定 Smx および Smy で指定します。
ix	: XML データマップ出力の指定 XML データマップを用いた出力実行の際には必須の設定です。
g	: 外字の自動出力の指定
pr	: ページデータ切り替え方法の指定
ps	: 印刷開始ページの指定

pe	: 印刷終了ページの指定
pse	: 印刷ページの指定
P	: 戻り値として総ページ数を返す ※ Linux/UNIX でのコマンド実行では使用できない
Pw	: 戻り値として総ページ数を返す (警告無視) ※ Linux/UNIX でのコマンド実行では使用できない
Pn	: 戻り値として総ページ数を返す (印刷データは出力しない) ※ Linux/UNIX でのコマンド実行では使用できない
Pnw	: 戻り値として総ページ数を返す (警告無視 / 印刷データは出力しない) ※ Linux/UNIX でのコマンド実行では使用できない
j	: 印刷ジョブのドキュメント名の指定
Lid	: ジョブ ID の指定
Lfn	: ログファイル名の指定
u	: RPX ファイルの指定 ※ PrintStage のみ

#### 4-1-2. Windows 版のみ

オプション	説明
z	: テキストファイルの強制削除指定 印刷終了後にテキストデータを強制的に削除します。
#	: 出力先の指定 出力させたいプリンタを PRTn もしくはプリンタ名で指定します。 指定しない場合は "DEFAULT" の指定と同じです。 プリンタ名を指定する際、プリンタ名にスペースが含まれる場合は、ダブルクォーテーションで括ってください。 また、PrintStage にてプリンタ名を指定して印刷した際には、Create!Form のトレイ番号と実際のプリンタのトレイ番号が順番通りに関連付いていないため、トレイ選択が有効にならない場合があります。その場合、マネージャ - 「設定」 - 「プリンタ」の「トレイ設定」より実際のプリンタのトレイ番号をご確認ください。

#### 4-1-3. オプション補足説明

##### [o] 出力ファイルの指定

ファイル名がフルパス指定でない場合は、ランタイムが実行されるカレントのディレクトリに出力されます。ただし、マネージャからのテスト実行の場合は、作業ディレクトリ直下に出力されます。

出力されるファイルには EOF コードは付加されません。

**[q]** QDF ファイル（辞書定義ファイル）の指定

通常は指定する必要はありませんが、印刷環境に問題がある場合やバーコード印刷を行う場合など、運用に合わせて辞書定義ファイルを指定します。

通常は標準 QDF ファイルで印刷可能ですので、このオプションを指定する必要はありません。  
※ QDF ファイルの詳細についてはマネージャのメニュー [ヘルプ]-[オンラインマニュアル] から「3. 機能リファレンス」-「3.10 ユーティリティ」-「QDF エディタ」をご覧ください。

**[St]** 給紙トレイの指定

1～6 までの数値で給紙トレイを指定します。手差しトレイを指定する場合は「m」、特殊トレイ 1 を指定する場合は「a」、特殊トレイ 2 を指定する場合は「b」を指定します。

なお、PrintStage 製品の場合はマネージャから起動するプリンタ設定ダイアログで設定されたトレイから給紙されます。

(例) 給紙トレイ 1 を指定 … -St1

**[So]** 排紙トレイの指定

1～24 までの数値で排紙トレイを指定します。

(例) 排紙トレイ 1 を指定 … -So1

**[Ss]** 拡大 / 縮小の指定

倍率を表す数値で拡大・縮小率を指定します。

(例) 80% 指定で縮小印刷 … -Ss80

(例) 120% 指定で拡大印刷 … -Ss120

**[Sd]** 両面印刷の指定

両面印刷を行う場合に使用します。

指定した設定は全てのユニットに適用されます。

-Sd … 両面印刷

-Sdt … 両面印刷 (Tumble)

-Sdr … 両面印刷リセット

-Sdtr … 両面印刷 (Tumble) かつ 両面印刷リセット

**[Sm]** 位置の指定

出力位置の X 座標 (Smx)、もしくは Y 座標 (Smy) をミリ単位で指定します。

Smx と Smy は、同時に設定が可能です。

(例) X 座標を 100、Y 座標を 50 に指定 … -Smx100 -Smy50

※ Y 座標に関しては、Print では設定した値だけ上に、PrintStage では設定した値だけ下に移動します。

**[x2]** 辞書の同時送出手の指定

以前のバージョンの互換のためのオプションですので、通常は使用しないでください。

辞書定義ファイル (\*.QDF) にキーワード "/dictsendeachjob" が記述されている場合はこのオプションは必要ありません。

**[nc]** コピー枚数の指定

non-collated copy です。丁合機能はありません。

**[S]** 拡張オプションの指定

印刷時の拡張オプションは S オプションで指定します。拡張オプションの内容はマネージャのジョブの編集の拡張オプションに準じています。

- [ix] XML データマップ出力の指定  
XML データマップを使用した出力の場合には、必ず実行オプションに“-ix”を指定してください。
- [pr] ページデータ切り替え方法の指定  
ユニット毎（もしくはセット毎）にページデータを切り替える場合に使用します。  
-pr1 … ユニット毎に切り替え  
-pr2 … セット毎に切り替え
- [ps] 印刷開始ページの指定  
印刷の開始ページを指定して印刷します。  
（例）3 ページ目から終了ページまでを印刷 … -ps3
- [pe] 印刷終了ページの指定  
印刷の終了ページを指定して印刷します。  
（例）開始ページから 6 ページ目までを印刷 … -pe6
- [pse] 印刷ページの指定  
指定したページのみを印刷します。  
（例）5 ページ目のみを印刷 … -pse5
- [P] 戻り値として総ページ数を返す  
通常通り印刷実行を行い、ランタイム実行の戻り値として総ページ数を返します。  
ただし、警告およびエラー発生時はエラーコードを返します。
- [Pw] 戻り値として総ページ数を返す（警告無視）  
通常通り印刷実行を行い、ランタイム実行の戻り値として総ページ数を返します。  
ただし、エラー発生時はエラーコードを返します。
- [Pn] 戻り値として総ページ数を返す（印刷データは出力しない）  
印刷実行は行わずに、ランタイム実行の戻り値として総ページ数を返します。  
ただし、警告およびエラー発生時はエラーコードを返します。
- [Pnw] 戻り値として総ページ数を返す（警告無視 / 印刷データは出力しない）  
印刷実行は行わずに、ランタイム実行の戻り値として総ページ数を返します。  
ただし、エラー発生時はエラーコードを返します。
- [j] 印刷ジョブのドキュメント名の指定  
印刷ジョブのドキュメント名を指定して印刷します。  
（例）印刷ジョブのドキュメント名を“document”として印刷 … -jdocument
- [lid] ジョブ ID の指定  
ランタイム実行のジョブ ID を指定します。
- [lfn] ログファイル名の指定  
ランタイム実行時に出力するログファイルをファイル名のみもしくは絶対パスで指定します

## [u] RPX ファイルの指定

RPX ファイル名を指定します。なお、Print では使用することはできません。

## 4-1-4. 印刷オプション使用例

オプションを指定したいいくつかのコマンドライン例を示します。

■テキストファイル document.txt を標準ジョブに従って 1 ページ 80 ラインで印刷します。

```
cprint -l80 document.txt
```

※この場合は、default.qdf と default 作業ディレクトリの default.sty が標準使用されます。

■テキストファイル document.txt を 2 部印刷します。

```
cprint -l80 -nc2 document.txt
```

■作業ディレクトリ wdir1 のジョブ tform.sty でテキストデータ tdata.txt を印刷します。

```
cprint -Dwdir1 -stform.sty tdata.txt
```

■QDF ファイル user の設定でテキストデータ tdata.txt を印刷します。

```
cprint -quser -Dwdir1 -stform.sty tdata.txt
```

■ジョブ form\_x.sty でテキストデータ text.nnn を [プリンタの設定] で PRT5 に設定された出力先に印刷します。(Windows のみ)

```
cprint -sform_x.sty -#PRT5 text.nnn
```

■作業ディレクトリ wdir1 のジョブ tpages.sty でテキストデータ tdata.txt を使用して 3 ページ目から 4 ページ目までを印刷します。

```
cprint -Dwdir1 -stpages.sty -ps3 -pe4 tdata.txt
```

## 4-2. PrintStage Web 製品

## 4-2-1. Windows 版 /UNIX 版共通

オプション	説明
c[time stamp]	: タイムスタンプによる圧縮ファイルの識別。
cr	: 資源ファイル、およびデータファイルを圧縮。
crb	: 資源ファイル、およびデータファイルを圧縮 (保存禁止)。
cd	: データファイルのみ圧縮。

## 4-2-2. オプション補足説明

[c[time stamp]] タイムスタンプによる圧縮ファイルの識別

各資源ファイルのタイムスタンプとオプションによって指定されたタイムスタンプを比較した結果に応じて圧縮ファイルの形式を変更します。

[timestamp] は “年月日時分秒” で指定します。(c20070110080000)

[cr] 資源ファイル、およびデータファイルを圧縮

資源ファイルとデータソースから作成されたデータファイルを圧縮します。

[crb] 資源ファイル、およびデータファイルを圧縮（保存禁止）  
資源ファイルとデータソースから作成されたデータファイルを圧縮します。  
クライアント側での保存はできません。

[cd] データファイルのみ圧縮  
データソースから作成されたデータファイルを圧縮します。

### 4-3. Cast 製品

#### 4-3-1. Windows 版 /UNIX 版共通

オプション	説明
D	: 作業ディレクトリの指定 帳票資源ファイルの場所（ディレクトリ）を指定します。
l	: ライン数 / 頁の指定 テキスト /CSV データで改ページコードがない場合に使用します。
o	: 出力ファイルの指定 出力をファイルに書き出す場合に使用します。
q	: “.qdf” ファイル（辞書定義ファイルまたはキュー定義ファイル）の指定 指定しない場合は“DEFAULT”の指定と同じです。
s	: ジョブファイル (.sty) の指定 指定しない場合は標準ジョブファイル (default.sty) の指定と同じです。
ix	: XML データマップ出力の指定 XML データマップを用いた出力実行の際には必ず指定してください。
g	: 外字の自動出力の指定
pr	: ページデータ切り替え方法の指定
X	: PDF セキュリティの設定
v	: ビューアプレファレンス設定
ps	: 出力開始ページの指定
pe	: 出力終了ページの指定
pse	: 出力ページの指定
P	: 戻り値として総ページ数を返す ※ Linux/UNIX でのコマンド実行では使用できない
Pw	: 戻り値として総ページ数を返す（警告無視） ※ Linux/UNIX でのコマンド実行では使用できない

Pn	: 戻り値として総ページ数を返す (PDF データは出力しない) ※ Linux/UNIX でのコマンド実行では使用できない
Pnw	: 戻り値として総ページ数を返す (警告無視 / PDF データは出力しない) ※ Linux/UNIX でのコマンド実行では使用できない
Lid	: ジョブ ID の指定
Lfn	: ログファイル名の指定
u	: RPX ファイルの指定
Fo	: PDF オブジェクト圧縮の指定
Ys	: 印刷ダイアログプリセットのページの拡大 / 縮小の指定
Yd	: 印刷ダイアログプリセットの両面印刷モードの指定
Yt	: 印刷ダイアログプリセットのページサイズに合わせて用紙を選択の指定
Yr	: 印刷ダイアログプリセットの印刷範囲のページ指定
Yu	: 印刷ダイアログプリセットの印刷範囲のユニット指定
Yc	: 印刷ダイアログプリセットの部数の指定
fs	: FormSwitch オプションを有効にする
Fts	: PDF ファイルを印刷に適した構文で出力する (表オブジェクトを利用時) ※設定すると、PDF ファイルの生成速度は低下します。

#### 4-3-2. Windows 版のみ

オプション	説明
z	: テキストファイルの強制削除指定 PDF 出力終了後にテキストデータを強制的に削除します。

#### 4-3-3. オプション補足説明

[l] ライン数 / 頁の指定

改ページ行数を表す数値を指定します。

(例) 30 行目で改ページを指定 … -l30

[o] 出力ファイルの指定

ファイル名がフルパス指定でない場合は、ランタイムが実行されるカレントのディレクトリに出力されます。ただし、マネージャからのテスト実行の場合は、作業ディレクトリ直下に出力されます。

[q] QDF ファイルの指定

通常は指定する必要はありませんが、特殊な指定を行う場合に使用します。

[ix] XML データマップ出力の指定

XML データマップを使用した出力の場合には、必ず実行オプションに“-ix”を指定してください。

[pr] ページデータ切り替え方法の指定

ユニット毎（もしくはセット毎）にページデータを切り替えるために使用します。

-pr1 … ユニット毎に切り替え

-pr2 … セット毎に切り替え

[X] PDF セキュリティの設定

PDF セキュリティの設定を行います。

※PDFセキュリティの設定については、マネージャのメニュー[ヘルプ]-[オンラインマニュアル]から「3. 機能リファレンス」-「3.7 PDF 出力設定」-「PDF セキュリティ」をご覧ください。

[v] ビューアプレファレンス設定

PDF ファイルが Adobe Acrobat や Adobe Reader などのビューアで開かれた際に、ビューアの表示状態を指定することが出来ます。

-vum … メニューバーを非表示にする

-vut … ツールバーを非表示にする

-vuw … ウィンドウコントロールを非表示にする

-vwf … ページサイズにウィンドウサイズを合わせる

-vwc … ウィンドウを画面中央に表示する

-vpd … ページ表示順序を右から左に切り替える

※ビューアプレファレンスの設定についてはマネージャのメニュー[ヘルプ]-[オンラインマニュアル]から「3. 機能リファレンス」-「3.7 PDF 出力設定」-「ビューアプリファレンス」をご覧ください。

[ps] 出力開始ページの指定

開始ページを指定して PDF 出力します。

(例)3 ページ目から終了ページまでを出力 … -ps3

[pe] 出力終了ページの指定

終了ページを指定して PDF 出力します。

(例)開始ページから 6 ページ目までを出力 … -pe6

[pse] 出力ページの指定

指定したページのみを PDF 出力します。

(例)5 ページ目のみを出力 … -pse5

[P] 戻り値として総ページ数を返す

通常通り PDF 出力を行い、ランタイム実行の戻り値として総ページ数を返します。

ただし、警告およびエラー発生時はエラーコードを返します。

[Pw] 戻り値として総ページ数を返す（警告無視）

通常通り PDF 出力を行い、ランタイム実行の戻り値として総ページ数を返します。

ただし、エラー発生時はエラーコードを返します。

- [Pn] 戻り値として総ページ数を返す (PDF データは出力しない)  
PDF 出力は行わずに、ランタイム実行の戻り値として総ページ数を返します。  
ただし、警告およびエラー発生時はエラーコードを返します。
- [Pnw] 戻り値として総ページ数を返す (警告無視 / PDF データは出力しない)  
PDF 出力は行わずに、ランタイム実行の戻り値として総ページ数を返します。  
ただし、エラー発生時はエラーコードを返します。
- [Lid] ジョブ ID の指定  
ランタイム実行のジョブ ID を指定します。
- [Lfn] ログファイル名の指定  
ランタイム実行時に出力するログファイルをファイル名のみもしくは絶対パスで指定します
- [u] RPX ファイルの指定  
RPX ファイル名を指定します。
- [Fo] PDF オブジェクト圧縮の指定  
PDF 内のオブジェクトを圧縮して出力します。  
※PDFセキュリティの設定については、マネージャのメニュー [ヘルプ]-[オンラインマニュアル] から「3. 機能リファレンス」-「3.7 PDF 出力設定」-「PDF オブジェクト圧縮」をご覧ください。
- [Ys] 印刷ダイアログプリセットのページの拡大 / 縮小の指定  
PDF ファイルに印刷ダイアログプリセットの“ページの拡大 / 縮小”を指定して出力します。  
-Ys0 … デフォルト  
-Ys1 … なし  
※デフォルトとは、Adobe Acrobat や Adobe Reader などのビューアのデフォルト値 (前回、印刷した時のページの拡大 / 縮小の設定) です。
- [Yd] 印刷ダイアログプリセットの両面印刷モードの指定  
PDF ファイルに印刷ダイアログプリセットの“両面印刷モード”を指定して出力します。  
-Yd0 … デフォルト  
-Yd1 … 両面 (左右に開く)  
-Yd2 … 両面 (上下に開く)  
※デフォルトとは、Adobe Acrobat や Adobe Reader などのビューアのデフォルト値 (前回、印刷した時の用紙の両面に印刷の設定) です。
- [Yt] 印刷ダイアログプリセットのページサイズに合わせて用紙を選択の指定  
PDF ファイルに印刷ダイアログプリセットの“ページサイズに合わせて用紙を選択”を指定して出力します。  
-Yto … チェックする  
-Ytn … デフォルト  
※デフォルトとは、Adobe Acrobat や Adobe Reader などのビューアのデフォルト値 (前回、印刷した時のページサイズに合わせて用紙を選択の設定) です。

## [Yr] 印刷ダイアログプリセットの印刷範囲のページ指定

PDF ファイルに印刷ダイアログプリセットの“印刷範囲のページ”を指定して出力します。

-Yrs ... 印刷開始ページの指定  
 -Yre ... 印刷終了ページの指定  
 -Yrse ... 指定したページのみ印刷指定  
 -Yr ... 印刷ページの範囲指定

(例)5 ページ目から終了ページまでを印刷 ... -Yrs5

## [Yu] 印刷ダイアログプリセットの印刷範囲のユニット指定

PDF ファイルに印刷ダイアログプリセットの“印刷範囲のユニット”を指定して出力します。

-Yus ... 印刷開始ユニットの指定  
 -Yue ... 印刷終了ユニットの指定  
 -Yuse ... 指定したユニットのみ印刷指定  
 -Yu ... 印刷ユニットの範囲指定

(例)3 ユニット目のみを印刷 ... -Yuse3

## [Yc] 印刷ダイアログプリセットの部数指定

PDF ファイルに印刷ダイアログプリセットの“部数”を指定して出力します。

1～5の値を指定します。

(例)3部を印刷 ... -Yc3

## [fs] FormSwitch オプション

入力フォームオブジェクトに設定された実行オプションの内容を有効にして出力を行います。

※印刷ダイアログプリセットの設定についてはマネージャのメニュー [ヘルプ]-[オンラインマニュアル] から「3. 機能リファレンス」-「3.7 PDF 出力設定」-「印刷ダイアログプリセット」をご覧ください。

## 4-3-4. 実行オプション使用例

オプションを指定したいいくつかのコマンドライン例を示します。

- 作業ディレクトリ wdir1 のジョブ tform.sty でデータ tdata.xml の PDF ファイル tdata.pdf を生成します。

```
ccast -Dwdir1 -stform.sty -ix -otdata.pdf tdata.xml
```

- 作業ディレクトリ wdir1 のジョブ tpages.sty でテキストデータ tdata.txt を使用して3ページ目から4ページ目までの PDF ファイル tfile.pdf を生成します。

```
ccast -Dwdir1 -stpages.sty -ps3 -pe4 -otfile.pdf tdata.txt
```

## 4-4. Screen 製品

## 4-4-1. 各種オプション説明

オプション	説明
D	: 作業ディレクトリの指定 帳票資源ファイルの場所 (ディレクトリ) を指定します。
l	: ライン数 / 頁の指定 テキスト / GSV データで改ページコードがない場合に使用します。

o	: 出力ファイルの指定 出力をファイルに書き出す場合に使用します。
s	: ジョブファイル (.sty) の指定 指定しない場合は標準ジョブファイル (default.sty) の指定と同じです。
ix	: XML データマップ出力の指定 XML データマップを用いた出力実行の際には必須の設定です。
g	: 外字の自動出力の指定
pr	: ページデータ切り替え方法の指定
ps	: 出力開始ページの指定
pe	: 出力終了ページの指定
pse	: 出力ページの指定
P	: 戻り値として総ページ数を返す
Pw	: 戻り値として総ページ数を返す (警告無視)
Pn	: 戻り値として総ページ数を返す (HTML データは出力しない)
Pnw	: 戻り値として総ページ数を返す (警告無視 /HTML データは出力しない)
Lid	: ジョブ ID の指定
Lfn	: ログファイル名の指定
wi	: イメージ URL の指定
wf	: フォント URL の指定
wj	: JavaScript URL の指定
wc	: CSS URL の指定
Omb	: 部分出力 -body 以下出力の指定
Omf	: 部分出力 -form 以下出力の指定
fs	: FormSwitch オプションを有効にする
ts	: 単一出力モードの指定
th	: ツールバー表示モードの指定

## 4-4-2. Windows 版のみ

オプション	説明
z	: テキストファイルの強制削除指定 HTML 出力終了後にテキストデータを強制的に削除します。

## 4-4-3. オプション補足説明

- [o] 出力ファイルの指定  
ファイル名がフルパス指定でない場合は、ランタイムが実行されるカレントのディレクトリに出力されます。ただし、マネージャからのテスト実行の場合は、作業ディレクトリ直下に出力されます。
- [ix] XML データマップ出力の指定  
XML データマップを使用した出力の場合には、必ず実行オプションに“-ix”を指定してください。
- [pr] ページデータ切り替え方法の指定  
ユニット毎（もしくはセット毎）にページデータを切り替える場合に使用します。  
-pr1 …… ユニット毎に切り替え  
-pr2 …… セット毎に切り替え
- [ps] 出力開始ページの指定  
開始ページを指定して HTML 出力します。  
(例)3 ページ目から終了ページまでを出力 …… -ps3
- [pe] 出力終了ページの指定  
終了ページを指定して HTML 出力します。  
(例)開始ページから 6 ページ目までを出力 …… -pe6
- [pse] 出力ページの指定  
指定したページのみを HTML 出力します。  
(例)5 ページ目のみを出力 …… -pse5
- [P] 戻り値として総ページ数を返す  
通常通り HTML 出力を行い、ランタイム実行の戻り値として総ページ数を返します。  
ただし、警告およびエラー発生時はエラーコードを返します。
- [Pw] 戻り値として総ページ数を返す（警告無視）  
通常通り HTML 出力を行い、ランタイム実行の戻り値として総ページ数を返します。  
ただし、エラー発生時はエラーコードを返します。
- [Pn] 戻り値として総ページ数を返す（HTML データは出力しない）  
HTML 出力は行わずに、ランタイム実行の戻り値として総ページ数を返します。  
ただし、警告およびエラー発生時はエラーコードを返します。
- [Pnw] 戻り値として総ページ数を返す（警告無視 /HTML データは出力しない）  
HTML 出力は行わずに、ランタイム実行の戻り値として総ページ数を返します。  
ただし、エラー発生時はエラーコードを返します。

**[Lid]** ジョブ ID の指定

ランタイム実行のジョブ ID を指定します。

**[Lfn]** ログファイル名の指定

ランタイム実行時に出力するログファイルをファイル名のみもしくは絶対パスで指定します。

**[wi]** イメージ URL の指定

イメージ URL を指定します。

(例) イメージ URL に「/example/image/」を指定 ... `-wi/example/image/`

**[wf]** フォント URL の指定

フォント URL を指定します。

(例) フォント URL に「/example/font/」を指定 ... `-wf/example/font/`

**[wj]** JavaScript URL の指定

JavaScript URL を指定します。

(例) JavaScript URL に「/example/js/」を指定 ... `-wj/example/js/`

**[wc]** CSS URL の指定

CSS URL を指定します。

(例) CSS URL に「/example/css/」を指定 ... `-wc/example/css/`

**[Omb]** 部分出力 `-body` 以下出力の指定

`<body>` 要素の子要素の部分を出します。

**[Omf]** 部分出力 `-form` 以下出力の指定

`<form>` 要素の子要素の部分を出します。

帳票の外側にサブミットを配置する場合に使用します。

**[fs]** FormSwitch オプション

入力フォームオブジェクトに設定された実行オプションの内容を有効にして出力を行います。

**[ts]** 単一出力モードの指定

複数ページを出力した場合にページ単位で画面が切り替わります。

**[th]** ツールバー表示モードの指定

ブラウザ表示画面上部にツールバーを出力します。

※イメージ URL、フォント URL、JavaScript URL、CSS URL の詳細につきましては、オンラインマニュアルの [3. 機能リファレンス] - [3-8. HTML 設定] の以下の場所を参照してください。

・フォント URL/ イメージ URL/JavaScript URL/CSS URL : 「9. URL 設定」

### 4-4-3. 実行オプション使用例

オプションを指定したいくつかのコマンドライン例を示します。

- 作業ディレクトリ `wdir1` のジョブ `tform.sty` でデータ `tdata.xml` の HTML ファイル `tfile.html` を生成します。

```
cscreen -Dwdir1 -stform.sty -ix -otfile.html tdata.xml
```

- 作業ディレクトリ `wdir1` のジョブ `tpages.sty` でテキストデータ `tdata.txt` を使用して 3 ページ目から 4 ページ目までの HTML ファイル `tfile.html` を生成します。

```
cscreen -Dwdir1 -stpages.sty -ps3 -pe4 -otfile.html tdata.txt
```

- 作業ディレクトリ `wdir1` のジョブ `tpages.sty` でテキストデータ `tdata.csv` を使用して `turl` に配置されている EUDC.TTE を参照させて外字を含む HTML ファイル `tfile.html` を生成します。

```
cscreen -Dwdir1 -stpages.sty -g -wgturl -otfile.html tdata.csv
```

## 4-5. Expage 製品

### 4-5-1. 各種オプション説明

オプション	説明
D	: 作業ディレクトリの指定 帳票資源ファイルの場所（ディレクトリ）を指定します。
o	: 出力ファイルの指定 出力をファイルに書き出す場合に使用します。
s	: ジョブファイル（.sty）の指定 指定しない場合は標準ジョブファイル（default.sty）の指定と同じです。
ix	: XML データマップ出力の指定 XML データマップを用いた出力実行の際には必須の設定です。
Lid	: ジョブ ID の指定
Lfn	: ログファイル名の指定

### 4-5-2. Windows 版のみ

オプション	説明
z	: テキストファイルの強制削除指定 Excel 出力終了後にテキストデータを強制的に削除します。

### 4-5-3. 実行オプション使用例

オプションを指定したコマンドライン例を示します。

- 作業ディレクトリ `wdir1` のジョブ `tform.sty` でデータ `tdata.xml` の Excel ファイル `tdata.xlsx` を生成します。

```
cexpage -Dwdir1 -stform.sty -ix -otdata.xlsx tdata.xml
```

# **Create!Form 11**

**API 第 5 版**

---

発行日	2019 年 4 月
発行者	インフォテック株式会社 〒 160-0023 東京都新宿区西新宿 7-5-25