Windows ターミナルの使用法

Version 2024.1
2024-06-03

InterSystems Corporation　1 Memorial Drive　Cambridge MA 02142　www.intersystems.com

Windows ターミナルの使用法
InterSystems IRIS Data Platform　Version 2024.1　2024-06-03

インターシステムズ社の製品に関するサポートやご質問は、以下にお問い合わせください：

InterSystems Worldwide Response Center (WRC)
Tel:　+1-617-621-0700
Tel:　+44 (0) 844 854 2917
Email:　support@InterSystems.com

目次

1

ターミナル・アプリケーションの基本情報

ターミナル・アプリケーション (Windows でのみ利用可能) は、ウィンドウに ObjectScript シェルを表示し、追加オプションのメニュー・バーも提供します。ターミナル・アプリケーションの主な目的は、ObjectScript コマンドの実行 (およびその結果の確認) とその他のさまざまなシェルへのアクセスを可能にすることです。ターミナル・アプリケーションでは、このアプリケーションに固有の構文を使用するスクリプトを実行することもできます。このページでは、基本情報を提供します。

注釈　このドキュメントでは、ターミナルという用語を使用する場合、ほとんどは単に ObjectScript シェル (特別なメニュー・バーは存在せず、すべてのオペレーティング・システムで使用できます) を意味します。同様に、ターミナル・セッションというフレーズは多くの場合、ObjectScript シェル内のセッションを意味します。

これらの概念の混乱を避けるために必要に応じて、このドキュメントでは Windows 専用アプリケーションを指す場合、ターミナル・アプリケーションというフレーズを使用します。

1.1 ターミナル・アプリケーションの起動

ターミナル・アプリケーションを起動するには、次のいずれかを実行します。

- ローカル・データベースに接続するには、[InterSystems ランチャー] を選択し、次に **[ターミナル]** を選択します。
- リモート・サーバに接続するには、[InterSystems ランチャー] を選択し、次に **[リモート・システム・アクセス]→[ターミナル]** を選択します。次にサーバ名を選択します。

いずれの場合も、関連する InterSystems IRIS® データ・プラットフォーム・サーバのセキュリティ設定に基づいて、ログインするよう求められるか、既定のユーザ名でログインされます。これにより、ターミナル・アプリケーションでは (既定では)、プロンプトに現在のネームスペースの名前が表示されます。例えば、USER ネームスペースにいる場合、既定のプロンプトは以下のようになります。

```
USER>
```

別のネームスペースに切り替えるには、以下の例のように、$namespace 変数を使用します。

Terminal

```
USER>set $namespace="SAMPLES"
SAMPLES>
```

これは、コード内でネームスペースを切り替えるのと同じ方法です。詳細は、"ObjectScript リファレンス" の "$namespace" のリファレンス・ページを参照してください。

1.2 ターミナル・アプリケーションの機能の概要

ターミナル・アプリケーションは、ObjectScript シェルを提供するウィンドウ、タイトル・バー、およびメニュー・バーで構成されます。

まず、ObjectScript シェルで提供されるすべてのオプションを使用できます。これは、ObjectScript コマンドの実行（およびその結果の確認）だけでなく、その他のさまざまなシェルへのアクセスに使用できます。また、ObjectScript シェルには、広範な行呼び出し機能および複数のカスタマイズ・オプションが用意されています。

このシェルが用意されているほかに、ターミナル・アプリケーションのメニュー・バーには、テキストのコピーと貼り付け、画面のクリア、ログ記録、印刷、スクリプトの実行、カスタマイズを行うためのオプションもあります。ターミナル・アプリケーションには、コピーと貼り付けオプションを含む右クリック・メニューもあります。

タイトル・バーには、ターミナル・アプリケーションが接続されている InterSystems IRIS® データ・プラットフォーム・サーバが示されます。これは、それぞれ異なるサーバに対する複数のセッションがある場合に特に便利です。タイトル・バーには現在使用されている通信モードも示されます。

- タイトル・バーには **InterSystems IRIS TRM**:pid(instancename) と表示される場合があります。
  - ここで、pid は、ターミナル・アプリケーションが通信している InterSystems IRIS プロセスのプロセス ID です。
  - instancename は、プロセスが実行されている InterSystems IRIS インスタンスです。

  この場合、ターミナル・アプリケーションは、共にインストールされている InterSystems IRIS サーバとの間で独自のローカル通信を使用しています。
- タイトル・バーには (server **NT — InterSystems IRIS Telnet**) と表示される場合があります。ここで、server はリモート・サーバのホスト名です。この場合、ターミナル・アプリケーションは、TCP/IP を介した TELNET プロトコルを使用して、Windows InterSystems IRIS サーバまたは UNIX® ホストと通信します。

Windows では、InterSystems IRIS の通信スタックは Winsock です。この通信モードから報告されるエラーには Winsock エラー・コードの名前が使用されます。例えば、**WSAECONNREFUSED** は、接続が拒否されたことを意味します。

1.3 コピーと貼り付け

ターミナル・アプリケーションでテキストをコピーして貼り付けるには、右クリック・メニュー、**[編集]** メニュー、または各種キーボード・ショートカットを使用できます。メニューでは、以下のオプションを使用できます。

- **コピー** — 選択されているテキストを、クリップボードにコピーします。
- **貼り付け** — クリップボードのコンテンツをカーソルの現在位置（スクロールバック・バッファの最後）に行ごとに貼り付けます。このテキストは、エコーが無効にされていない限り、ウィンドウに表示されます。
- **コピー + 貼り付け** — 選択したテキストをクリップボードにコピーして、カーソルの現在位置に行ごとに貼り付けます。

1.3.1 キーボードによるショートカット

使用できるキーボードによるショートカットは、以下のとおりです。

アクション	基本のショートカット	Windows のショートカット
コピー	Ctrl-Insert	Ctrl-C

アクション	基本のショートカット	Windows のショートカット
貼り付け	Shift-Insert	Ctrl-V
コピーと貼り付け		Ctrl-Shift-V

基本のショートカット列に示したショートカットは、常に有効です。

Windows のショートカット列に示したショートカットは、[Windowsエディットアクセラレータ] オプションを [はい] に設定した場合にのみ有効になります。この設定の詳細は "ユーザ設定" を参照してください。

1.3.2 コピーと貼り付けに関する注記

- 前述のように、[Windowsエディットアクセラレータ] オプションを [はい] に設定すると、Ctrl-C によって、選択したテキストが Windows クリップボードにコピーされます。実行を中断するには、代わりに Ctrl-Shift-C を押す必要があります。
- ホストに処理中のマウス・リクエストがあるときに、切り取りおよび貼り付けをローカルで実行したい場合は、対象領域を選択しながら Ctrl キーを押すと、マウスのアクションがホストに報告されなくなります。
- コピーされたテキストに行の境界が含まれる場合は、キャリッジ・リターンと改行としてクリップボードに保存されます。改行を貼り付けない場合は、"ユーザ設定" を参照してください。
- ターミナル・アプリケーションのデータ貼り付け速度が、ホストのデータ受信速度よりも速い場合がしばしばあります。貼り付け速度を制御するための設定は、"ユーザ設定" を参照してください。また、貼り付けコマンドの実行時に、改行を破棄することもできます。

1.4 実行の中断

フォアグラウンドのすべての実行内容を中断するには、次のいずれかのキー組み合わせを使用します。

- Ctrl-C ― [Windowsエディットアクセラレータ] オプションが有効になっていない場合に使用します。
- Ctrl-Shift-C ― [Windowsエディットアクセラレータ] オプションが有効になっている場合に使用します。

[Windowsエディットアクセラレータ] オプションの詳細は "ユーザ設定" を参照してください。

1.5 画面のクリア

[編集] メニューには、画面をクリアするための 2 つの異なるオプションが用意されています。

- 画面をリセットするには、[編集]→[リセット] を選択します。このオプションを使用すると、現在のページ上のマージン、スクロール領域、およびその他の処理がリセットされ、ウィンドウが再描画されます。
- 画面を再初期化するには、[編集]→[削除] を選択するか、Ctrl-Del を押します。このオプションは、ウィンドウを再初期化し、すべてのセッション・データを消去し、スクロールバック領域を 0 にリセットします。

1.6 ターミナル・アプリケーション・セッションのログへの記録

現在のセッションのログ記録を開始するには、以下の手順を実行します。

1. **[ファイル]→[ログ]** を選択するか、Alt-L を押します。

   ターミナル・アプリケーションは、ログ・ファイルの場所と名前の入力を促すダイアログ・ボックスを表示します。既定のディレクトリは、install-dir/**mgr** です。既定のファイル名は **TERMINAL.LOG** です。

   パスとファイル名の合計長は 126 文字を超える長さにはできません。

2. 必要な場合は、別のディレクトリとファイル名を指定します。

3. [OK] をクリックします。

   ログ・ファイルが存在する場合は、上書きするかどうか尋ねるダイアログ・ボックスが表示されます。以下のいずれかを選択します。

   - **[はい]** は、新しいログ・データでファイルを上書きします。
   - **[いいえ]** は、新しいログ・データをファイルに追加します。
   - **[キャンセル]** は、ファイルをそのまま残します (ロギングは行われません)。

その後、ログへの記録を終了するには、**[ファイル]→[ログ]** を選択するか、Alt-L を押します。ログ・ファイルが閉じられたことを示すダイアログ・ボックスがターミナル・アプリケーションに表示されます。[OK] を選択します。

ログ・ファイルには、接続からの出力のみが記録されます (現行のラップ・モードには関係ありません)。

スクリプトからログ記録を実行することもできます。**[ファイル]→[ログ]** によりロギングを開始した場合、ロギングも実行するスクリプトは開始できないことに注意してください。そのようにした場合の動作は不確定です。

"学習モード" も参照してください。

1.7 印刷

印刷するには、**[ファイル]** メニューの以下のオプションを使用します。

- プリンタを選択して、ターミナル・アプリケーションでそのプリンタを使用するように設定するには、**[ファイル]→[印刷設定]** を選択します。
- 画面の内容を印刷するには、**[ファイル]→[印刷]** を選択します。
- ログ・ファイル (または他の ASCII ファイル) を印刷するには、**[ファイル]→[ログの印刷]** を選択します。このオプションでは印刷するファイルを選択でき、改ページ文字を適切に処理する以外は、特別な操作は必要ありません。印刷中は、マウスおよびキーボード入力がメイン・ウィンドウからロック・アウトされ、キャンセル用のダイアログ・ボックスが表示されます。印刷はドラフト・モードで実行されます。

1.8 終了

ターミナル・アプリケーションを終了するには、以下のいずれかの操作を実行します。

- **[ファイル]→[終了]** を選択します

- Alt-F4 を押します
- HALT または H と入力します (大文字/小文字の区別なし)

これらのオプションにより、ターミナル・アプリケーションの現行のコピーが終了し、開いているすべてのファイルが閉じられて、フォアグラウンドのすべての実行内容が停止されます。

このターミナル・アプリケーションが起動時にサーバに接続されていた場合は、通信チャネルが閉じたときに自動的に終了します。

InterSystems ランチャーの [InterSystems Telnet] を使用してこのターミナル・アプリケーションにアクセスした場合、ターミナルは通信チャネルが閉じたときに自動的に終了せず、アクティブ状態のままになるため、[**接続**] メニューを使用して再接続できます。

1.9 技術メモ

プロセスは、Windows にログインしてターミナル・アプリケーション (iristerm.exe) を実行しているユーザによって所有されます。

また、すべての環境変数および共有ドライブ文字の指定は、アプリケーションを実行しているユーザによって定義されます。

1.10 関連項目

- リモート・ホストへの接続
- ターミナル・スクリプトの使用法
- スクリプト・コマンドのリファレンス
- ターミナル・アプリケーションのカスタマイズ
- コマンド行からのターミナル・アプリケーションの実行
- ターミナル使用に関するその他のトピック

2
リモート・ホストへの接続

リモート・ホストのデータベースに現在のターミナル・セッションを接続するには、[接続] メニューを使用します。

注釈　ターミナルを起動する際に /console または /server という制御引数を指定した場合は、[接続] メニューは表示されません。"接続オプション" を参照してください。

2.1 [接続] メニューのオプション

[接続] メニューには以下のオプションがあります。

- リモート・ホストに接続するには、[接続]→[ホスト] を選択するか、Alt-O を押します。これにより、目的のホストのアドレスを入力するためのダイアログ・ボックスが表示されます。

  または、[接続] メニューからホスト名を選択します。
- 通信チャネル経由でブレークを送信するには、[接続]→[ブレーク信号を送る] を選択するか、Alt-B を押します。
- 切断する、または接続の試行をキャンセルするには、[接続]→[切断] をクリックします。

2.2 リモート接続の例

この例では、ターミナルのインスタンスを起動して、それをローカル・ホスト上の TELNET ポートに手動で接続し、コンソール・セッションを有効化します。この例は、既定のユーザ ID とパスワードを使用できることを前提にしています。

1. [接続]→[ホスト] を選択します。
2. 表示されるダイアログ・ボックスで、[リモート・システム: アドレス] に 127.0.0.1 を、[ポート番号] に 23 を入力します。[OK] をクリックします。

   ターミナル・アプリケーションが TELNET ポートを介してローカル・ホストに接続しようとします。
3. Username: プロンプトで、SYS と入力し、Enter を押します。
4. 次に、Password: プロンプトで、パスワードを入力し、Enter を押します。

   %SYS プロンプトが表示されます。これは、接続が完了して、現在は %SYS ネームスペースにいることを意味します。

セッションを終了するには、**[接続]→[切断]** を選択します。また、このターミナルを終了するには、ウィンドウの右上にある [閉じる] ボックスを選択します。

3

ターミナル・スクリプトの作成と使用

ここでは、ターミナル・アプリケーションでスクリプト・ファイルを作成して使用する方法について説明します。

注釈　ターミナル・スクリプトが役立つ場合もありますが、通常はルーチンを作成して使用する方が（各ルーチン・プログラミング言語が非常に豊富なオプション・セットを提供しているので）はるかに簡単です。

ユーザ環境変数および共有ドライブ文字の指定は、InterSystems IRIS® データ・プラットフォーム・コントロール・サービスを実行するユーザ・アカウントによって定義されます。

3.1 スクリプト・ファイルのコンテンツ

スクリプトは 1 行指向で、行継続の表記規則がありません。各行は他の行から完全に分離されています。セミコロンで始まる行はコメントと見なされます。読みやすさを向上させるために空白行を自由に使用できます。通常、無効な行は無視されます。スクリプト・コマンドは、スペースおよび/またはタブの後に続けます。

スクリプト・コマンドを含む行の形式は以下のとおりです。引数は文字列または数値と解釈されます。

```
ScriptCommand: ScriptArguments
```

ここで ScriptCommand は、ターミナル・スクリプト・コマンドの 1 つで、ScriptArguments は、そのコマンドの引数リストです（具体的なコマンドの詳細を参照）。スクリプト・コマンドが 2 つ以上の単語で構成されている場合は、コマンドの各語間を 1 つのスペースで区切る必要があります。また、コマンドとコロンの間にはスペースを入れません。

引数のないコマンドを以下に示します。

```
ScriptCommand
```

制御の移動箇所の定義には、ラベルを使用できます。ラベルはドル記号（$）で始まり、大文字と小文字が区別されません。ラベルには、スペースを埋め込むことができます。ラベルは、1 行に単独で記述する必要があります。

"学習モード" も参照してください。

3.2 スクリプト・コマンドの概要

次の表は、使用可能なスクリプト・コマンドの一覧です。詳細を示すリファレンス・コンテンツへのリンクが付いています。

コマンド	アクション
break	ブレークをサポートする通信デバイスにブレークを送信します。
call script	現在のスクリプトを終了して、他のスクリプトを開始します。
case match	"wait for" コマンドの文字列が大文字/小文字まで一致する必要があるかどうかを指定します。
closelog	ログ・ファイルを閉じます。
connect	ホストに未接続の場合に、ホスト接続を強制します。
debug	スクリプトのデバッグを有効化または無効化します。
disconnect	ホストに接続されている場合に、接続の切断を強制します。
display	ディスプレイにテキストを送信します。
echo	入力文字のエコーのオン/オフを切り替えます。
execute	Windows プログラムを実行します。
exit	スクリプトを終了します。
goto	制御をスクリプト内の別の場所に移します。
if empty	最後のテスト文字列が空の場合に、制御を移動します。
key_starttime	キーの時間計測の開始をシミュレートします。
key_stoptime	キーの時間計測の停止をシミュレートします。
key_timer	キーの時間計測のオン/オフを切り替えます。
logfile	ログ・ファイルを開始します。
multiwait for	通信デバイスからの任意の複数文字列を待機します。
notify	ダイアログ・ボックスを表示して、ユーザからの応答を待ちます。
on error	タイマが起動された場合の分岐先ラベルを指定します。
pause	スクリプトを一時停止します。
return	スクリプト・ファイルのサブルーチンから戻ります。
send	通信デバイスにテキストを送信します。
subroutine	スクリプト・ファイル内のサブルーチンを呼び出します。
terminate	エミュレータを完全に終了します。
test	テスト対象の文字列を構築します。
timer	"wait for" コマンド用のタイマを制御します。
title	ウィンドウ・タイトルを設定します。
wait for	通信デバイスからの特定の文字列を待機します。

これらのコマンドのリファレンス情報は、"スクリプト・コマンドのリファレンス" を参照してください。

3.3 スクリプト・コマンドの引数

引数の前後のスペースおよびタブは、すべて無視されます。

数値引数は、すべて整数値です。必須の数値引数が指定されていない場合は、既定で 0 になります。また、OFF は 0 と同等で、ON は 1 と同等です。

文字列は、コマンドの後に続く行上の個々のデータを連結したものにすぎません (先頭と末尾の空白は除く)。引用符は必要ありません。また、次のコマンド行を使用することで、パラメータ置換が実行されます。

<P1>, <P2>, ..., <Pn>

これで、<Pn> が、n 番目のコマンド行パラメータに置換されます。

操作を簡略化するために、特定の ASCII 文字には、この後のテーブルに示す同等のショートカット表記が用意されています。

注釈　NUL (000) 以外のすべての ASCII (拡張) 文字は、<ddd> によって生成できます。ddd はその文字を表す 10 進数値です。

文字	解釈	送信シーケンス
<CR>	キャリッジ・リターン	<13>
<F10>	F10 キー	<27>[21~
<F7>	F7 キー	<27>[18~
<DO>	Do キー	<27>[29~
<TAB>	Tab キー	<9>
<LF>	改行	<10>
<ESC>	Esc キー	<27>
<DCS>	デバイス制御文字列の接頭部	<144>
<ST>	デバイス制御文字列の終了	<156>
<EMU>	拡張エミュレータ・コマンドの開始	<144>i
<NL>	新規行	<CR><LF>
<CSI>	制御文字列の接頭部	<155>

3.4 スクリプト例

スクリプトの例を以下に示します。

```
; initialization -- turn match off to make comparisons more lenient
case match: off

; wait for the terminal to initialize and ask for our identification
echo: off
wait for:Username
send: SYS<CR>
wait for:Password
send: XXX<CR>
title: Terminal Example
echo: on
```

```
; log everything in a log
logfile: C:\TermExample.log
; wait a second
pause:10

; display a header to let the user know we are ready
; you need <CR><LF> because "display" does not
; have a prompt to advance to another line
display:<CR><LF>
display:------------------------------<CR><LF>
display:<<< Terminal Example >>><CR><LF>
display:------------------------------<CR><LF>
; wait a second
pause:10

; switch to the USER namespace
send: set $namespace="USER"<CR>
wait for:USER>

; display some basic information about the system
; Use the debugging routine to do so
send: Do ^%STACK<CR>
wait for: action:

; have it outline our options
send: ?<CR>
wait for: action:

; wait 5 seconds for user to absorb
pause: 50

; ask for the basic process info
send: *s
pause: 50
send: <CR>
wait for: action:

; wait another 10 seconds
pause: 100

; finish the session
send: <CR>

; close the log file
closelog

; finished
terminate
```

3.5 スクリプトの起動

通常、スクリプト・ファイル (既定の拡張子は .scr) は作業ディレクトリにありますが、任意の場所に配置可能です。

スクリプトを実行するには、**[ファイル]→[スクリプト]** を選択するか、Alt-S を押します。Windows の標準的なファイル検索ボックスが表示され、そこでスクリプトを選択します。

コマンド行の引数としてスクリプトが指定されている場合は、コマンド・モードをロックしているスイッチがなければスクリプトが直ちに開始され、スイッチがあればホスト接続が確立されるまで開始が延期されます。

注釈　通信オプションをシングル・モードに編集することは、ターミナル・アプリケーションをシングル・オプションにロックすることと同じです。したがって、スクリプト・ファイルの起動はホスト接続の確立後まで延期されます。

3.6 スクリプトの一時停止

スクリプトの実行を停止するには、**[ファイル]→[一時停止]** を選択するか、Alt-P を押します。現在のスクリプトを一時停止することを確認するプロンプトが表示されます。

3.7 スクリプトの停止

スクリプトを停止するには、[ファイル]→[スクリプト] を選択するか、Alt-S を押します。現在のスクリプトを停止することを確認するプロンプトが表示されます。

3.8 関連項目

- スクリプト・コマンドのリファレンス
- 学習モード
- コマンド行からのターミナル・アプリケーションの実行

4

スクリプト・コマンドのリファレンス

ここでは、ターミナル・スクリプトの作成時に、ターミナル・アプリケーションで使用可能なスクリプト・コマンドのリファレンス情報を提供します。

注釈　スクリプトが役立つ場合もありますが、通常はルーチンを作成して使用する方が（各ルーチン・プログラミング言語が非常に豊富なオプション・セットを提供しているので）はるかに簡単です。

4.1 break

ブレークをサポートする通信ノードにブレークを送信します。それ以外には何も実行しません。引数は取りません。使用例は以下のとおりです。

```
break
```

4.2 call script

スクリプトの実行を開始します。このコマンドを実行したときにスクリプトが実行中の場合は、ターミナル・アプリケーションで実行中のスクリプトを終了してから、新しいスクリプトが開始されます。使用例は以下のとおりです。

```
call script: login fred <p3>
```

この例は、現行のスクリプトを停止し（スクリプトが実行中の場合）、login.scr という名前の別のスクリプトを開始します。サンプル・スクリプト（login.scr）の最初のパラメータは `fred` です。第 2 パラメータは現行のスクリプト・ファイル（呼び出しを行っているスクリプト・ファイル）の第 3 パラメータと同じになります。既定のファイル拡張子は .scr と見なされます。最初に、現行の作業ディレクトリで login.scr のインスタンスが検索されます。

4.3 case match

wait for コマンドで大文字/小文字の照合を有効化または無効化します。使用例は以下のとおりです。

```
case match: off
```

大文字/小文字の照合を無効化すると、個々の文字の大文字/小文字が異なっていても、文字列が一致していると見なされます。このスイッチの既定は on です。

4.4 closelog

現在開いているログ・ファイルがあれば、それを閉じます。使用例は以下のとおりです。

```
logfile: mydirect.log
send: dir *.*/FULL<CR>
wait for: <NL>$
closelog
```

4.5 connect

リモート・ホストへの接続を開始するためのダイアログ・ボックスを開きます。使用例は以下のとおりです。

```
connect
```

4.6 debug

デバッグ・モードを有効にします。デバッグ・モードでは、通常はターミナル・アプリケーションが無視する無効なスクリプト・コマンドをトラップします。デバッグ・モードを有効にすると、オペレータの注意を喚起するメッセージ・ボックスに、無効なコマンドの最初の部分が表示されます。使用例は以下のとおりです。

```
debug: on
```

4.7 disconnect

ホストとの接続を切断します。ターミナル・アプリケーションが未接続の場合は、何の動作もしません。使用例は以下のとおりです。

```
disconnect
```

4.8 display

データを画面に出力します。通信デバイスに、データは送信されません。使用例は以下のとおりです。

```
display: <CSI>H<CSI>J<LF>Here are the choices for today:
```

この例を実行すると、カーソルがホーム位置に戻され、ウィンドウがクリアされます。次に、1 行進めて `Here are the choices for today:` というテキストが出力され、テキストの末尾にカーソルが置かれます。

4.9 echo

ウィンドウおよびログ・ファイルへの出力の表示を有効化または無効化します。これは、出力を非表示にする必要がある場合 (ユーザにとって情報価値がないためなど) に役立ちます。例は、″wait for″ を参照してください。

4.10 execute

Windows プログラムを起動し、そのウィンドウに SHOW 属性を設定します。使用例は以下のとおりです。

```
    execute: notepad.exe myfile.not
```

この例では、Windows のメモ帳プログラムを起動し、そのアプリケーション内でファイル myfile.not を開きます。次のような使用方法にも注目してください。

```
    logfile: mydat.lst
    echo: off
    send: dir *.dat/full
    wait for: <NL>$
    closelog
    echo: on
    execute: notepad mydat.lst
```

注釈　　プログラムが実際に開始されたかどうかを確認するテストは実行されず、その完了も待機されません。

4.11 exit

スクリプトを終了します。通常、スクリプトは最後の行まで実行されると終了しますが、特定のイベント (ログインなど) が生じないときに終了させたい場合があります。使用例は以下のとおりです。

```
    on error: $byebye
    timer: 40
    wait for: event:
    goto: $Got event

$byebye:
    notify: Did not find event prompt, exiting script
    exit

$Got event:
    timer: 0
    ; more commands
```

4.12 goto

制御をスクリプト・ファイル内の別の場所に移します。このコマンドは、ループ処理のコントロール・フローの管理や、タイムアウト分岐への対応の使用に便利です。使用例は以下のとおりです。

```
    on error: $Not There
    timer: 30
    wait for: abc<CR>
    goto: $Got It

$Not There:
    ;failed to see it, send Ctrl+C
    send: <3>
    goto: $bad

$Got It:
    ;turn timer off because we got abc<CR>
    timer: 0

    ;more commands ...
```

4.13 if empty

最後の test コマンドが空文字列を検出した場合に、指定したラベルに分岐させることができます。使用例は以下のとおりです。

```
    test: <p1>
    if empty: $No First Arg
```

最初のコマンドは、コマンド行で指定された第 1 パラメータが見つかるかどうか、つまり空でないかどうかを検証します。2 番目のコマンドは、それが見つからない場合に、ラベル `$No First Arg` に分岐させます。

4.14 key_starttime

キー・シーケンスの時間計測を開始します。このコマンドは、1 つの数値引数を取ります。引数が 0 の場合は、Enter を押したときに、統計が蓄積されます。それ以外の場合は、F10 を押したときに、統計が蓄積されます。使用例は以下のとおりです。

```
    key_starttime: 0
```

時間計測を停止するには、key_stoptime コマンドを使用します。

4.15 key_stoptime

時間計測が現在アクティブな場合に、時間計測を停止し統計を蓄積します。使用例は以下のとおりです。

```
    key_starttime: 0
    wait for: <esc>[14;22H
    key_stoptime
```

4.16 key_timer

キーの時間計測情報のデータ収集を開始または停止します。または、Alt-Shift-T を使用して、タイマを開始または停止できます。使用例は以下のとおりです。

```
key_timer: on
; rest of your script commands
key_timer: off
```

ファイル (KEYTIMER.LOG) がシステム・マネージャ・ディレクトリに作成され、キーの時間計測のヒストグラムが記録されます。時間計測シーケンスは現行の統計ファイルに追加されるのではなく上書きされるため、使用可能な時間計測シーケンスは 1 つのみです。

注釈　　時間計測をスクリプト・ファイルから排他的に起動するには、<13>、<27>[21- の代わりに、それぞれ <CR>、<F10> を使用する必要があります。

4.17 logfile

指定されたログ・ファイルで受信データの収集を開始します。アクティブなログ・ファイルがある場合は、適切に終了されます。ロギングの停止には、closelog コマンドを使用します。使用例は以下のとおりです。

```
logfile: mydirect.log
send: dir *.*/FULL<CR>
wait for: <NL>$
closelog
```

既定のディレクトリはスクリプトの存在するディレクトリとなります。これはフル・パス名を与えることで変更できます。

通常、ログ・ファイルは上書き方式で開きます。つまり、ログ・ファイルが既に存在する場合は、新規データで既存のものが置換されます。

4.18 multiwait for

スクリプト・ファイルとホストを同期化します。ホストから受信したデータが、引数に指定されたいくつかの文字列の 1 つに一致するまで、処理が中断されます。使用例は以下のとおりです。

```
multiwait for:  =USER>=***ERROR,=DONE
```

この例では、指定された 3 つの文字列の中の 1 つが到着するまで、スクリプト・ファイルが待機することになります (永久に待機する可能性もあります)。引数の最初の非空白文字 (この例では等号記号) は、引数を複数の部分文字列に分割する区切り文字として機能します。したがって、このコマンドは、次のいずれかのシーケンスが到着するまでサンプル・スクリプトを待機させます。

```
USER>
***ERROR,
DONE
```

タイマを使用することで、このコマンドを終了できます。

大文字/小文字の完全一致を有効または無効にする場合は、case match コマンドを参照してください。

case match コマンドの引数には 1 つの部分文字列しか指定できないため、次の 2 つのスクリプト・コマンドの機能は同じになります。

```
multiwait for:  =USER>
wait for: USER>
```

4.19 notify

Windows メッセージ・ボックスを表示し、ユーザが [OK] ボタンを押すまで待機します。スクリプトを実行するユーザへのメッセージに使用できます。使用例は以下のとおりです。

```
notify: Ready to send commands...
send: copy *.lst backup:*.lst<CR>
send: delete *.lst;*
```

注釈　　このメッセージ・ボックスはモーダルであるため、他のユーザが割り込むことはできません。

4.20 on error

タイマが時間切れになった場合 (通常はテキストの到着を待っている最中) に実行される暗黙の goto のターゲット・ラベルを設定します。厳密な正確さを確保するには、このコマンドは timer の使用前に使用する必要がありますが、実際にはこの順序は重要ではないことがあります。

例は、"wait for"、"exit"、"goto"、および "subroutine" を参照してください。

また、"timer" コマンドも参照してください。

4.21 pause

実行中のスクリプトを一時停止します。停止時間は 10 分の 1 秒単位で指定します。使用例は以下のとおりです。

```
pause: 30
```

このコマンド例では、スクリプトの実行が 3 秒間停止されます。引数が 0 の場合は永久停止となります。永久停止から再開するには、Alt-P を使用します。

4.22 return

subroutine コマンドと共に使用して、スクリプト内でそのサブルーチンを呼び出した箇所に戻ります。使用例は、subroutine コマンドを参照してください。

4.23 send

現在接続中のホストに送信する入力データをシミュレートします。使用例は以下のとおりです。

```
send: set $namespace="%SYS"<CR>
```

この行はネームスペースを %SYS に変更します。send コマンドはキャリッジ・リターンを暗黙的に追加しないため、末尾の <CR> が必要です。

もう 1 つの使用例は以下のとおりです。

```
send: 1<cr>2<cr>A1234<F10><32>
```

このコマンドは、以下の順で入力したことと同等です。

- 文字の 1
- キャリッジ・リターン・キー
- 文字の 2
- キャリッジ・リターン・キー
- A1234 という文字列
- F10 キー
- 1 つの空白文字

<32> は先頭または末尾のスペースを送信する唯一の方法であることに注意してください。普通に入力した場合、これらの文字はコマンド・インタープリタによって削除されます。

重要

wait for コマンドを各 send コマンドの後ろに組み込むことをお勧めします。wait for コマンドにより、スクリプト内のコマンドをターミナル・アプリケーションからの入力と後で同期させることができます。ターミナル・スクリプト・メカニズムは、wait for コマンドを検出した場合を除き、InterSystems IRIS® データ・プラットフォームから返される入力に関係なく、コマンドを順次送信します。

wait for コマンドを各 send コマンドの後ろに組み込まず、ログ・ファイルを生成する場合、そのログ・ファイルには後続の send コマンドの情報は含まれません。

4.24 subroutine

スクリプト内で同じコマンドを何度も使用する場合に役立ちます。このコマンドを使用すると、記憶域を節約できると同時に、異なる多くのラベルを保持する必要がなくなります。このコマンドは return コマンドと共に使用します。使用例は以下のとおりです。

```
    subroutine: $Send It Again
    ; some other processing
    exit

$Send It Again:
    send: <F7>Q
    on error: $skip
    timer: 30
    wait for: [22;5H
    timer: 0
    return

$skip:
    send: <3>
    ; note on error still set to $skip
    timer: 30
    wait for: function:
    timer: 0
    send: <CR>
    exit
```

注釈　サブルーチン・スタックは 16 アドレスを保持します。それより深くサブルーチンをネストしようとすると、スクリプトは失敗します。

4.25 terminate

終了して Windows に戻るようターミナル・アプリケーションに命令します。開いているすべてのファイルが閉じ、不要なウィンドウが消去され、接続が閉じます。使用例は以下のとおりです。

```
    terminate
```

4.26 test

パラメータまたはウィンドウ・プロパティが空でないかどうかを調べます。このコマンドは、if empty コマンドと連携して使用します。使用例は、if empty コマンドを参照してください。

4.27 timer

wait for コマンドと連携して使用されるタイマを設定します。timer コマンドは、Windows の SetTimer() コマンドを実行します。このタイマが始動すると、スクリプト・プロセッサは on error で指定されたラベルに移動します (指定されている場合)。このスクリプトは、on error によってラベルが指定されていない場合には直ちに終了します。

使用例は以下のとおりです。

```
    timer: 100
```

引数の数値は待機時間で、10 分の 1 秒単位で指定します。このコマンド例では、タイマに 10 秒が設定されています。goto コマンドの使用例も参照してください。

タイマをオフにするには、次のコマンドを使用します。

```
    timer: 0
```

timer の使用例は、"wait for" を参照してください。

4.28 title

ターミナル・アプリケーション・ウィンドウのタイトルを指定された文字列に設定します。使用例は以下のとおりです。

```
    title: This is my window
```

タイトルの設定は、拡張エミュレータ・コマンドを使用してリモートで実行することもできます。

4.29 wait for

スクリプト・ファイルとホストから受信したデータを同期化します。使用例は以下のとおりです。

```
    wait for:  USER>
```

この例では、USER> と完全に一致する文字列が到着するまで、スクリプト・ファイルが待機することになります (永久に待機する可能性もあります)。この特別なシーケンスは、USER ネームスペースにあるときのターミナル・アプリケーションからの既定のプロンプトです。つまり、このコマンドは、ターミナル・アプリケーションが次の入力を受け入れ可能になるまで待機する場合に使用できます。

timer を使用することで wait for コマンドを終了できます。対象のテキストが到着しない場合、またはそのテキストが時間計測や大文字/小文字の不一致が原因で見つからない場合は、timer が wait for に割り込みスクリプトの実行を継続する唯一の方法です。timer を使用する際は、そのタイマが時間切れになった場合にフローを受け取るラベルを指定できます ("on error" を参照してください)。wait for で検索対象のテキストが見つかった場合は、タイマが強制終了されて、on error によって設定されたラベルがクリアされます。使用例は以下のとおりです。

```
    echo: off
    on error: $Failed Login
    timer: 50
    wait for: Name:
    send: <p1><CR>
    wait for: Password:
    send: <p2><CR>
    wait for: <NL>$
    echo: on
    Notify: Login is complete
    display: <CSI>H<CSI>J
    send: <CR>
    goto $Process

$Failed Login:
    echo: on
    notify: Login failed.
    exit

$Process:
    ;processing begins
```

このコマンド例は、最初の 2 つのスクリプト・パラメータで指定された名前とパスワードを使用するログイン・シーケンスを非表示にします。ログインに成功すると、指定されたラベルの箇所で処理が開始されます。失敗の場合は、ログインに失敗したことを示すダイアログ・ボックスが表示され、[OK] を選択すると終了します。

wait for コマンドでテキストの大文字と小文字が区別されるかどうかは、case match コマンドの使用状況によって異なります。

5

ターミナル・アプリケーションのカスタマイズ

ここでは、ターミナル・アプリケーションの外観と動作をカスタマイズするさまざまな方法について説明します。

5.1 フォントの指定

フォント・サイズを指定するには、**[編集]→[フォント]** を選択します。これによりダイアログ・ボックスが表示されます。このダイアログ・ボックスで、使用しているモニタと解像度に適した書体、サイズ、およびスタイルを選択できます。

注釈　画面の枠を越えてウィンドウが拡張するようなフォント・サイズを選択した場合は、画面とフォントの両方が使用可能な最大サイズに自動調整されます。

また、異なるサイズ画面に切り替えたときは常に、そのサイズに対して事前選択されているフォントが使用されます。

5.2 色の指定

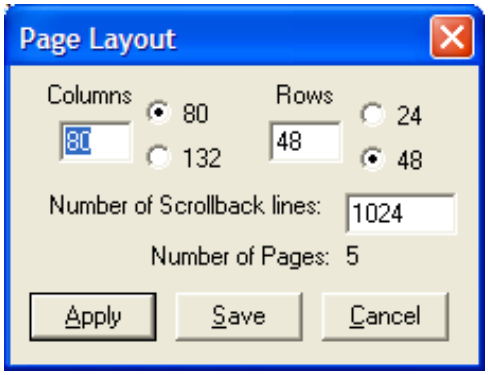
色を指定するには、**[編集]→[色]** を選択します。これにより、ターミナル・アプリケーションの既定の前景色と背景色を選択できるダイアログ・ボックスが表示されます。以下のいずれかを選択します。

- **[適用]** をクリックすると、現在のセッションのみが変更されます。
- **[保存]** をクリックすると、現在のインスタンスは変更されず、新規セッション用に色情報が保存されます。

ANSI 名で事前指定された色から、ディスプレイ・ボードが提供する任意の色に調整できます。これらの色は、前景色および背景色と共に保存されます。既定の色を選択するには、**[デフォルト]** を選択してから色を選択します。

5.3 ウィンドウ・サイズの指定

ターミナル・アプリケーション・ウィンドウのサイズを指定するには、**[編集]→[ウィンドウサイズ]** を選択します。次のようなダイアログ・ボックスが表示されます。



列の最大数は 132 で、行の最大数は 64 です。変更を加えると、ダイアログ・ボックスで、使用可能なスクロールバックの行数とページ数が更新されます。以下のいずれかを選択します。

- [適用] をクリックすると、現在のセッションのみが変更されます。
- [保存] をクリックすると、現在のインスタンスは変更されず、新規セッション用にウィンドウ・サイズ情報が保存されます。

注釈　ウィンドウのサイズを変更すると、ターミナル・アプリケーションでは、現在の表示ページとすべてのバック・ページにある現行のすべてのデータが消去されます。さらには、新しいサイズ用に選択されているフォントがある場合は、そのフォントが選択されます。

5.4 カスタム・キーの組み合わせの定義

カスタム・キーの組み合わせを定義するには、[編集]→[ユーザキー] を選択します。これによりダイアログ・ボックスが表示されます。このダイアログ・ボックスでは、ObjectScript コマンドを、Alt-Shift-F1 から Alt-Shift-F10 までのいずれかのキーの組み合わせと関連付けることができます。

[OK] と [保存] を続けて選択すると、現在のインスタンスが更新されると同時に、以降のセッション用にキー・シーケンスが保存されます。

出力できない文字をコマンドに含めるには、その文字と同等の 10 進数値 (nnn) を使用します。また、<CR>、<F10>、<F7>、<DO>、<TAB>、<LF>、<ESC>、<CSI>、<NL> (= <CR><LF>) のいずれかを使用できます。

<P1>、<P2> などのコマンド行パラメータも使用できます。

注釈　[ユーザーキー] 機能に関する既知の問題があります。最新の情報は、インターシステムズのサポート窓口にお問い合わせください。

5.5 ユーザ設定の指定

ユーザ設定を指定するには、[編集]→[ユーザ設定] を選択します。これによりダイアログ・ボックスが表示されます。このダイアログ・ボックスでは、ターミナル・アプリケーションによって使用される各種パラメータの現行設定値と初期値の両方を指定できます。この設定は以下のとおりです。

設定	説明
Wrap	右側の列での自動折り返しを設定します。

設定	説明
<- Key sends ^H	<X> キーが Delete ではなく Ctrl-H を送信するように設定します。
Application Keypad	アプリケーション・モードのキーパッドを有効化します。
Force Numeric Pad	標準の PC キーパッドを強制します。
Disable Special ID	特殊なターミナル ID を送信しません。
Disable Mouse Reports	マウス・レポートを送信しません。
Enable Fast Paint	高速描画モードを有効化します。
Paste Keeps Linefeed	クリップボード内の改行を (キャリッジ・リターンと共に) 送信するように設定します。
Pass XOFF Through	リモート・ホストが XOFF/XON を処理するように設定します。
Windows edit accelerators	基本的な編集ショートカット (Ctrl-Insert と Shift-Insert) に加えて、ターミナル・アプリケーションで一般的な Windows の編集ショートカット (Ctrl-C、Ctrl-V、Ctrl-Shift-V) を有効にするかどうかを指定します。 これらの Windows ショートカットが有効化されていない場合は、文字がデータ・ストリームで InterSystems IRIS® データ・プラットフォーム・サーバ・プロセスに渡されます。
Paste burst size (in bytes)	一度の送信に貼り付ける文字数を設定します。
Paste pause time (in msec)	バースト・サイズよりも長いマテリアルが貼り付けられた場合の、連続する送信の間隔をミリ秒単位で設定します。

一部のシステムでは、そのデータ受信速度が、ターミナル・アプリケーションのデータ送信速度よりも遅い場合があります。その場合は、[Paste burst size] で一度に送信するデータ量を指定し、[Paste pause time] で送信間の一時停止時間を指定します。いずれかの設定が 1 未満の場合は、クリップボード全体が一度に送信されます。

5.6 ネットワーク・エンコードの指定

ターミナル・アプリケーションのネットワーク・エンコードでは、以下の場合に文字を変換する方法を制御します。

- キーボード入力が、Unicode を使用してターミナル・アプリケーションのディスプレイ・メモリに変換される場合。キーボードから受け取った文字は、現行の Windows 入力文字セットを使用して、シングルバイトまたはマルチバイトの文字ストリームから変換されます。これは、入力言語を変更した場合に、その変更がアプリケーションによって認識され、対応処理されることを意味します。これによって、複数言語の混在入力が可能となり、混在入力はアプリケーションによって認識され、内部の Unicode 表現に適切に変換されます。複数言語を混在入力する場合は、ネットワーク・エンコードおよび $ZMODE 入出力変換テーブルとして [UTF8] を選択します。
- ターミナル・アプリケーションがピア・サーバと通信する場合。サーバに転送される文字は内部の Unicode 表現からネットワーク・エンコードに変換され、サーバから受け取った文字はネットワーク・エンコードから Unicode に変換されます。

既定のネットワーク・エンコードは UTF8 です。

ネットワーク・エンコードを指定するには、[編集]→[ネットワーク・エンコーディング] を選択します。これにより、ターミナル・アプリケーションで使用するネットワーク・エンコードを選択できるダイアログ・ボックスが表示されます。選択できるエンコードは、[UTF8]、[Windows]、[ISO]、および [EUC] の 4 つです。これらのエンコードのすべてがあらゆる入力ロケールに関連するわけではないため、関連するエンコードのみがメニューに表示されます。

5.6.1 UTF8 エンコード

UTF8 オプションを選択すると、ターミナル・アプリケーションで、内部の Unicode 文字は、サーバへの出力時は UTF8 に変換され、サーバからの受信時は UTF8 から変換されます。UTF8 を選択する場合は、主入出力デバイスの InterSystems IRIS® データ・プラットフォームの入出力変換が UTF8 である必要があります。この入出力変換は、$ZMODE によって確認できます。円記号 (\) で区切られた 4 番目のフィールドにあります。

5.6.2 Windows エンコード

Windows オプションを選択すると、ターミナル・アプリケーションでは、ターミナル・アプリケーションとサーバ間での内部の Unicode 文字セット・エンコードに対する入出力変換に、現行の Windows 入力コード・ページが使用されます。Windows エンコードを使用するときは、InterSystems IRIS 入出力変換 ($ZMODE) を、アクティブな Windows コード・ページが示す文字セットになるように設定する必要があります。

5.6.3 ISO エンコード

ISO オプションを選択すると、ターミナル・アプリケーションでは、ピア・サーバとの入出力変換に、以下の ISO 8859-X コード・ページが使用されます。適切な ISO コード・ページは、現行の Windows 入力コード・ページに基づいて選択されます。有効な対応関係は、以下のとおりです。

言語地域	ISO 標準	Windows コード・ページ	ネットワーク・コード・ページ
西ヨーロッパ	8859-15	1252	28605
中央ヨーロッパ	8859-2	1250	28592
キリル文字	8859-1	1251	28591
ギリシャ語	8859-7	1253	28597
トルコ語	8859-9	1254	28599
ヘブライ語	8859-8	1255	28598
アラビア語	8859-6	1256	28596
バルト・リム語	8859-4	1257	28594
韓国語	iso-2022-kr	949	50225
日本語 (JIS)	N/A	932	50220

ISO ネットワーク・エンコードが選択されている場合は、他のすべての Windows 入力コード・ページで Windows コード・ページが使用されます。

ISO エンコードを使用するとき、$ZMODE で表示される InterSystems IRIS 入出力変換が、ターミナル・アプリケーションによって使用されるアクティブな ISO コード・ページで示す文字セットと矛盾しないように確認する必要があります。

5.6.4 EUC エンコード

EUC エンコードは、極東地域の言語に関連するエンコードであり、一部の UNIX® システムとの通信に使用します。EUC オプションを選択すると、ターミナル・アプリケーションでは、サーバとの入出力変換に、以下のコード・ページが使用されます。適切な EUC コード・ページは、現行の Windows 入力コード・ページに基づいて選択されます。有効な対応関係は、以下のとおりです。

言語地域	ISO 標準	Windows コード・ページ	ネットワーク・コード・ページ
日本語	N/A	932	51932
簡体字中国語	N/A	936	51936
韓国語	N/A	949	51949

日本語 (JIS) のサポートは、50220 コード・ページを使用する ISO ネットワーク・エンコードによって実現され、内部の Unicode との変換が行われます。

5.7 表示の物理文字設定の指定

物理文字設定を指定するには、**[編集]→[物理文字表示の設定]** を選択し、**[論理]** または **[物理]** を選択します。このオプションでは、アプリケーション・ウィンドウに表示される文字の形態を制御できます。両者の相違がわかるのは、マルチバイトの文字セットの使用時のみです。

5.8 関連項目

- ObjectScript シェルのカスタマイズ
- その他のトピック

6

コマンド行からのターミナル・アプリケーションの実行

Windows のコマンド行からターミナル・アプリケーションを起動できます。その際、接続するサーバ、アクセスするネームスペース、実行するコード、およびウィンドウ・サイズと位置を指定できます。ルーチンを実行したり、ターミナル・スクリプトを実行することもできます。

6.1 コマンド行からのターミナル・アプリケーションの起動

Windows のコマンド行からターミナル・アプリケーションを起動するには、以下のいずれかの一般的な形式のコマンドを使用します。

- ローカル・インスタンスに接続する場合

```
iristerm /console=ConnectString Arg1 Arg2 ... ArgN ScriptFilePath
```

- リモート・インスタンスに接続する場合

```
iristerm /server=ConnectString Arg1 Arg2 ... ArgN ScriptFilePath
```

- 接続なしでアプリケーションを起動する場合

```
iristerm Arg1 Arg2 ... ArgN ScriptFilePath
```

注釈　この方法でターミナル・アプリケーションを起動すると、メニュー・バーに追加のメニュー（**[接続]** メニュー）が表示されます。このメニューを使用して、インスタンスに接続できます。

以下はその説明です。

- ConnectString は、接続先のインスタンスを示します。"接続オプション" を参照してください。
- Arg1 ... ArgN は、空白で区切られた追加のオプション引数です。これらの引数は接続するサーバを指定し、セッションの開始環境に関するその他の情報を提供します。実行するルーチンの名前を指定することもできます。
- ScriptFilePath は、実行するスクリプト・ファイルのオプションのパス名です。

PATH 環境変数に InterSystems IRIS® データ・プラットフォーム・バイナリの場所が設定されている場合は、コマンド名 `iristerm` または `iristerm.exe` を使用します。それ以外の場合は、完全なパス名または部分的なパス名を使用する必要があります。InterSystems IRIS の既定のインストールでのバイナリの場所は、install-dir ディレクトリです。¥Bin

6.2 接続オプション

Windows のコマンド行からターミナル・アプリケーションを起動する際は、/console 引数または /server 引数を指定することも、どちらも指定しないこともできます。次のバリエーションが許可されます。

/console=cn_iptcp:HostAddr

この構文は、ターミナル・アプリケーションが TELNET 接続を介してやり取りするターゲット・システムを指定します。これは、ローカル・マシン上でスクリプトを実行するのに便利です。この場合は、HostAddr にローカル・マシンの IP アドレスとポートを指定します。次に例を示します。

```
iristerm /console=cn_iptcp:127.0.0.1[23]
```

/console=cn_ap:Instance[Namespace]

この構文はローカル・インスタンスに接続し、指定されたネームスペースに切り替えます (指定されている場合)。以下に例を示します。

```
iristerm /console=cn_ap:iris[USER]
```

この場合、インスタンス名は iris で、ネームスペース名は USER となります。

ネームスペース名はオプションです。ネームスペース名を指定しない場合は、既定のネームスペースが使用されます。

/console=cn_ap:Instance[Namespace]:Routine

この構文はローカル・インスタンスに接続し、指定されたネームスペースに切り替えて、指定されたルーチンを実行します。以下に例を示します。

```
iristerm /console=cn_ap:iris[USER]:^^%%D
```

この場合、インスタンス名は iris で、ネームスペース名は USER となります。ルーチン名は **^%D** (現在の日付を出力する) となります。

ネームスペース名はオプションです。ネームスペース名を指定しない場合は、既定のネームスペースが使用されます。

ルーチンが終了したら、セッションは閉じます。

/server=ServerName

この構文は、リモート・サーバに安全に接続します。以下に例を示します。

```
iristerm /server=TESTIRIS
```

ServerName には InterSystems IRIS サーバを指定します。使用できるサーバのリストを確認するには、[InterSystems ランチャー] を選択してから、**[優先接続サーバ]** を選択します。サーバのリストが表示されます。

この方法でサーバにアクセスするには、以下を確認してください。

- 所望のサーバにおいて、Telnet サービス (**%Service_Telnet**) が有効であること。(既定ではこのサービスが有効ではないことに注意してください。)

  詳細は、"セキュリティ管理ガイド" の "サービス" を参照してください。

- サーバが起動していること。

UNIX® では、サーバが起動している必要はありませんが、InterSystems IRIS に直接ではなく、シェルにログインすることになります。

6.3 追加の引数

以下の追加の引数を指定することもできます。

/size=RowsxCols

ターミナル・アプリケーション画面の初期サイズを行数と列数で指定します。Rows と Cols はどちらも必ず符号なし整数で指定します。行数と列数の間に入る x は上記のとおりに挿入する必要があります。この制御引数ではスペースは使用できません。

Rows および Cols は次の範囲で指定します。

- 5 <= Rows <= 64
- 20 <= Cols <= 132

/pos=(X,Y)

ディスプレイ・デバイス・ウィンドウに表示されるターミナル・アプリケーション画面の原点の初期値をピクセル単位で指定します。X と Y はどちらも必ず符号なし整数で指定します。X と Y を囲む括弧、およびこれらを区切るコンマは省略できません。この制御引数ではスペースは使用できません。

注釈　ディスプレイ・デバイスのサイズよりも大きな値を X と Y に指定すると、表示領域の外側にウィンドウを配置することができます。ただし、これは行わないようにしてください。

/ppos=(Xpct,Ypct)

ディスプレイ・デバイス・ウィンドウに表示されるターミナル・アプリケーション画面の原点の初期値を表示領域のパーセンテージで指定します。Xpct と Ypct はどちらも必ず符号なし整数で指定します。X と Y を囲む括弧、およびこれらを区切るコンマは省略できません。この制御引数ではスペースは使用できません。

XPct および Ypct は次の範囲で指定します。

- 0 <= Xpct <= 40
- 0 <= Ypct <= 40

つまり、ウィンドウの原点はデバイス原点の上と左側には配置できません。ディスプレイ・デバイスの 40% よりも下または右側にも配置できません。

/UnbufferedLogging

ログがアクティブである場合は、バッファされる代わりに、ログ・ファイルに出力が直ちに書き込まれます。これは、別のプロセスがログ・ファイルの出力を検査しているときに便利です。

6.4 例

6.4.1 例 : バッチ・モードでのスクリプトの実行

この例では、スクリプトをバッチ・モードで実行します。

```
C:\InterSystems\iris\bin\iristerm.exe /console=cn_iptcp:127.0.0.1[23] C:\TestScript.scr
```

6.4.2 例 : ルーチンの実行

この例では、ターミナル・アプリケーションを起動し、基本的なデバッグ・ルーチンの ^%STACK を起動して、現在のユーザおよびターミナル・アプリケーション・プロセスに関する情報を表示します。

```
C:\InterSystems\iris\bin\iristerm.exe /console=cn_ap:iris[USER]:^^%STACK
```

7

その他のトピック（ターミナル・アプリケーション）

ここでは、ターミナル・アプリケーションに関連するその他の項目について説明します。ObjectScript シェルに固有のオプションについて記載されている ″ObjectScript シェルのカスタマイズ″ も参照してください。

7.1 ターミナル・ウィンドウに影響を与えるエスケープ・シーケンス

ターミナル・アプリケーションでは、以下のエスケープ・シーケンスをサポートしています。これらは、ターミナル・ウィンドウに影響を与えたり、ターミナル・ウィンドウの情報を提供します。

シーケンス	結果
`ESC [ 1 t`	ウィンドウをリストアする。
`ESC [ 2 t`	ウィンドウを最小化する。
`ESC [ 11 t`	ウィンドウの状態をレポートする。
`ESC [ 8;rows;columns t`	ウィンドウ・サイズを設定する。
`ESC [ 18 t`	ウィンドウ・サイズをレポートする。

ウィンドウの状態は、以下の READ コマンドの $ZB に次のように報告されます。

- 正常：`ESC [ 1 t`
- 最小化：`ESC [ 2 t`

Set コマンドで行または列が 0 の場合、現在の値は変更されません。サポートされている値の範囲は、行が 10-120、列が 10-160 です。

ウィンドウ・サイズは、以下の READ コマンドの $ZB に `ESC [ 8;rows;columns t` と報告されます。

サイズを変更すると、リセットによってスクロールバックのバッファがクリアされます。また、行の値が大きい場合、ウィンドウを画面に合わせるためにフォント・サイズが小さくなります。

さらに、以下のシーケンスによってウィンドウ・タイトルが設定されます。

```
OSC 2; title ST
```

以下はその説明です。

- OSC（オペレーティング・システム・コマンド）は 7 ビット・シーケンス ESC] または 8 ビット文字 $C(157) です。
- ST（文字列ターミネータ）は 7 ビット・シーケンス ESC ¥ または 8 ビット文字 $C(156) です。

タイトルの最大長は 80 文字です。

例えば、以下の文はターミナル・ウィンドウのタイトルを変更します。

```
write $C(157)_"2;a new title"_$C(156)
```

7.2 キーの時間計測モード

キーの時間計測モードに入る、または終了するには、Alt-Shift-T を押します。

このモードは、さまざまな負荷状況におけるホスト・システムのパフォーマンスの測定に役立ちます。時間計測の実行結果の出力先は、システム・マネージャのディレクトリにある KEYTIMER.LOG ファイルです。

7.3 学習モード

学習モードでは、わずかな編集を行っただけで、ターミナル・アプリケーションにより、簡単にスクリプト・ファイルに変換できるログ・ファイルが生成されます。このモードが有効な場合は、ログ・ファイルが一連の wait for および send スクリプト・コマンドになります。wait for コマンドは、送信したデータに先行する文字を 16 文字まで表示します。

学習モードに入るには、以下の手順を実行します。

1. Alt-L を押してロギングを有効にします。次に、Logging the Session の説明のように、ログ・ファイル名とディレクトリを指定します。
2. Alt-Shift-L を押します。

学習モードを終了するには、Alt-Shift-L を押します。

7.4 ターミナルの閉じるボタンの無効化

ターミナル・アプリケーションの閉じるボタン（X）を無効化する必要がある場合は、以下のようにレジストリ・キーを追加します。

- 32 ビット Windows マシンの場合 ： HKEY_LOCAL_MACHINE¥SOFTWARE¥InterSystems¥Terminal¥NoExit

  ″Terminal″ にあるスペースに注意してください。
- 64 ビット Windows マシンの場合 ：

  HKEY_LOCAL_MACHINE¥SOFTWARE¥ Wow6432Node ¥InterSystems¥Terminal¥NoExit=1

どちらの場合も、NoExit 値は REG_SZ 型です。

7.5 拡張キーボードのマッピング

ターミナル・アプリケーションでは、拡張キーボードに対して、次に示すアプリケーション・キーボード・モードがサポートされます。

キー	マップされた値
Num Lock	PF1
キーパッドの除算記号 (/)	PF2
キーパッドの乗算記号 (*)	PF3
キーパッドのマイナス記号 (-)	PF4
キーパッドのプラス記号 (+)	キーパッドのコンマ
Shift-キーパッドのプラス記号 (+)	キーパッドのマイナス記号 (-)
F1、F2、F3、F4	PF1、PF2、PF3、PF4 (それぞれのキーに対応)
Shift-F1 ...Shift-F10	F11 ...F20 (それぞれのキーに対応)

拡張キーボードのキーパッド部分は、次のようにマップされます。

キー	マップされた値
Insert	ここに挿入
Home	検索
Page Up	前の画面
Delete	削除
End	選択
Page Down	次の画面

Pause キーは、単独の XON/XOFF トグル・キーとして機能します。

7.6 DDE を使用したターミナル・アプリケーションの使用法

ターミナル・アプリケーションは DDE (Dynamic Data Exchange) リンクをサポートすることによって、他のアプリケーションがリモート・ホストとやり取りすることを実現しています。このセクションでは、ユーザが DDE に精通していることが前提となっています。ここで説明するトピックは以下のとおりです。

- Layout ー ステータス情報の取得に使用されます。例えば、行や列のサイズ、接続があるかどうかなどが取得されます。
- Screen ー ターミナル・アプリケーション画面からのデータ収集に使用されます。

- Message ― ターミナル・アプリケーション画面またはホストへのデータの送信に使用されます。

注釈　Windows タスクでは、DDE の使用時にターミナル・アプリケーションの複数インスタンスを区別できません。このため、実行されているターミナル・アプリケーションが 1 つの場合に限り、DDE を使用します。

7.6.1 DDE Layout 接続

ターミナル・アプリケーションは、Layout トピックを通じて、静的情報と見なされるものに対する DDE リクエストをサポートします。

アイテム	返り値の意味
Column	ウィンドウの列数。
Row	ウィンドウの行数。
hWnd	メイン・ウィンドウ・ハンドルの 10 進数の同等値。
Connected	接続がない場合は NULL 文字列、それ以外の場合はタイトル文字列の "mode: node" と同等値。
Read	最後に受信した文字が CTRL/A の場合は 1。この使用目的は、画面描画の末尾の検出です。
Script	スクリプトが実行中の場合は 1、それ以外の場合は 0。
Title	ウィンドウのタイトル。

7.6.2 DDE Screen 接続

ターミナル・アプリケーションは、Screen トピックを通じて、画面データに対する DDE リクエストをサポートします。現在、対象とする画面行部分の選択には、1 つの POKE コマンドを使用できます。

アイテム	返り値の意味
Cursor	row;col 形式での現在のカーソル位置。
Line	現在の行（CR LF を除く）。
LeftLine	現在行のカーソル位置より左の部分（カーソル下の文字は含まない）。
RightLine	現在行のカーソル位置より右の部分（カーソル下の文字を含む）。
All	画面全体（各行は CR LF で改行される）。
Piece	現在選択されている画面行部分（CR LF を除く）。

注釈　アイテム "Piece" は、"RnnCmmLpp" という形式の文字列を使用して POKE コマンドと同様の実行ができます。Piece の要求により、nn 行の mm 列から始まる（最大）pp 文字の文字列が取得されます。画面の左上隅は、行 1、列 1 になります。

7.6.3 DDE Message 接続

ターミナル・アプリケーションは、Message トピックを通じて、データ通信に対する DDE リクエストをサポートします。これらは DDE POKE コマンドによって実装されています。

アイテム	返り値の意味
Send	接続がアクティブな場合、DDE メッセージ値がホストに送信されます。
Display	DDE メッセージ値が、ホストから取得されたかのように ″画面″ に送信されます。