



1.	はじ	はじめに			
	1.1.	CO-GIN	IA とは	4	
	1.2.	インス	トール	5	
		121	インストールの進備	5	
		122	インストール作業		
		123	インストール時のトラブルシューティング	,	
	1.3.	万が一	の場合からの復帰方法		
	1.3.	7513	シックボロン シックアルフェム	10	
	1.4.	/ / ·			
	1.5.	CO-GIN	NA 設定のテクニック	10	
	1.6.	CO-GIN	VA の制限事項	11	
	1.7.	Perl を	導入する		
	1.8.	マニュ	アルの概要		
2.	CO-0	SINA 概要	5	12	
	2.1.	CO-GIN	NA 構成		
		211	co-gina dll	13	
		2.1.1.		13	
		2.1.2.	つ	13	
		2.1.3.	フプラッシュダイアログについて		
		2.1. <del>4</del> . 2.1.5		13	
		2.1.5.	フィークフ		
		2.1.0.			
	2.2	2.1.7. プニ <i>ド</i>	ロフ	20	
	۷.۷.	221			
	2.2	2.2.1. =n	代表的なフラクイン		
	2.3.	設定ノ	アイルの記法		
		2.3.1.	文字コード		
		2.3.2.	コメント・空行		
		2.3.3.	値の設定方法		
		2.3.4.	時間リテラル		
		2.3.5.			
		2.3.6.			
		2.3.7.	メッセージリテラル		
		2.3.8.			
		2.3.9.	特殊変数		
3.	GUI	マクロと	イベント駆動スクリブト	27	
	3.1.	概要			
		3.1.1.	カスタマイズできるダイアログの一覧		
	3.2.	GUIマ	クロ作成の手順		
		3.2.1.	必要なツール		
		3.2.2.	手順の概要		
		3.2.3.	手順① リソースを編集する		
		3.2.4.	手順② GUI マクロへの変換		
		3.2.5.	手順③ エディタで編集		
		3.2.6.	GUI マクロの確認手順		
		3.2.7.	設定ファイルに GUI マクロを指定する		
	3.3.	GUIマ	クロの仕様		
		3.3.1.	GUI マクロで利用可能なコントロールの種類		
		3.3.2.	GUI マクロのタグ一覧		
		3.3.3.	独自に name を設定するときの基準		
		3.3.4.	MSGINA と機能互換にするための名前		
	3.4.	イベン	ト駆動スクリプト		
		3.4.1.	スクリプトの指定		
		3.4.2	起動方法の設定		
		3.4.3.	起動方法設定の例		
		3.4.4.	 特殊変数について		

		3.4.5. スクリプトの出力	
		3.4.6. イベント駆動スクリプトの例	
4.	プラ	ラグインの設定	54
	4.1.	co-notice.dll	
	4.2.	co-logon.dll	
		4.2.1. ログオンダイアログのカスタマイズ	
		4.2.2. logon ダイアログ特有のイベント駆動スクリプトの仕様	
		4.2.3. ユーザー名とパスワードが同じだと許可する例	61
		4.2.4. 独自の認証方法を使う場合の例	
		4.2.5. Active Directory を使う場合の例	
	4.3.	co-logon-mini.dll	
		4.3.1. ダイアログ表示スクリプト	
		4.3.2. Active Directory を使う場合の例	
	4.4.	co-option.dll	
		4.4.1. option ダイアログ特有のイベント駆動スクリプトの仕様	
		4.4.2. chpwd ダイアログ特有のイベント駆動スクリプトの仕様	
		4.4.3. chpwd ダイアログのイベント駆動スクリプトの出力例	
		4.4.4. パスワード変更のソリューション	
	4.5.	co-lock.dll	
		4.5.1. locknotice ダイアログ特有のイベント駆動スクリプトの仕様	
		4.5.2. locklogon ダイアログ特有のイベント駆動スクリプトの仕様	
		- 4.5.3. locknotice ダイアログでのタイムアウトの設定	
	4.6.	co-lock-mini.dll	
	4.7.	co-script.dll	
		4.7.1. prelogoff スクリプトの例	
5.	付録	<u>.</u>	
	5.1.	過去の変更履歴	

# 1. はじめに

# 1.1. CO-GINA とは

### 📕 GINA とは

GINA(Graphical Identification aNd Authentication)は認証などのユーザー識別にかかわる作業を行うライブラリです。

標準の Windows は msgina.dll を利用します。msgina.dll は Microsoft 社によって提供される gina であり、system32 フォルダの中に あります。

GINA には次のような機能があります。

- 認証
- 認証前の画面・ログオン作業中の画面
- パスワード変更
- スクリーンセーバー・スクリーンロック
- ログオン・ログオフ
- シャットダウン・ハイバネーション

### CO-GINA

CO-GINA (シー・オー・ギナ)は GINA を拡張するための開発用ライブラリです。

CO-GINA にはいくつかのプラグインから構成されており、「Ctrl+Alt+Del を押してください」の画面をカスタマイズする co-notice.dll や、 ログオン画面をカスタマイズする co-logon.dll、いろいろなタイミングでスクリプトを実行する co-script.dll などがあります。

CO-GINA をインストールした直後の状態では、msgina.dll と完全に同じ動作をします。各種設定ファイルをカスタマイズすることで、 外部パスワード認証やタイムアウト付きロックなどの機能を利用することができるようになります。

例えば、co-logon.dllを設定すると、認証方法を置き換えたり、ログオン画面の見た目を変えたりすることが可能です。



# 1.2. インストール

### 1.2.1. インストールの準備

CO-GINA をインストールする前に、準備作業が必要となります。準備作業は管理者権限で行ってください。

#### ■ Windowsの設定変更

(Windows XP 利用者のみ)

1. CO-GINA をインストールすると、同時に「ようこそ画面」を利用することはできません。「コントロールパネル」の 「ユーザーアカウント」を開き、「ユーザーのログオンやログオフの方法を変更する」における「ようこそ画面を使用する」の チェックをオフにしてください。

11 ユーザー アカワント ローサー アカワント	
<ul> <li>Ri 2 52 ホーム</li> <li>Bielchi 王</li> <li>アカントと望まる</li> <li>ヘルフ</li> <li>③ 52オンオフリン・</li> </ul>	DYカムとロジオつのオブションとを選択します     Common State (1997)     State State State (1997)     State State State (1997)     State State State (1997)     State State State State (1997)     State State State (1997)     State State State (1997)     State State State State (1997)     State State State (1997)     State State State State (1997)     State State State State (1997)     State S

2. CO-GINA ではログオン時に Ctrl + Alt + Del キーを押す設定にしなければ、一部の機能 (notice 画面や locknotice 画面での タイムアウト)を利用することができません。

これらの機能を利用する場合は、XP Professional では次のようにします。スタートメニューより「ファイル名を指定して実行」 を選択して、「名前」に secpol.msc と入力して「OK」をクリックします。「ローカルポリシー」「セキュリティオプション」の順に選択し、 「対話型ログオン: Ctrl+Alt+Del を必要としない」をダブルクリックして「無効」をチェックしてから OK をクリックします。

II - The Control of Carlo		
	10日 10日 10日 10日 10日 10日 10日 10日	

XP Home では Microsoft サポート文書 KB291559 を参照してください。

(Windows2000 利用者のみ)

1. CO-GINA ではログオン時に Ctrl + Alt + Del キーを押す設定にしなければ、一部の機能 (notice 画面や locknotice 画面での タイムアウト)を利用することができません。

これらの機能を利用する場合は、コントロールパネルから「ユーザーとパスワード」を開き、「詳細」タブの「ユーザーが ログオンする前には必ず Ctrl+Alt+Del キーを押す」をチェックしてください。

<u>1-9-CNX7-</u> F	12
ユーザー 詳細	
· 証明管理	
証明を使用すると、個人、認証局、発行者の識別を行うことができます。	
新儿、证明(如)	
高度なユーザー管理	
ローカル ユーザーとローカル グループは、高度なユーザー管理業務に使用 します。	
Ĩ¥ŧ∎⊘	
ブート時のセキュリティ設定	
ユーザーがログオンする前には必ず Otrl+Alt+Delキーを押すよびに設定す ることなわ勧めします。この最安定ではバスワードによるセキュリティが確実に行 われ、有害なプログラムからシステムを守るのに役立ちます。	
マ ユーザーがログオンする前には必ず Ctrl+Alt+Del キーを押す(B)	
OK キャンセル 適用(A)	

\_

# 1.2.2. インストール作業

インストールは管理者権限で行ってください。GINA のインストールは Windows の根幹にかかわる作業であるため、慎重に作業を 行ってください。万が一、システムが利用できない状態になってしまった場合には、「1.3. 万が一の場合からの復帰方法」をご覧ください。

- 1. 配布された gina-release-2-X-Y-Z.zip を展開します。(X,Y,Z にはバージョン番号を表す数字が入ります)
- 2. C:¥GINA を作成し、展開して現れたファイルやフォルダをその中に移動してください。
- 3. 別途配布されるライセンスファイルを C:¥GINA¥lib¥license.txt に置きます。

ライセンスの入手方法(下記2つのうちどちらかを利用できます)

- 製品版を購入してユーザー登録を行う
- webから試用ライセンスを入手する
   (試用版は利用期限がある点と起動時に「試用版」と表示される点を除いて、製品版との違いはありません)
- 4. C:¥GINA¥lib の中にある installer.exe をダブルクリックして実行します。



- 5. 「次へ」「完了」の順にクリックしてください。「インストールが完了しました」と表示されれば成功です。
- C:¥GINA¥Sampleの中にあるサンプルの設定ファイルやスクリプトのうち必要なものを C:¥GINA の対応するフォルダにコピー してください。全てをコピーする必要はありませんが、C:¥GINA¥Sample¥lib 中の設定ファイルは C:¥GINA¥lib にコピー してください。
- 7. C: ¥GINA に存在するフォルダのうち、C: ¥GINA ¥user 以外のファイルやフォルダに、一般ユーザーがアクセスできないように します。
  - ・ Windows XP Professional の場合
    - C:¥GINA で user フォルダ以外のフォルダを選択してプロパティを表示する
      - →「セキュリティ」タブ →「詳細設定」ボタン
      - →「子オブジェクトに適応するアクセス許可エントリを…」のチェックをはずす
      - →「コピー」ボタンを押す →「OK」ボタンを押す
      - → グループ名またはユーザー名から「Users」を選択して、「削除」ボタンを押す
      - → OK ボタンを押してプロパティダイアログを閉じる
      - → ReadMeFirst.txt に関しても同様にする
  - ※ Windows XP Home では上記の方法でセキュリティを設定することはできません。そのため、MS-DOS プロンプトで「cacls.exe」を利用してアクセス権を設定するか、別のアクセス権設定ツールを利用してください。また、ファイルシステムが FAT32 である場合には十分なセキュリティーが確保できませんのでご注意ください
- 8. 再起動してください。
- 9. 管理者権限をもつアカウントでログオンし、C:¥GINA¥log¥gina.log が作成されているか確認します。
- (C.¥GINA¥lib¥Sample¥co-gina.ini を C.¥GINA¥lib¥co-gina.ini にコピーしていない時には C.¥GINA¥lib¥co-gina.log にログが作成されます) 10. 念のため、ユーザー権限のアカウントでも従来通りログオンできることを確認してください。

# ご注意ください

CO-GINA は必ずしも「C:¥GINA」にインストールする必要はありません。ただし、「C:¥GINA」以外の場所に インストールされた方は、設定ファイルに記載されているパスを適宜、書きかえた上で、以下のマニュアルに 出てくる「C:¥GINA¥lib¥co-gina.dll」などのパスを自分の環境のパスに置き換えて読み進めてください。



# 1.2.3. インストール時のトラブルシューティング

CO-GINA をインストールして再起動したときに、エラーダイアログが表示されたときには、このトラブルシューティングを参考にして 設定を修正してください。なお、修正を行うには「1.3. 万が一の場合からの復帰法」に従い、一旦 CO-GINA をアンインストールしてから 行うことをお薦めします。

### ■ CO-GINA をインストールできていない場合



### ■ ライセンスファイルが正しく認識されない場合



以下に、ライセンスが正しく認識されないときに最初に表示されるダイアログ毎の対策方法を示します。

ダイアログ	解決方法
ライセンスファイルを開くことができませんでした。 co-gina.ini の設定を見直してください。	co-gina.ini にライセンスファイルの場所を正しく指定できているか確認 してください。
	▶ co-gina.dll と同じフォルダに co-gina.ini がありますか?
	▶ co-gina.ini の中で指定してあるライセンスファイルの場所
	([license]のpath)にライセンスファイルが置いてありますか?
ライセンスのバージョンが異なります。	ライセンスファイルのバージョンが旧バージョンのものです。
	▶ 最新バージョンのライセンスを入手する方法については、
	別途お問い合わせください。
ライセンスファイルの形式が不正です。弊社から提供	ライセンスファイルが不正なものと判断されたときに、このエラーが
されたものかどうかご確認ください。	表示されます。
	▶ co-gina.ini の中で指定してあるライセンスファイルの場所
	( [license] の path ) のライセンスファイルが弊社提供のもの
	ですか?
ライセンスの有効期限が過ぎているため、CO-GINA を	ライセンスの有効期限が過ぎてしまったため、CO-GINAを利用
利用できません。	できません。
	▶ 最新版のライセンスを入手してください。
ライセンスが正しくないため、CO-GINAを利用	ライセンスが正しくありません。
できません。	▶ 弊社提供のライセンスの内容をそのままライセンスファイルに
	記述できていますか?
co-gina.dllの設定ファイルがライセンスに登録した	設定ファイルが改変されているか壊れています。
設定ファイルと異なっているため起動できません。	▶ ライセンスファイル、設定ファイルを確認してください。
XX プラグインの設定ファイルがライセンスに登録した	設定ファイルが改変されているか壊れています。
設定ファイルと異なっているため起動できません。	▶ ライセンスファイル、設定ファイルを確認してください。
(XXには logon や lock などのプラグイン名が入ります)	

# 1.3. 万が一の場合からの復帰方法

ログオンできない等の状態になってしまった端末がネットワーク経由でアクセスできる場合

- 1. レジストリエディタ (regedt32.exe) を開きます。メニューの [レジストリ] → [コンピュータの選択] から復帰させたい端末を 選択してください。
- 2. HKLM¥software¥microsoft¥WinNT¥CurrentVersion¥Winlogon¥GinaDIIをmsgina.dll に変更してください。
- 3. ログオンできなくなった状態の端末を再起動してください。

ネットワーク経由でアクセスできない場合

- 1. 起動時にセーフモードで起動してください。(ネットワークは選択しない)
- 2. Administrator でログオンし、C:¥GINA¥lib¥gina-uninstall.vbs を実行してください。
- 3. 再起動してください。

# 1.4. アンインストール

CO-GINA をアンインストールするには次のようにしてください。アンインストール作業を行うと、Microsoft 製のデフォルトの GINA である msgina.dll が有効になります。

- 1. 管理者権限でログオンし、C:¥GINA¥lib¥gina-uninstall.vbs をダブルクリックしてください。
- 2. 「標準のものに戻します。よいですか?」という質問に「はい」と答えてください。「アンインストールが完了しました」と表示 されると成功です。
- 3. 再起動します。
- 4. 管理者権限でログオンし、CO-GINAが作成したレジストリを削除します。レジストリエディタ (regedt32.exe)を開き、 「HKEY\_LOCAL\_MACHINE¥Software¥co-conv¥co-gina」キーを削除してください。
- 5. 「C:¥GINA」フォルダが不要ならば削除してください。

# 1.5. CO-GINA 設定のテクニック

GINA のフォルダをネットワーク共有

CO-GINA をインストールした端末と同じネットワークに別の端末がある場合は、ネットワーク経由で設定ファイルや各種スクリプトを 編集することができます。これにより、いちいちログオンしなくても設定ファイルを編集することができ、設定を行うときの手間が大幅 に削減されます。

C:¥GINA を共有するには CO-GINA をインストールした端末で次のようにします。

(Windows XP Home 利用者の方は)

 エクスプローラで「C:¥GINA」を右クリックし、プロパティーの共有タブから「ネットワーク上でこのフォルダを共有する」を チェックしてください。さらに、「ネットワーク ユーザーによるファイル変更を許可する」をチェックして OK ボタンを押して ください。

(Windows XP Professional 利用者の方は)

- Professional では「簡易ファイルの共有」を利用しない方が詳細なアクセス権を設定できます。エクスプローラのメニューの「ツール」から「フォルダオプション」を選択してください。「表示」タブの詳細設定のうち「簡易ファイルの共有を使用する」のチェックをはずして OK ボタンを押してください。
- 2. エクスプローラで「C:¥GINA」を右クリックし、プロパティーの共有タブから「このフォルダを共有する」を選択してください。 その後、「アクセス許可」でアクセス権を指定します。

#### ■ 再起動しなくても反映される項目

CO-GINA はシステムの根幹にかかわる製品であるため、設定を確認するのが煩雑になってしまいます。CO-GINA において再起動 しなければ反映されない設定項目と、再起動しなくても反映される設定項目をきちんと理解すると設定作業の効率が上がります。 CO-GINA において再起動しなければ設定が反映されない項目は次のものです。

- co-gina.ini のログの設定 (log セクションの設定項目)
- co-gina.iniのステータスの設定 (status セクションの設定項目)
- co-gina.ini のライセンスの設定 (license セクションの設定項目)
- encrypt.txt の暗号化の設定

co-gina.ini で利用するプラグインの設定は、プラグイン実行時に毎回読みこまれます。

各プラグインの設定は、プラグインの処理開始時に読み込まれます。(例えば co-logon.dll ならログオンダイアログ表示前、 co-script.dll ならスクリプト実行前です)

また、イベント駆動スクリプトの設定は、イベント発生時に読まれます。

### 1.6. CO-GINA の制限事項

CO-GINA をインストールすると、ファイル共有やリモートデスクトップ、telnet サービスなどいくつかのサービスが利用できなくなる 可能性があります。

# 1.7. Perl を導入する

Perl の導入は CO-GINA を利用する上で必須ではありません。しかし、CO-GINA に付属する各種のサンプル スクリプトのいくつかは Perl を利用して書かれています。そのため、これらのサンプルを利用する場合は、Perl を導入してください。

Perl を導入するには次の手順に従ってください。

http://www.activestate.com/Products/ActivePerl/ に接続し、"Download" を選択してダウンロードします。ここでは最新の 5.10.0.1004 をダウンロードしたものとします。インストーラを起動して、標準設定でインストールします。

### 1.8. マニュアルの概要

「2. CO-GINA 概要」では CO-GINA の基本的な設定方法について解説します。「2.1. CO-GINA 構成」では CO-GINA のファイル構成および co-gina.ini の設定方法について説明します。「2.2. プラグイン仕様」ではプラグインの概要について説明します。「2.3. 設定ファイルの 記法」では CO-GINA の設定ファイルを記述する方法について説明します。

「3. GUI マクロとイベント駆動スクリプト」では、デザインやレイアウトをカスタマイズする GUI マクロと、イベント発生時に起動 されるスクリプトの設定方法について説明します。

「4. プラグインの設定」では、実際に各プラグインの設定方法や設定例を説明します。

CO-GINA の概要について知りたい方は、「2.1. CO-GINA 構成」と「2.2. プラグイン仕様」をご覧ください。 CO-GINA の詳細な設定方法を知りたい方は「2.3. 設定ファイルの記法」と「3. GUI マクロとイベント駆動スクリプト」をご覧ください。 CO-GINA の具体的な設定方法や設定例を知りたい方は「4. プラグインの設定」の各プラグインの説明をご覧ください。

# 2. CO-GINA 概要

# 2.1. CO-GINA 構成

CO-GINA のパッケージは次のようなファイルから成り立っています。ここでは、これらのファイルのうち重要なものに関して解説します。

📁 GINA	
📁 doc	ドキュメントがあるフォルダ
📁 admin	管理者権限で動作するスクリプトを格納するフォルダ
🃁 user	利用者権限で動作するスクリプトを格納するフォルダ
📁 gui_macro	GUI マクロを格納するフォルダ
📁 lib	CO-GINA の DLL ファイルが含まれるフォルダ。.ini ファイルもここに置きます。
у co-gina.dll	CO-GINA の根幹となる DLL ファイル
🐋 co-guimacro.dll	GUI マクロを表示するための DLL ファイル
🖻 co-guimacro.idlist	GUI マクロの名前の対応付けを保存しているファイル (通常は編集しないでください)
co-lock.dll	lock プラグイン ( ロック画面で GUI マクロとイベント駆動スクリプトを利用する )
co-lock-mini.dll	lock プラグイン ( 任意プログラムをロック画面として利用する )
co-logon.dll	logon プラグイン ( ログオン画面で GUI マクロとイベント駆動スクリプトを利用する )
co-logon-mini.dll	logon プラグイン ( 任意プログラムをログオン画面として利用する )
🖠 co-notice.dll	notice プラグイン
👏 co-option.dll	option プラグイン
🖠 co-script.dll	script プラグイン
💰 gina-uninstall.reg	CO-GINA のアンインストーラ ( msgina を利用するよう設定する )
🚞 installer.exe	CO-GINA のインストーラ
🃁 Sample	設定ファイルのサンプルがあるフォルダ
📁 admin	管理者権限で動作するスクリプトのサンプルが置いているフォルダ
📁 gui_macro	GUI マクロのサンプルが置いてあるフォルダ
📁 dll	GUI マクロを生成するための DLL が置いてあるフォルダ
📁 lib	設定ファイルのひな形が置いてあるフォルダ
📁 user	利用者権限で動作するサンプルのスクリプトが置いているフォルダ
📁 tools	CO-GINA と組み合わせて利用する実行ファイルのサンプルがあるフォルダ
ReadMeFirst.txt	

📕 lib フォルダ

lib フォルダには CO-GINA の動作を行うバイナリファイルが格納されています。このフォルダに設定ファイルを置いてください。 設定ファイルのサンプルは C:¥GINA¥Sample¥lib にあります。

#### Sample フォルダ

Sample フォルダには、CO-GINA を利用するための各種スクリプトや設定ファイルが格納されています。 なお、これらのサンプルを利用する際には、ファイルを C:¥GINA の対応する場所にコピーしてからお使いただくことを推奨して おります。(例:C:¥GINA¥Sample¥user¥logoff\_check.vbs は C:¥GINA¥user¥logoff\_check.vbs にコピーしたものを使う)

#### 📕 tools フォルダ

CO-GINA で利用するいくつかのツールが格納されています。tools フォルダ中の tools.txt にそれぞれのファイルの解説が記述されています。

### 2.1.1. co-gina.dll

#### ■ 概要

```
co-gina.dll は Microsoft 社製の msgina.dll に代わる DLL ファイルです。
```

CO-GINA をインストールしたあとにおいても、ini ファイルが初期設定の状態では、msgina.dll と同じ挙動をします。co-gina.dll の 設定ファイル co-gina.ini を編集すると、例えば co-script.dll を利用して色々なタイミングでスクリプトを実行できるようになります。 これらの co-gina.dll から利用できる DLL ファイルのことをプラグインと呼びます。

### 2.1.2. co-gina.ini

co-gina.ini は co-gina.dll の設定ファイルであり、C:¥GINA¥lib¥co-gina.ini に存在します。 co-gina.ini ではライセンスファイルやログの設定、プラグインの設定などを行います。

#### ■ 設定ファイルの例

Sample フォルダ内の co-gina.ini は次のような内容です。設定ファイルのフォーマットに関しては、「2.3. 設定ファイル」をご覧ください。



```
#利用するプラグインを設定する
#-
[plugin]
# co-notice プラグインを利用するときにはコメントをはずす
#notice=C:¥GINA¥lib¥co-notice.dll
# co-notice プラグインの設定ファイルを変更するには以下のようにする [パスリテラル]
#notice.ini=C:¥GINA¥lib¥co-notice.ini
# Web から設定ファイルを更新するには notice.ini のコメントをはずした上で
# 以下のようにする
#notice.ini.update=http://xxxxxxx/co-notice.ini
                                                                   「2.1.3. プラグインの設定」
#notice.ini.timeout=10s
                                                                   で説明
#notice.ini.retry=0
# co-logon プラグインを利用するときにはコメントをはずす
#logon=C:¥GINA¥lib¥co-logon.dll
# co-logon-mini プラグインを利用するときにはコメントをはずす
#logon=C:¥GINA¥lib¥co-logon-mini.dll
# co-logon プラグインの設定ファイルを変更するには以下のようにする [パスリテラル]
#logon.ini=C:\u00e4GINA\u00e4lib\u00e4co-logon.ini
#logon.ini=C:\GINA\Lib\Loo-logon-mini.ini
```

# 2.1.3. プラグインの設定

co-gina.ini の重要な役割の1つに、利用するプラグインの選択があります。ここでは logon プラグインを有効にする方法を例に説明 します。

まず、初期設定においては次のようになっています。

[plugin] #logon=C:¥gina¥lib¥co-logon.dll

これは、logon プラグインを利用しない状態を表します。この時、ログオン画面では通常の msgina.dll と同じ動作を行います。

logon プラグインとして「co-logon.dll」を用いる場合は次のようにします。



logon プラグインとして「co-logon-mini.dll」を用いる場合は次のようにします。

[plugin] logon=C:¥gina¥lib¥co-logon-mini.dll

このようにすれば、logon プラグインとして co-logon-mini.dll が用いられます。

利用できるプラグインの種類の一覧は「2.2. プラグイン仕様」で列挙しています。

```
co-gina-ini ではプラグインの設定ファイルのパスを変更することができます。
例えば、logon プラグインの設定ファイルは次のようにして指定します。
```

[plugin]	
logon=C:¥GINA¥lib¥co-logon.dll	
logon.ini=C:¥GINA¥lib¥co-logon.ini	

設定ファイルが指定されなかったときは、プラグインと同じ場所にある ini ファイルが設定ファイルとして利用されます。

$\left( \right)$	<b>1</b> 補足
	プラグインの設定ファイルの指定はパスリテラルであるため、Web 上に最新の設定ファイルを置いておけば、 自動的に設定ファイルが更新されるようになります。Web から最新版を取得するには次のようにします。
	[plugin]  ogon=C:¥GINA¥co- ogon.d    ogon.ini=C:¥GINA¥co- ogon.ini
	logon.ini.update=http://www.xxxxxx.co.jp/tmp/co-logon.ini logon.ini.update.timeout=10s logon.ini.update.retry=3
	詳しくは、「2.3.5. パスリテラル」をご覧ください。

# 2.1.4. スプラッシュダイアログについて

CO-GINA は、電源起動時およびログオフ直後に次のような「スプラッシュダイアログ」を表示します。 製品版のライセンスを使用しているときには「これは試用版です」は表示されません。

Powered by CO-	GINA	
Powered by	<b>CO</b> -	-GINA
		株式会社 CO-CONV.
パージョン情報: ファイル名 製品 バージョン co-ginadil co-notice.dil co-lock.dil co-sptian.dil co-sptian.dil co-sptian.dil	<u>) パージョン</u> 2,4,1,0 2,4,2,1 2,4,1,0 2,4,2,3 2,4,1,0 2,4,1,0 2,4,1,1	これは試用版です
開じる		

スプラッシュダイアログを表示しないようにするためには、co-gina.iniの「license」セクションに「splash=no」と設定してください。 ただし、試用版ではこのように設定してもスプラッシュダイアログは表示されます。

```
[license]
# ライセンスファイルへのパスをここに記述する(要再起動)[パスリテラル(Web更新不可能)]
# 弊社から提供されたライセンスの内容をこの場所に記述する
path=C:¥GINA¥|ib¥|icense.txt
# 起動時にスプラッシュ ウインドウを表示するかどうか [真偽リテラル]
# 試用期間中はここで表示しない設定にしても消えません
splash=no
```

### 2.1.5. ライセンスファイル

弊社から提供されたライセンスファイルを一般ユーザーからアクセスできないフォルダに置き、そのパスを co-gina.ini に記述してください。

```
[license]
path=C:¥GINA¥lib¥license.txt
```

ライセンスファイルが正しくない場合や試用期限を過ぎた場合は、電源投入時に、「ライセンスファイルが不正です」といったメッセージ が表示されます。このエラーが表示されたときは、「1.2.3. インストール失敗時のトラブルシューティング」を参照してください。

### 2.1.6. ステータス

#### ■ 概要

Windows はログオン状態やロック状態など様々な状態をとります。CO-GINA ではこのような状態のことを「ステータス」と呼び、 ステータスに関係する情報をレジストリやファイルに出力することができます。

### ■ 出力先

出力先はレジストリやファイルなど複数の出力先を指定することができます。 レジストリにステータス情報を出力するには次のようにします。

```
[status]
# GINA の状態を出力するファイルをここに記述する
# レジストリに出力するには次のようにする
path1=registry
```

出力先は「HKEY\_LOCAL\_MACHINE¥Software¥co-conv¥co-gina¥status」です。出力は全て REG\_SZ 形式 (文字列) で行われます。 指定したファイルに INI ファイル形式で出力するには次のようにします。

```
[status]
path1=C:¥GINA¥log¥status.txt
```

レジストリとファイルに出力するには次のように、path1, path2, path3…と順番に指定してください。

```
[status]
path1=registry
path2=C:¥GINA¥log¥status.txt
```

なお、CO-GINA は電源起動時に前回終了時のステータスを読み取りますが、その際は path1 に書かれている場所から読み取ります。 出力先の変更を有効にするためには再起動が必要です。



### 📕 出力例

電源を起動して notice のダイアログを表示したときには次のような情報が出力されています。ここではファイルに出力した場合の出力 例を示します。

[status] CurStatus=100 CurStatusStr=notice CurStatusModified=2003/04/01 01:08:03 logoff=1 PrevStatus=001 PrevStatusStr=onboot PrevStatusModified=2003/04/01 01:08:01 MsUserName= MsDomain= LogonUserName= LogonDomain= LogonTime=

ログオン直後は次のような情報が出力されています。

[status]
CurStatus=201
CurStatusStr=loggedon
CurStatusModified=2003/04/01 01:08:42
logoff=0
PrevStatus=200
PrevStatusStr=onlogon
CurStatusModified=2003/04/01 01:08:42
MsUserName=test
MsDomain=MYPC
LogonUserName=test
LogonDomain=MYPC
LogonTime=2003/04/01 01:08:41

#### それぞれの値の内容は以下の通りです。

	CurStatus	現在の状態を3桁の数字で表します。	
	CurStatusStr	現在の状態に対応する文字列を表します。	
	CurStatusModified	現在の状態になった時間。	
状態情	logoff	現在ログオフしているかどうかを表します。 1 がログオフ、0 がログオンを表します。 ログオン認証が終わった後に0になり、ログオフ・シャットダウン時に1に設定 されます。	
報	PrevStatus	1つ前の状態を3桁の数字で表します。	
	PrevStatusStr	1つ前の状態を文字列を表します。	
	PrevStatusModified	1つ前の状態になった時間。	
ア	LogonUsername	利用者がログオンダイアログに入力したユーザー名。	
カウ	LogonDomain	利用者がログオンダイアログに入力したドメイン。	
ント	MsUsername	実際に Windows にログオンしたときのユーザー名。	
雨報	MsDomain	実際に Windows にログオンしたときのドメイン。	
	LogonTime	利用者がログオンした時間を表します。	

アカウント情報はログオン時に設定され、ログオフが完了後に初期化されます。

LogonUsername は利用者がログオンダイアログに入力したユーザー名を表し、MsUsername は Windows にログオンしているユーザー 名を表します。特別な設定をしなければこの2つは同じ値となりますが、logon プラグインの設定によっては、利用者がダイアログに 入力したユーザー名と実際にログオンするユーザー名が異なる場合があります。

また、logoff は、電源投入時スクリプトで「正常なシャットダウンであったかどうか」を判断するために用いられます。電源投入時 スクリプトを呼び終わるまでは logoff の情報は前回起動時の値のままです。電源投入時スクリプトの呼び出しが終了した段階で logoff は 1 に設定されます。

### ■ ステータスの数字について

ステータスには以下のような数字が用いられます。各状態間の遷移については次ページの図を参照してください。

起動	001	onboot( co-script の電源起動時スクリプトを実行中 )		
ログオフ状態	100	notice(「Ctrl+Alt+Del を押してください」のダイアログを表示中 )		
	110	logon(ログオン認証を行うダイアログを表示中)		
	200			
	201	 oggedon(ログオンした状態)		
	210	option( ログオン中に Ctrl+Alt+Del を押した状態 )		
ログオン中	211	chpwd(パスワード変更を行うダイアログを表示中)		
	220	ocknotice(ロック通知ウインドウを表示中 )		
	221			
	230	prelogoff( co-script のロック前スクリプトを実行中 )		
	300	logoff( co-script のログオフスクリプトを実行中 )		
	301	 shutdown( co-script のシャットダウンスクリプトを実行中 )		
ログオフへの移行	302	shutdown( co-script のシャットダウンスクリプトを実行中。パワーオフ時 )		
	303	shutdown( co-script のシャットダウンスクリプトを実行中。再起動時 )		
	304	shutdown( co-script のシャットダウンスクリプトを実行中。スリープ時 )		
	305	shutdown( co-script のシャットダウンスクリプトを実行中。スリープ時 )		
	306	shutdown( co-script のシャットダウンスクリプトを実行中。ハイバネート時 )		
	400	screensaver(co-script のスクリーンセーバー前スクリプトを実行中)		

# ご注意ください

スクリーンセーバーを表示しているときの PrevStatus は 400( screensaver )、CurStatus はスクリーンセーバー を表示する前のステータスになります。

(例)

- noticeの状態からスクリーンセーバーが起動したとき、スクリーンセーバー起動中のステータスは 「PrevStatus=400, CurStatus=110(noticeの番号)」となっています。
- loggedonの状態からスクリーンセーバーが起動したとき、スクリーンセーバー起動中のステータスは 「PrevStatus=400, CurStatus=201 (noticeの番号)」となっています。
- loggedon の状態からスクリーンセーバーが起動し、マウスを動かして locknotice に移ったときのステータスは 「PrevStatus=201, CurStatus=220 (locknotice の番号)」となります。

### ■ CO-GINA 状態遷移図

以下に CO-GINA の状態遷移図を示します。なお、枠内の右下の数字はステータスとして用いられる数字を表します。



# 2.1.7. ログ

ログを出力する場所を co-gina.ini の「log」セクションの path で指定してください。指定がない場合は DLL と同じフォルダの co-gina. log に出力します。

このpathは、path2、path3……と複数指定することができます。最初の指定はpath1ではなくpathとなることに注意してください。

また、ログには詳しさのレベルがあります。レベルは0~5まであり次のようになっております。デフォルトの設定はレベル2です。

0	ログを出力しない
1	エラー
2	警告
3	設定確認用メッセージ
4	開発者用メッセージ
5	開発者用メッセージ (パスワードを含む)

レベル5に設定すると、ログにスクリプト実行時の変数(パスワード情報を含む)が記録されます。それらの情報を記録する必要が ない場合はレベル4に設定してください。CO-GINA で各種の設定を試験するときは レベル3 で確認し、運用時にはレベル2にするのが 望ましいでしょう。

記録されるログのサイズを size で指定することができます。ログが指定されたサイズを超えると、バックアップを作成し、新しいファイルにログを記録します。size の指定がないときには、同じファイルにログを書き続けます。

count にバックアップの個数を指定することができます。デフォルトの値は 10 です。0 ないしはマイナスの値を指定した場合は バックアップを作成せず、現在存在するファイルの内容を空にして書き込みをします。

ログの設定を変更したときには、再起動するようにしてください。

#### ■ ログの設定例

レベル2のログを C:¥GINA¥log¥gina.log に出力する例は以下の通りです。

path=C:¥GINA¥log¥gina.log level=2 # size=0 # count=1

レベル 5 のログを C:¥GINA¥log¥gina-debug.log に出力する例は以下の通りです。この例では 1,000,000 byte (1M バイト) ごとに、 gina-debug.log.0、gina-debug.log.1 ……を 10 個まで作成します。

path=C:¥GINA¥log¥gina-debug.log level=5 size=1000000 count=10

ログを複数出力する例を以下に示します。

```
path=C:¥GINA¥log¥gina.log
level=2
# size=0
# count=1
path2=C:¥GINA¥log¥gina-debug.log
level2=5
size2=1000000
count2=10
```



path や level は DLL が再読み込みされたときに有効になります。co-gina.dll は、起動時にのみ読み込まれる ので再起動後にログの場所が変わります。ところが、各種プラグインは呼び出されるたびに設定ファイルを 読み込むため、直ちにログの場所が変わります。大きな混乱を招いてしまうことがあるため、co-gina.ini の level や path を編集したときは、通常は再起動するようにしてください。

# 2.2. プラグイン仕様

CO-GINA のプラグインは、DLL として提供されます。プラグインの種類には次のようなものがあります。

notice	Ctrl+Alt+Del を押してくださいの画面を表示する
logon	ログオンダイアログの表示、認証、プロファイルの設定を行う
option	ログオン中に Ctrl+Alt+Del を押したときのセキュリティダイアログを表示する
lock	ロックの解除ダイアログを表示する
script	各種スクリプトをいろいろなタイミングで実行する

# 2.2.1. 代表的なプラグイン

弊社が用意するプラグインを列挙します。それぞれのプラグインの具体的な使い方については「4. プラグインの設定」で説明します。

co-notice.dll	<ul> <li>Ctrl+Alt+Del を押してくださいの画面に任意の BMP ファイルを表示できる</li> <li>一定時間キーボードを触らないと、電源切断を行う</li> </ul>
co-logon.dll	<ul> <li>GUI マクロによるログオン ダイアログのカスタマイズ</li> <li>ボタンが押されたときに特殊変数つきで外部のスクリプトの呼び出し</li> <li>スクリプトからの任意のエラーメッセージを返す</li> <li>スクリプトからどのようなユーザーでログオンするかを設定する</li> </ul>
co-logon-mini.dll	<ul> <li>ダイアログを表示する実行ファイルを作成し、スクリプトが出力したアカウント 情報に基づいてログオン処理を行う</li> </ul>
co-option.dll	• パスワード変更時の動作拡張
co-lock.dll	・ タイムリミットつきロック ダイアログの表示
co-lock-mini.dll	<ul> <li>ダイアログを表示する実行ファイルを作成し、スクリプトが出力したアカウント 情報に基づいてロック解除処理を行う</li> </ul>
co-script.dll	<ul> <li>・設定ファイルに記述された各種スクリプトを起動時やログオフ処理直前・ログオフ時・ シャットダウン時などのタイミングで実行する</li> </ul>

# 2.3. 設定ファイルの記法

ここでは co-gina.ini や各種プラグインの設定で使用する設定ファイルの記法に関して説明します。

### 2.3.1. 文字コード

文字コードは Shift JIS 、改行コードは CR+LF で記述します。メモ帳で編集する場合には問題となりませんが、その他のエディタを 使用する場合は注意してください。

### 2.3.2. コメント・空行

# (半角英数字)から始まる行はコメントとして扱います。コメントは人間が見て理解するためのものであり、CO-GINA は解釈しません。 空行も同様に無視されます。

# これはコメント

# すぐ上に空行がありますが、これも無視されます

### 2.3.3. 値の設定方法

値を設定するには次のようにします。

[セクション名] 設定項目 = 値

設定ファイルごとに、どのセクション・項目にどういう値を入力できるかが決まっています。これらの項目に関しては、「4. プラグインの設定」 をご覧ください。

### 2.3.4. 時間リテラル

時間を表すときには時間リテラルで記述する必要があります。時間リテラルは、数字と単位で構成されています。



数字の部分は -1 より大きい整数で記述します。(小数は使えません)数字と単位の間には空白を入れてはいけません。正しく解釈 できなかった場合には、0 であると解釈されます。-1 の場合は無限の時間を表し、単位は不要です。

単位には次のものを用いることができます。

ms	ミリ秒
S	秒
m	分
h	時間
d	日

単位を省略したときには s として解釈されます。

以下に時間リテラルの例を挙げていきます。

```
# 100 ミリ秒
100ms
# 1000 秒
1000s
# 1000 秒
1000
# 無限の時間
-1
# 10 と ms の間に空白があるため、単位なしとして解釈されてしまう。よって、10 秒になる。
10 ms
# 解釈できないので 0 秒
aaa
```

ただし、10msなどの短い時間を指定した場合は、多少の誤差が出ることがあります。

### 2.3.5. パスリテラル

特定のファイルへのパスを指定する場合に利用します。必ず絶対パスで指定するようにしてください。パスの中にスペースが含まれて いても問題ありません。

以下に例を挙げます。

値が指定されなかったときの挙動は、用いられる場所によって異なります。各項目の説明を参照してください。

#### ■ パスで指定したファイルの更新

パスリテラルでは、ファイルをWebからダウンロードして常に最新のものにすることができます。この機能により、複数の端末の 設定やスクリプトを一斉に更新でき、管理コストを大幅に削減することができます。

```
path=C:\Program Files\Internet Explorer\iexplore.exe
path=C:\Program Files\Internet Explorer\iexplore.exe
```

以下に例を示します。パスリテラルを解釈するたびに http://www.xxxx.co.jp/tmp/xxxxx.bmp に最新版をダウンロードし、C:¥GINA¥xxxx.bmp を上書きします。

```
bitmap=C:¥GINA¥xxxx.bmp
bitmap.update=http://www.xxxxxx.co.jp/tmp/xxxxx.bmp
bitmap.update.timeout=10s
bitmap.update.retry=1
```

パスリテラルの項目名 (この場合は「path」) に「.update」をつけたした項目名で、Web からダウンロードするときの URL を指定します。 URL が指定されていないときは、ファイルの更新は行いません。

パスリテラルの項目名に「.update.timeout」をつけたした項目名で、ダウンロードするときのタイムアウト値を指定できます。 Web サーバーから timeout の時間が経過しても応答がないとダウンロードは失敗します。タイムアウトの指定は任意です。指定がない場合は 「-1(無限時間)」となります。

パスリテラルの項目名に「.update.retry」をつけたした項目名で、ダウンロードに失敗したときに再試行を行う回数を指定できます。 指定がないときのリトライ回数は0回です。(1度しかダウンロードを試みません)



### 2.3.6. 実行リテラル

実行するプロセスに関する情報を記述します。コマンドライン引数も与えることができます。実行ファイルまでのパスや引数にスペース 文字が含まれている場合は、"で囲む必要があります。デフォルトの動作は状況によりますが、明記されていない場合は何も実行しません。 プロセスを設定するときには念のためにフルパスで指定してください。

以下に例を示します。

path=C:\u00e4windows\u00e4system32\u00e4notepad.exe
path=C:\u00e4windows\u00e4system32\u00e4notepad.exe C:\u00e4sample.txt
path=C:\u00e4windows\u00e4system32\u00e4notepad.exe "C:\u00e4Documents and settings\u00e4sample.txt"
path="C:\u00e4Program Files\u00e4Internet Explorer\u00e4iexplore.exe" http://xxxxx.jp/

#### ■ ファイルの更新

実行リテラルでは、実行するファイルやスクリプトを Web から最新版をダウンロードすることができます。

更新機能を用いるには次のようにします。

```
path=C:¥perl¥bin¥perl.exe C:¥GINA¥admin¥auth.pl C:¥GINA¥message.txt
path.file1=C:¥GINA¥admin¥auth.pl
path.file1.update=http://www.xxxxx.co.jp/tmp/auth.pl
path.file2=C:¥GINA¥message.txt
path.file2.update=http://www.xxxxx.co.jp/tmp/message.txt
path.file2.update.timeout=10s
path.file2.update.retry=3
```

実行リテラルに「file1」「file2」というように複数の更新ファイルを指定することができます。それぞれの「file 〇」の更新設定の書き方は、「2.3.5. パスリテラル」の「パスで指定したファイルの更新」を参照してください。

なお、「file 〇」は、1から順に解析され、update の指定がなくなった段階で更新作業を終了します。例えば、「file1」「file2」と「file4」の設定がされていたとき、CO-GINA は「file1」から順にファイル更新をしていき、「file3」の設定がされていない時点でファイル更新を 終了し、「file4」の更新は行われません。path.file1 path.file2 はこの順番に設定ファイルに記述する必要はありません。

### 2.3.7. メッセージリテラル

複数行の文字列を表すために用いられます。デフォルト値は状況によって異なります。

次のような変数が利用可能です。

¥¥	¥ マークを表す
¥n	改行
¥t	タブ文字

以下に具体例を挙げます。

message= ログオンしているユーザー名は domain ¥¥ test さんです。¥n ここはメッセージの2行目です。

```
は、メッセージリテラルとして解釈されると
```

ログオンしているユーザー名は domain ¥ test さんです。

```
ここはメッセージの2行目です。
```

という内容になります。

### 2.3.8. 真偽リテラル

真偽を指定するために用います。デフォルト値は状況によって異なります。

yes	真
no	偽

yes, no の代わりに true, false を利用することもできます。

### 2.3.9. 特殊変数

特殊変数の使い方

特殊変数は「%(パーセント)」記号のあとに変数名をつけることで、その状況に応じた文字列に置き換わります。特殊変数は、一部の パスリテラル・実行リテラル・メッセージリテラルで利用できます。どのような特殊変数を利用できるかについては、次のページを参照 してください。

特殊変数は、「% {変数名 }」という形で記述します。変数名が「LogonUsername」の場合、「% {LogonUsername}」とすることで特殊変数 として解釈されます。



変数名が1文字の場合は、「{」および「}」を省略できます。例えば、変数名が「u」の場合、「% {u}」もしくは「% u」は同じ特殊変数 として解釈されます。

変数名が1文字の場合のみ、大文字と小文字が区別されます。つまり、「u」と「U」は別の変数名ですが、「LogonUsername」と「LogonUsernAme」は同じ変数を指すものとして解釈されます。

例えば、ログオンダイアログにおいて変数名「u」はユーザー名の欄に入力された文字列として解釈されます。「test」というユーザー 名が入力されているときは「% u」は test と置換されます。「C:¥perl¥bin¥perl.exe C:¥GINA¥admin¥auth.pl "% u"」という行は「% u」が test に置換されて、「C:¥perl¥bin¥perl.exe C:¥GINA¥admin¥auth.pl "test"」と解釈されます。



### ■ 全ての状況で利用できる特殊変数

特殊変数は、利用される場所に応じて利用できるものが決まっています。

以下には特殊変数が利用できる場所では必ず利用できる特殊変数を列挙します。別名が存在するものは、別名を利用しても参照する ことができます。

特殊変数名	別名	置換文字列
% {% }	%%	%記号
% {CurStatus}	% s	現在のステータス
% {PrevStatus}	% S	1つ前のステータス
% {CurStatusStr}		現在のステータスを表す文字列
% {PrevStatusStr}		1 つ前のステータスを表す文字列
% {LogonUsername}		利用者が logon ダイアログに入力したユーザー名 ( ログオンしていない ときは空文字列になります )
% {LogonDomain}		利用者が logon ダイアログに入力したドメイン名 ( ログオンしていない ときは空文字列になります )
% {MsUsername}		実際に端末にログオンしたユーザー名 ( ログオンしていないときは空文 字列になります )
% {MsDomain}		実際に端末にログオンしたドメイン名 ( ログオンしていないときは空文 字列になります )

ステータスについての詳細は、「2.1.6. ステータス」を参照してください。

### ■ イベント駆動スクリプトで利用できる特殊変数

3.4 で述べるイベント駆動スクリプトでは、画面に入力された値を特殊変数によって参照することができます。ウインドウ名に関しての詳細は「3.3.4. MSGINA と機能互換にするための名前」をご覧ください。

# 3. GUI マクロとイベント駆動スクリプト

この章では、GINAが表示する各種ダイアログのデザインや挙動をカスタマイズする方法を説明します。 CO-GINAでは、ダイアログの見た目をカスタマイズするために「GUIマクロ」と呼ばれる XML ファイルを利用します。また、ボタン が押されたときの挙動をカスタマイズするために「イベント駆動スクリプト」を利用します。

# 3.1. 概要

### ■ GUI マクロの概要

GUI マクロはダイアログのデザインやレイアウトを規定します。GUI マクロは XML によって記述され、1ファイルにつき1つの ダイアログのデザインを表わします。

CO-GINA ではログオンダイアログやロック通知ダイアログなどで GUI マクロを利用してダイアログを自由にカスタマイズすることが できます。以下にその一例を示します。



### ■ イベント駆動スクリプトの概要

CO-GINA では、ダイアログでボタンが押されたときやタイムアウトが発生したときに任意のスクリプトを起動することができます。 このスクリプトをイベント駆動スクリプトと呼びます。

ダイアログでボタンが押されると、ボタン名と同名のイベントが発生します。このイベントに対して「イベント駆動スクリプト」が 設定されていれば、指定されたスクリプトを起動します。

イベント駆動スクリプトにはダイアログに入力された値を変数として与えることができます。また、スクリプトは実行結果を標準出力 を通して既定の書式で「アクション」などを出力し、実行後の動作を指定することができます。

この仕組みを利用して、例えば次のような設定を行うことができます。

- ログオンダイアログでログオンをしたときの認証方法を変更する
- ログオンダイアログにボタンを追加し、ゲストでログオンするように設定する
- ボタンが押されたときに、スマートカードなどからデータを読みとって認証を行うように設定する

### ■ GUI マクロとイベント駆動スクリプトの関係

「GUIマクロ」と「イベント駆動スクリプト」が設定されているとき、設定されていないときの処理の流れを図示します。



図に示したように、ダイアログを表示するときに「GUI マクロ」が設定されていると、GUI マクロの内容に応じた画面が表示されます。 設定されていないときは MSGINA と同じダイアログを表示します。

ボタンが押されると押されたボタンに対して「イベント駆動スクリプト」が設定されているかによって処理が変化します。設定されて いるときは、イベント駆動スクリプトを実行します。設定されていないときは、MSGINAと同じの処理となるようにアクションやメッセージ を出力します。CO-GINAは、メッセージとアクションに応じた処理を行います。

# 3.1.1. カスタマイズできるダイアログの一覧

GUIマクロやイベント駆動型スクリプトを利用できるダイアログには次のようなものがあります。これらのダイアログでは「GUIマクロ」 および「イベント駆動型スクリプト」を利用できます。

ダイアログ名	表示するプラグイン	意味
notice	co-notice.dll	ログオン時に通知を行うダイアログです。 Windows へ」う; そ Comment Classical Windows や Professional Windows や Windows や
logon	co-logon.dll	認証を行うログオンダイアログです。 Windows AD94/ Convert a metal Professional 1-ザー&&@ Convert a metal 1/27-F& Convert a metal Convert a metal Conver
option	co-option.dll	ログオン中に Ctrl+Alt+Del を押したときに表示されるオプション ダイアログです。
chpwd	co-option.dll	オプションダイアログから「パスワードの変更」を選択した ときに表示されるパスワード変更ダイアログです。
locknotice	co-lock.dll	ロック中に「Ctrl+Alt+Delを押してください」と表示する locknotice ダイアログです。
locklogon	co-lock.dll	ロックを解除するためにアカウント情報を入力するロック ログオンダイアログです。

# 3.2. GUI マクロ作成の手順

GUI マクロを用いると、各種ダイアログのデザインを簡単な手順で変更することができます。 ここでは、GUI マクロを作成するための手順をまとめています。

## 3.2.1. 必要なツール

GUIマクロを作成するには次のようなツールが必要です。

- リソースを編集するアプリケーション(リソースエディタ)
   Microsoft 社製品の「Visual C++」に付属するリソースエディタや Angus Johnson 氏作成のフリーソフト「Resource Hacker」などが利用可能です。
- GUIマクロコンバータ (gui\_convert.exe)
   CO-GINA に付属しているものをご利用ください。
- XML エディタ XML 形式のファイルを編集できるエディタをご利用ください。メモ帳などのテキストエディタでもかまいません。

### 3.2.2. 手順の概要

GUIマクロを作成するには次のような3つの手順をふむことになります。以下では、それぞれの手順に関して詳細な手順を解説します。



MSGINA のダイアログをもとに GUI マクロを作成するのであれば、②の段階において msgina.dll を指定するところから始めてください。

### 3.2.3. 手順① リソースを編集する

GUIマクロ作成の1つ目の手順はリソースの編集です。この段階では表示するコントロールの種類や位置、スタイルなどを決定します。 利用できるコントロールの種類については、「3.3.1. GUIマクロで利用可能なコントロールの種類」をご覧ください。

「C:¥GINA¥Sample¥gui\_macro¥dll」を「C:¥GINA¥gui\_macro¥dll」にコピーしてください。

「Visual C++ 6.0」を用いるときは次のようにします。

「C:¥GINA¥gui\_macro¥dll¥gui\_macro\_dll.dsw」というワークスペースを「Visual C++ 6.0」で開きます。「Resource」タブを開き、 ツリーの中からダイアログの中にいくつかサンプルのダイアログがあります。メニューの挿入から「リソース」を選び、新しいダイアログ を作成してください。

追加するコントロールの種類や位置、スタイルなどを決定します。特にスタイルや位置をあとから編集することが困難ですので、 この段階で決定してください。

編集が終了したら、メニューの「ビルド」の「アクティブな構成の設定」を開き、「Release」になっているか確認してください。 (なっていない場合は、Release に変更します)

次に、「ビルド」を行います。ビルドに成功すると、「C:¥GINA¥gui\_macro¥dll¥gui\_macro\_dll.dll」に DLL が作成されます。

このようなプロジェクトを作成する方法は、MSDN の「リソースのみの DLL を作成する」もしくは「リソースのみの DLL の作成」を ご覧ください。



補足

ここでは、「Visual C++ 6.0」のリソースエディタの使い方を簡単に捕捉しておきます。詳細は Visual C++ の へルプをご覧ください。

- 新しいコントロールを追加するには、「コントロールウインドウ」から追加したいコントロールを選択して、 ダイアログの上でクリックします。(「コントロールウインドウ」が表示されていないときは、メニュー 上で右クリックして「コントロール」を選択してください)
- スタイルを変更するには、各ウインドウを選択して右クリックしてメニューから「プロパティ」を選択してください。
- タブキーを押したときに移動する順番を変更するには、メニューの「レイアウト」から「タブオーダー」
   を選択してください。
- 作成したダイアログを試しに表示してみるには、メニューの「レイアウト」から「テスト」を選択して ください。

# 3.2.4. 手順② GUI マクロへの変換

2つ目の手順は GUI マクロへの変換です。手順①で作成した DLL ファイルをもとにして GUI マクロファイルを作成します。 変換は CO-GINA 付属の「GUI マクロコンバータ」を用いて行います。「C:¥GINA¥tools¥gui\_convert.exe」を実行してください。

マクロコンバータ	×
DLLからマクロに変換します	
	参照( <u>B</u> )
(保存する名前(N):	
	参照( <u>R</u> )
変換する②	
ウィンドウハンドルからマクロに変換します。	
変換する ウィンドウハンドル(型):	-
	•
保存する名前(N):	6 m m 1
変換する( <u>C</u> )	
マクロを確認します(P)	
	参照(0)
確認する(S)	
閉じる	

- 1. 「変換する DLL」に手順①で作成した DLL へのパスを入力します。
- 2. 「保存する名前」に変換後の GUI マクロを保存するファイル名をフルパスでを入力します。
- 3. 「変換する」ボタンを押します。
- 4. リストから GUI マクロに変換するダイアログを選びます。
- 5. 「ダイアログの選択」ダイアログで OK ボタンを押すと出力されます。

₩indows ヘログオン	×
市立○△図書館	
アカウントを持っていない方は、右下の「一時利用」ボタンを押してください。	
「 ドメ1ン名型: 「	
パスワード(D): OK キャンセル シャットダウン(S)…	→時利用①
	Windows へログオン         市立○△図書館         アカウントを持っていない方は、右下の「一時利用」ボタンを押してください。         ユーザー名似:         ドメイン名位:         パスワード(P):         OK       キャンセル         シャットダウン(S)

## 3.2.5. 手順③ エディタで編集

最後の手順では、labelの設定と name の設定を行います。

手順②で作成した GUI マクロファイルを XML エディタで開いてください。次のような XML の形式になっていることが分かります。

```
<?xml version="1.0" encoding="Shift JIS"?>
<dialog>
   <name></name>
   <style>-1798831932</style>
   <exstyle>65793</exstyle>
   <label>Windows にログオンします </label>
   <position x="0" y="0" width="439" height="275"/>
   <font face="MS UI Gothic" size="9"/>
   <children>
       <bitmap>
           <name></name>
           <style>1342177294</style>
           <exstyle>4</exstyle>
           <label></label>
           <position x="0" y="0" width="440" height="275"/>
        </bitmap>
        <button>
           <name></name>
           <style>1342242816</style>
           <exstyle>4</exstyle>
           <label>閉じる</label>
           <position x="347" y="249" width="88" height="21"/>
        </button>
   </children>
</dialog>
```

children タグの下に、ダイアログ中の各コントロールが列挙されています。コントロールとは、ダイアログ中のビットマップやボタン、 ユーザー名を入力するエディットボックスを表します。例えば、ログオン画面で表示されるログオンダイアログには、次の図の赤線で 囲まれたものがコントロールです。



それぞれのコントロールを表す XML ノードのうち、label と name の値を設定する必要があります。 それぞれの設定方法については、以下の説明をご覧ください。

### ■ label の設定

GUIマクロでビットマップやアイコンを利用するときには、label タグの値をビットマップやアイコンへのパスに設定する必要があります。 パスに特殊変数を含むことはできません。

次のようにして、bitmap タグの label に表示したいビットマップのパスを指定します。

label では一部の特殊変数を利用可能です。利用できる特殊変数は「3.3.2. GUI マクロのタグ一覧」の「label で利用できる特殊変数」 をご覧ください。

また、label のかわりに「label\_dynamic」を用いると、表示するテキストや画像ファイルを動的に Web からダウンロードすることが できます。これにより、ダイアログに表示される内容を、設定ファイルや GUI マクロを編集することなく動的に変更することができます。 label\_dynamic の詳細は、「3.3.2. GUI マクロのタグ一覧」の「label\_dynamic の例」をご覧ください。

#### name の設定

GUI コンバータ変換後は、「children」タグの子要素の「name」が設定されていません。次のような3つのいずれかに当てはまるとき、「name」を設定する必要があります。

- 1. GUI マクロを利用するそれぞれのケースで、必須とされているコントロールや既定の動作を行う(ログオンダイアログで ユーザー名やパスワードを入力するエディット ボックスなど)
- 2. クリックしたときにスクリプトを実行したいボタン
- 3. 各種スクリプトから値を参照するコントロール

#### 1. 必須とされているコントロールや既定の動作を行いたいコントロール

コントロールに MSGINA と同じ機能を割り当てるための名前がダイアログの種類ごとに決まっています。これらの名前は「3.3.4.MSGINA と機能互換にするための名前」に並べています。特に、「必須」に丸がついている項目は、GUI マクロにその名前が存在しなければなりません。 存在しないときにはエラーになり、ダイアログを表示できません。なお、MSGINA で表示される文字列と同じ内容を表示させたい場合には、 名前を% { と } で囲った特殊変数を label に設定してください。

例えば、ログオンダイアログではユーザー名を表す「LogonUsername」が必須になっています。edit コントロールに LogonUsername という名前を設定するには次のようにします。ここでは、前回ログオンしたアカウント名が表示されるよう、label に「% {LogonUsername}」と設定しています。

```
<?xml version="1.0" encoding="Shift JIS"?>
<!--style, position 等のタグは省略しています -->
<dialog>
   <label>Windows ヘログオン </label>
   <children>
       <static>
           <name></name>
           <label>ユーザー名 (&amp;U):</label>
       </static>
       <edit>
           <name>LogonUsername</name>
           <label>% {LogonUsername}</label>
       </edit>
       <button>
           <name>ok</name>
           <label>ログオン</label>
       </button>
        ( 中略 )
   </children>
</dialog>
```

ボタンが押されたときにシャットダウンダイアログを表示させるためには「shutdown」という名前をつけます。「shutdown」は必須で はありませんが、ボタンのクリック時に既定のシャットダウンダイアログを表示させるためにはこの名前をつける必要があります。

(略) <button> <name>**shutdown**</name> <label> シャットダウン(&amp;S)...</label> </button> (略)

#### 2. クリックしたときにスクリプトを実行したいボタン

ボタンがクリックされたときにスクリプトが実行されるよう設定するためには、各ボタンに対して名前を設定しておく必要があります。 名前を設定するときには「3.3.3. 独自に name を設定するときの基準」に注意してください。

例えば、ログオンダイアログにゲストアカウントでログオンするボタンを追加するときには、次のようにしてボタンに名前をつける 必要があります。ここでは、「GuestLogon」という名前をつけました。

```
<?xml version="1.0" encoding="Shift_JIS"?>
<!--style, position 等のタグは省略しています -->
<dialog>
<label>Windows ヘログオン </label>
<font face="MS UI Gothic" size="9"/>
<children>
(中略)
<button>
<label> --時利用(&amp:T)</label>
</button>
</button>
</button>
</button>
</button>
```

 $\langle /dialog \rangle$ 

co-logon.ini において次のように設定すると、このボタンが押されたときにスクリプトを実行することができます。

```
[dialog]
logon=C:¥GINA¥gui_macro¥logon_macro.xml
[logon.GuestLogon]
script=C:¥perl¥bin¥perl C:¥GINA¥admin¥guestlogon.pl
timeout=10s
display=no
```

#### 3. 各種スクリプトから値を参照するコントロール

ログオンダイアログで認証をカスタマイズするときには、ユーザー名やパスワードに入力されている値をスクリプトが参照します。 このように、イベント駆動スクリプトはコントロールの値を読みとることができます。イベント駆動スクリプトにコントロールの値を 与えるためには、コントロールの名前を設定する必要があります。「3.3.3 独自に name を設定するときの基準」に注意して名前をつけて ください。

例えば、ログオンダイアログに language という入力欄を追加したときには、co-logon.ini で次のように設定すると、「language」の 値をスクリプトに標準入力から与えることができます。なお、この例では「language」の他に「LogonUsername」「LogonDomain」 「LogonPassword」も与えています。

[logon.ok]			
script=C:¥perl¥bin¥perl	C:¥GINA¥admin¥auth.pl	%{LogonUsername}	%{LogonDomain}
stdin=%{LogonPassword}¥	n%{language}		
timeout=10s			
display=no			
# 3.2.6. GUI マクロの確認手順

GUIマクロを確認するには次のような手順になります。

デザインを変更するときには、再度 DLL ファイルを編集しなければならないことに注意してください。



#### ■ GUI マクロコンバータによる表示試験

GUI マクロコンバータには、XML ファイルの表示試験を行う機能があります。「C:¥GINA¥tools¥gui\_convert.exe」を実行してください。 「マクロを確認します」の欄に確認する GUI マクロのファイルへのパスを入力し、「確認する」ボタンを押すと、GUI マクロの内容に 基づいて表示を行います。

クロコンバータ	x
- DLLからマクロに変換します 変換する DLL( <u>D</u> ):	
	参照( <u>B</u> )
保存する名前(N):	
変換する(2)	
-マクロギー####CLます(P)	#87(0)
( ) (型する(S) )	
 閉じる	

# 3.2.7. 設定ファイルに GUI マクロを指定する

CO-GINA の各種ダイアログに GUI マクロを設定する方法について説明します。

「3.1.1.カスタマイズできるダイアログ」に示したとおり、ダイアログを表示するプラグインが存在します。そのプラグインの設定 ファイルで、表示させたい GUI マクロファイルを指定します。指定の仕方は、設定ファイルの「dialog」セクションに「ダイアログ名 =GUI マクロへのパス」という形で行います。

例えば、logon ダイアログに GUI マクロ「C.¥GINA¥gui\_macro¥logon\_macro.xml」を指定する場合には、logon ダイアログは co-logon.dll が 表示するため、その設定ファイルである「co-logon.ini」に次のように記述します。

#### [dialog] logon=C:¥GINA¥gui\_macro¥logon\_macro.xml

また、GUI マクロを指定する部分はパスリテラルですので、最新版を Web からダウンロードして、管理の手間を省くことができます。 詳しくは、「2.3.5. パスリテラル」をご覧ください。

#### ■ ラベルの上書き

MSGINA のダイアログと GUI マクロのどちらを利用する場合においても、設定ファイルでラベルの文字列を上書きすることができます。 設定ファイルの「dialog」セクションに「ダイアログ名.コントロール名.label」という値を設定します。ダイアログのタイトルを変更 するには「dialog」セクションに「ダイアログ名.label」という値を設定します。

ラベルの上書きで指定できる特殊変数は「2.3.9. 特殊変数」の「全ての状況で利用できる特殊変数」にあるものと、タイムアウトまでの時間を表す「%t」「%T」です。(タイムアウトまでの時間についての詳細は「3.2.7. 設定ファイルに GUI マクロを指定する」の「タイムアウトの設定」をご覧ください)

例えば、logon ダイアログの ok ボタンのラベルを変更するには、co-logon.ini に次のように設定します。



次のようにラベルが「OK」から「ログオンする」に変更されます。



#### ■ タイムアウトの設定

MSGINAのダイアログと GUI マクロのどちらを利用する場合においても、ダイアログにタイムアウトを設定することができます。 初期設定のままではタイムアウトが発生したときには何も行いませんが、イベント駆動スクリプトを設定することでタイムアウト時に 様々な動作を行うことができます。

タイムアウトには「表示時間タイムアウト」と「無入力タイムアウト」の2種類あります。表示時間タイムアウトはダイアログが表示されてからの時間に応じたタイムアウトであり、「timeout」という名前で表します。無入力タイムアウトはキーボードやマウスの入力が最後にされてからの時間に応じたタイムアウトであり、「timeout\_noinput」という名前で表します。「無入力タイムアウト」は notice ダイアログと locknotice ダイアログでしか利用できません。タイムアウトを設定するには、設定ファイルの「dialog」セクションに「ダイアログ名.timeout」もしくは「ダイアログ名.timeout\_noinput」を設定してください。

例えば、locknotice ダイアログの表示時間タイムアウトを設定するには、co-lock.ini に次のように設定します。

[dialog]
locknotice.timeout=5m

なお、「timeout」「timeout\_noinput」のタイムアウトまでの時間は、それぞれ特殊変数「% t」「% T」でダイアログに表示することができます。 この機能により、利用者にタイムアウトまでの時間を知らせることが可能になります。ただし、利用者が入力可能なコントロール (エディットやコンボボックス)に「% t」や「% T」を設定しないでください。

タイムアウト時のイベントの記述方法については、「3.4.1. スクリプトの指定」をご覧ください。



# 3.3. GUI マクロの仕様

# 3.3.1. GUI マクロで利用可能なコントロールの種類

GUIマクロでは次のような種類コントロールを利用することができます。

種類	内容		
dialog	<ul> <li>ダイアログを表します。</li> <li>1つの GUI マクロファイルのルートノードとして記述します。</li> </ul>		
static	<ul> <li>・ 文字列を表します。色やフォントを変更することはできません。変更したいときはビットマップを利用して ください。</li> <li>・ 下線を引いて「U」のようにするには label を「&amp;U」とします。(GUI マクロでは「&amp;U」と表記する ことに注意してください)</li> </ul>		
	ユーザー名(山):		
	<ul> <li>ビットマップやアイコンをダイアログに貼り付けるときに利用します。</li> </ul>		
bitmap			
	<ul> <li>エディットボックスを表します。利用者に文字列を入力してもらうときに利用します。数字のみが入力される</li> </ul>		
edit	ようにしたり、ハスワートを入力できるよう状ゼ子にするように設定できます。 • 伏せ字にするには、Visual C++ のスタイルで「パスワード」を設定してください。		
	<ul> <li>チェックボックスを表します。チェック記号を入れたり外したりすることによって2択の選択を行います。</li> </ul>		
check	▼ダイヤルアップ接続を使用してログオンする(D)		
	<ul> <li>ラジオボタンを表します。複数の選択項目の中から、ただ1つだけを選択できます。</li> </ul>		
radio	<ul> <li>別のフォルダを開くときに新しいウィンドウを作らない(M)</li> <li>フォルダを開くたびに新しいウィンドウを作る(W)</li> </ul>		
button	<ul> <li>ボタンを表します。利用者が何らかの動作を行うときに利用します。</li> <li>「Enter」キーを押すと、「標準のボタン」として設定しているボタンが押されたものとして処理されます。</li> <li>「cancel」という名前をつけると、「ESC」キーを押したときに、キャンセルボタンが押されたものとして 扱われます。</li> </ul>		
	<u> </u>		
	<ul> <li>コンボボックスを表します。一覧から選択するときに利用します。コントロール内で文字を入力できるよう にしたり、一覧から選択することしかできないようにも設定可能です。</li> </ul>		
combobox	administrator のログオフ		

# 3.3.2. GUI マクロのタグ一覧

タグ名	意味	内容		
name	コントロールの 名前	<ul> <li>コントロールの名前です。</li> <li>どのようなときに name を設定すべきかは、「3.2.5. name の設定」をご覧ください。</li> <li>独自に名前をつけるときの基準は「3.3.3. 独自に name を設定するときの基準」を ご覧ください。</li> </ul>		
style	スタイル	<ul> <li>コントロールのスタイルを表す値です。</li> <li>この値を直接編集することは推奨されていません。</li> </ul>		
exstyle	拡張スタイル	<ul> <li>コントロールの拡張スタイルを表す値です。</li> <li>この値を直接編集することは推奨されていません。</li> </ul>		
label	ラベル	<ul> <li>コントロールのラベルを指定します。</li> <li>ラベルに指定した文字列が表示されます。コントロールの種類が bitmap のときは、 表示するビットマップへのパスを指定します。</li> <li>利用できる特殊変数については「label で利用できる特殊変数」をご覧ください。</li> </ul>		
label_ dynamic	動的ラベル	<ul> <li>コントロールのラベルを取得する URL を指定します。ダイアログを表示するたびに、 URL からダウンロードして、そこに記述されている文字列を表示します。(文字コード は Unicode のみ)</li> <li>ビットマップの時は、表示するビットマップファイルの URL を指定します。</li> <li>path 属性(任意)で、ダウンロードしたファイルを保存するパスを指定できます。 サーバーに接続できなかったときには、前回ダウンロードしたファイルを用います。 path 属性がないときは、一時ファイルとしてダウンロードされ、毎回廃棄されます。</li> <li>retry 属性(任意)で、ダウンロードに失敗したときに何回リトライを繰り返すかを 指定できます。timeout 属性(任意)で、サーバーからレスポンスがないとき、 何秒間待機するかを時間リテラルで指定します。</li> <li>利用できる特殊変数については「label で利用できる特殊変数」をご覧ください。</li> </ul>		
value	値	<ul> <li>(check と radio のみ) コントロールの選択状態を指定します。</li> <li>check の場合、チェックされているとき checked、されていないときは空文字列を設定します。radio の場合、選択されているとき selected、されていないときは空文字列を設定します。</li> </ul>		
position	コントロールの 場所とサイズ	<ul> <li>コントロールを表示する場所とサイズを指定します。x,y 属性で表示する場所、 width,height 属性で大きさを指定します。</li> <li>dialog タグの x,y 属性および bitmap タグの width,height 属性は auto に設定しなければ なりません。</li> <li>dialog タグの width,height はクライアント領域のサイズとなります。(タイトルバーや 境界線は含みません)</li> </ul>		

また、dialog タグには上記のタグに加え次のようなタグを含むことができます。

タグ名	種類	内容
font	ダイアログのフォント	<ul> <li>ダイアログで利用するフォントを指定します。</li> </ul>
Ion		• size 属性でサイズ、face 属性でフォント名を指定します。
		<ul> <li>ダイアログに含まれる子コントロールを列挙します。</li> </ul>
childrenダイアログの子コントロール・ children タグには dialog 以外のコント ・ children タグの子要素の 順番に応じ	<ul> <li>children タグには dialog 以外のコントロールを含めることができます。</li> </ul>	
		<ul> <li>children タグの子要素の 順番に応じて、ダイアログのタブオーダー</li> </ul>
		キーを押したときに移動する順番)が決められます。

GUI マクロのルート要素は dialog タグでなければなりません。

#### ■ label で利用できる特殊変数

label では次のような特殊変数を利用することができます。

種類	内容
特殊変数	ステータスやログオン中のユーザー名を表す特殊変数です。 「2.3.9. 特殊変数」の「■ 全ての状況で利用できる特殊変数」を参照してください。
タイムアウト までの時間	「% t」「% T」です。 タイムアウトまでの時間についての詳細は「3.2.7. 設定ファイルに GUI マクロを指定する」の 「タイムアウトの設定」をご覧ください。
MSGINA の値	MSGINA で表示される値を利用するときに指定します。利用したいコントロールの name を "% {" と "}" で 囲んで参照します。 利用できる name は「3.3.4. MSGINA と機能互換にするための名前」に記述されているものです。 例えば、オプション画面でのユーザーの情報を表す「userinfo」を「% {userinfo}」として記述すると、 MSGINA が現在ログオン中のユーザーに関する情報として表示する値に自動的に置き換わります。

#### ■ label\_dynamic の例

次のようにすると、ダイアログを表示するたびに notice.txt をダウンロードして、ファイルに記述されている内容を表示します。 テキストファイルは Shift JIS で記述されていなければなりません。

次のようにすると、ダイアログを表示するたびに mybmp.bmp をダウンロードして表示します。

```
<br/>
```

次のようにすると、notice.txt をダウンロードして、C.¥GINA¥notice.txt に保存して、ファイルの内容を表示します。ダウンロード できなかったときは、前回ダウンロードした内容を利用します。

> <static> <name></name> <label\_dynamic path="C:\GINA\notice.txt">http://xxxxxxx/notice.txt</label\_dynamic>

リトライの回数やタイムアウトを指定するには次のようにします。

<static> <name></name> <label\_dynamic path="C:¥GINA¥notice.txt" retry="3" timeout="10s"> http://xxxxxx/notice.txt</label\_dynamic>

#### 3.3.3. 独自に name を設定するときの基準

GUIマクロで作成したコントロールに名前をつけるときの制約について説明します。

name は半角英数字2文字以上でなければなりません。大文字小文字は区別しません。また、1つの GUI マクロ中の複数のコントロール に同じ name を設定することはできません。

また、次のような名前をつけることはできません。

- MSGINA と機能互換にするための名前
- ・ sas で始まる文字列
- timeout で始まる文字列

# 3.3.4. MSGINA と機能互換にするための名前

CO-GINA は、MSGINA 標準ダイアログのそれぞれのコントロールに対して名前を定義しています。例えば、logon ダイアログでは次のようになっています。



GUI マクロにおいて、MSGINA と同じ名前をコントロールに対して割り当てると、MSGINA 標準の機能を実現することができます。 例えば、ダイアログを表示するときに、自動的にログオン中のユーザー名の情報を表示させたり、ログオンするドメインの一覧が設定 させることができます。

以下に、各種ダイアログで利用できる名前を並べています。「必須」に丸がついている項目は、GUI マクロにその名前が存在しなければ なりません。存在しないときにはエラーになり、ダイアログを表示できません。どうしても表示したくない場合は、リソース編集時 (「3.2.3. 手順① リソースを編集する」の段階)に「可視 (WS\_VISIBLE)」のチェックをはずしてください。

	名前	働き	必須
notice ダイアログ	(なし)		
	LogonUsername	ログロオンするユーザー名を入力するエディット ボックスです。MSGINA で 前回ログオンしたユーザー名を表示するときは、ユーザー名が自動的に設定 されます。	•
	LogonPassword	ログオンするパスワードを入力するエディット ボックスです。	
2 2 <sup>10</sup> 2 -	LogonDomain	ログオンするドメインを選択するコンボボックスです。ドメイン参加時には ログオンできるドメインの一覧が自動的に表示されます。	•
logon タイアロク	ok	ログオンボタンです。認証を行い、成功するとログオンを開始します。	
	cancel	キャンセルボタンです。このボタンを押すと logon ダイアログを終了し、 notice ダイアログに戻ります。notice ダイアログを表示しない設定の場合は 不要です。	
	shutdown	シャットダウンボタンです。シャットダウンダイアログを表示します。	
	option	ドメイン入力欄を表示するかどうかを切り替えます	
	userinfo	ログオンしている利用者の情報を表示します。現在ログオンしている利用 のアカウントが自動的に設定されます。	
	date	ログオンした時間です。ログオン時間が自動的に設定されます。	
	lock	ロックボタンです。押されるとロックします。	
option ダイアログ	logoff	ログオフボタンです。押されるとログオフします。	
	shutdown	シャットダウンボタンです。シャットダウンダイアログを表示します。	
	chpwd	パスワード変更ボタンです。 chpwd ダイアログを表示します。	
	taskmgr	タスクマネージャボタンです。タスクマネージャを表示します。	
	cancel	キャンセルボタンです。option ダイアログを終了します。	
	Username	パスワードを変更するユーザー名です。現在ログオンしているユーザー名が 自動的に設定されます。	
	Domain	パスワードを変更するドメイン名です。現在のホスト名やドメイン名が自動的に 設定されます。	
	Oldpwd	古いパスワードです。	
chpwd ダイアログ	Newpwd1	新しいパスワードです。	
	Newpwd2	新しいパスワードの確認入力です。	
	backup	バックアップボタンです。	
	ok	パスワードの変更を行うボタンです。	
	cancel	パスワードの変更をせずにダイアログを終了します。	

	名前	働き	必須
locknotice ダイアログ userinfo ロックを解除できるアカウントを表示します。		ロックを解除できるアカウントを表示します。	
	userinfo	ロックを解除できるアカウントを表示します。	
	Username	ロック解除を行うアカウント名を入力するエディット ボックスです。 現在ログオンしているユーザー名が自動的に設定されます。	•
	Password	ロック解除を行うパスワードを入力するエディット ボックスです。	•
locklogon ダイアログ	Domain	ロック解除を行うドメインを選択するコンボ ボックスです。ドメインの一覧 が自動的に入力されます。	•
	DomainInfo	ドメイン入力欄につくラベルです	
	ok	ロック解除を試みるボタンです。	
	cancel	ロック解除を行うのをやめ、locknotice に移動できる場合は移動します	
	option	ドメイン入力欄を表示するかどうかを切り替えます	

# 3.4. イベント駆動スクリプト

イベント駆動スクリプトは、CO-GINAが表示する各種ダイアログでボタンが押されたときやタイムアウトが発生したときの挙動を規定します。 スクリプトのプログラミング言語は、標準入力もしくはコマンドライン引数から入力を読み取り、標準出力に結果を出力することが できるものならばどのようなでも利用可能です。

# 3.4.1. スクリプトの指定

イベント駆動スクリプトはダイアログごとに指定します。「3.1.1.カスタマイズできるダイアログ」に示したとおり、ダイアログを表示する プラグインが存在します。イベント駆動スクリプトの設定は、そのプラグインの設定ファイルに「ダイアログ名.イベント名」という セクションを作り、スクリプトのパスや標準入力などの起動方法を記述します。詳しい起動方法の設定については「3.4.2. 起動方法の設定」 をご覧ください。

発生	解説	イベント名
ボタンが押されたとき	<ul> <li>ボタンが押されたときに発生するイベントです。</li> <li>特定のアクションを行ったり、他のウインドウに入力された内容を 読みとって処理を変更したりします。</li> <li>スクリプトが設定されていないときは、MSGINAと同様の動作を行い ます。</li> </ul>	ボタンの name
ダイアログ 初期化時	<ul> <li>ダイアログが表示される直前に発生するイベントです。</li> <li>特定のアクションを行ったり、GUIマクロや画像ファイルの動的な変更 を行うことができます。</li> </ul>	sas_init
タイムアウト時	<ul> <li>ダイアログが表示されてから一定の時間が経過したときに発生する イベントです。</li> <li>タイムアウトの指定方法については「3.2.7 設定ファイルに GUI マクロ を指定する」の「タイムアウトの設定」をご覧ください。</li> </ul>	timeout もしくは timeout_noinput

イベント名は次のようなものがあります。

例えば、co-logon.dll の logon ダイアログで、「ok」ボタンにイベント駆動スクリプトを設定する押されたときに「C.¥GINA¥admin¥auth.pl」 を実行するには co-logon.ini に次のように設定します。

[logon.ok]

script=C:¥perl¥bin¥perl.exe C:¥GINA¥admin¥auth.pl

co-logon.dllの logon ダイアログでダイアログ初期化スクリプト (sas\_init)は co-logon.iniに次のように設定します。

[logon.sas\_init]

script=C:¥perl¥bin¥perl.exe C:¥GINA¥admin¥auth.pl

co-lock.dllのlocknoticeダイアログでタイムアウトを3分に設定し、タイムアウト時に「C.¥GINA¥timeout.pl」を実行するには co-lock.ini に 次のように設定します。

[dialog] locknotice.timeout=3m

[locknotice.timeout]
script=C:\u00e4perl\u00e4bin\u00e4perl.exe C:\u00e4GINA\u00e4admin\u00e4timeout.pl

# 3.4.2. 起動方法の設定

それぞれのスクリプトの起動方法として設定できる項目名には以下のようなものがあります。

(大文字と小文字は区別されません)

項目名	意味		
msgina_account	• 入力されたアカウントごとにイベント駆動スクリプトを利用するかを指定します。		
cogina_account	• logon, locklogon, chpwd ダイアログで ok ボタンが押されたときのみ使用します。		
script	<ul> <li>ボタンが押されたときに起動するスクリプトを指定します。</li> </ul>		
[実行リテラル]	<ul> <li>このスクリプトが標準エラーを出力すると、ログファイルにレベル5で出力します。</li> </ul>		
[特殊変数:(状況依存)]			
stdin	• スクリプトに標準入力として与える文字列を記述してください。		
[メッセージリテラル]	<ul> <li>メッセージリテラル、特殊変数の順に解釈されます。</li> </ul>		
[特殊変数:(状況依存)]			
display	<ul> <li>スクリプトを表示するかどうかを指定します。</li> </ul>		
[真偽リテラル]	・ デフォルトの値は no です。		
	<ul> <li>noに設定しても、アプリケーションによってはウインドウが表示されてしまうことが</li> </ul>		
	あります。		
exec_user	• 実行者の権限を指定します。system または user を指定します。system は端末に関する		
	全ての権限をもちます。userはログオンしたユーザーの実行権限をあらわします。		
	• user は option, chpwd, locknotice, locklogon どのみ利用できます。		
aatiya daaktan	・ フクリプトな System C 9。		
active_desktop	<ul> <li>ヘクリノトを起動時に表示するケスクトッフを指定します。</li> <li>「winlogon」よしくは「usor」を指定できます。</li> </ul>		
	<ul> <li>・デフォルトの値は状況依存です。</li> </ul>		
timeout	<ul> <li>スクリプトの終了を待つ時間を指定します。</li> </ul>		
[時間リテラル]	• タイムアウト時間が経過してもスクリプトが終了しないときには、スクリプトは強制終了		
	されます。		
	• デフォルトは -1( 無限時間 ) です。		
debug	• スクリプトの標準出力と標準エラーをダイアログで表示するかを決定します。yes に		
[ 真偽リテラル ]	すると表示し、スクリプトのデバッグを行いやすくなります。		
	・ デフォルトは no です。		
action	<ul> <li>ボタンが押されたときに起こすアクションを指定します。</li> </ul>		
	• script や stdin が設定されているときには、script の設定が優先されます。 ( 詳細は		
	P49 の■ action の項を参照 )		

#### ■ msgina\_account と cogina\_account

logon, locklogon, chpwd において ok ボタンが押された時、どのアカウント名のときにイベント駆動スクリプトを実行するかを指定 します。

ユーザー名の欄が msgina\_account にマッチするときは、イベント駆動スクリプトが設定されていたとしても MSGINA 標準の方法で ログオン、ロック解除、パスワード変更を行います。msgina\_account にマッチせず、cogina\_account にマッチしたときには、イベント 駆動スクリプトを実行します。(イベント駆動スクリプトが設定されていないときは、MSGINA 標準の動作を行います)msgina\_account にも cogina\_account にもマッチしないときには、ボタンを押しても何も起こりません。

複数のアカウントを区切るには、(カンマ)を使ってください。全てのアカウントにマッチさせるには\*(アスタリスク)を使ってください。

例えば、次のように設定すると、イベント駆動スクリプトの作成に失敗した場合でもユーザー名が Administrator のときだけは MSGINA標準の方法でログオン・ロック解除・パスワード変更を行うことが可能です。 [logon.ok]
msgina\_account=Administrator
cogina account=\*

以下のように設定してしまうと、イベント駆動スクリプトの作成が失敗したときには、いっさいログオン・ロック解除・パスワード変更 ができなくなってしまうので注意してください。



#### script

イベント駆動スクリプトのパスを指定します。

スクリプトを実行できない場合に表示されるメッセージの内容は、error.message で指定することができます。

error.message= スクリプトの実行に失敗しました。スクリプトに問題があるかもしれません。

error.message を指定しない場合、message が指定されていればそれを表示します。どちらも指定されていない場合には、何も表示 されません。

状況に応じて、特殊変数を用いることができます。これらの特殊変数をコマンドライン引数として与えることができます。ただし、 コマンドライン引数のエスケープ方法について十分注意してください。詳しくは、この節末尾の「イベント駆動スクリプトで利用できる 特殊変数」を参照してください。

特殊変数については、「3.4.4.特殊変数について」をご覧ください。

#### stdin

スクリプトに標準入力として与える文字列を指定します。

例えば、ログオン時に%dがドメイン名、%pがパスワードとして解釈されるため、「stdin=%d¥n%p」と設定して、ユーザー名:test、パスワード:testpasswd、ドメイン名:mydomainとしてログオンすると、スクリプトの標準入力には次のような文字列が与えられます。

#### mydomain testpasswd

パスワードが「test¥ntest」だった場合でも、メッセージリテラル、特殊変数の順に解釈されるため が標準入力として与えられます。

```
mydmain
test¥ntest
```

#### display

スクリプトを表示するかどうかを指定します。デフォルトは no です。

Perl などのコンソールアプリケーションを起動するときは「no」にすると DOS プロンプトの画面が表示されません。スクリプトが ダイアログを表示するときは「yes」を設定してください。

#### exec\_user

スクリプトを実行する権限を指定します。user もしくは system のどちらかを設定します。デフォルトは system です。system は システム権限をあらわし、端末に対するすべてのアクセス権を持ちます。user は端末にログオンしているユーザーの実行権限をあらわし ます。

メモ帳や Internet Explorer などの(ファイルを開くダイアログなどから)外部プロセスを起動できるインタラクティブなアプリケーション は絶対にシステム権限で実行しないでください。利用者が system 権限で任意のアプリケーションを起動できるようになってしまいます。

user は option, chpwd, locknotice, locklogon ダイアログでのみ利用できます。(これらの場所以外で user を用いるとエラーになり、 ログにエラーレベル 1 で出力されます)ただし、user 権限では winlogon のデスクトップにウインドウを表示できません。また、 user 権限で起動するアプリケーションやスクリプトは、「C:¥GINA¥user」もしくはその他のユーザー権限で読み込むことができる場所に 置いてください。

#### desktop

スクリプトを実行するデスクトップの名前を指定します。

例えば、次のような3つのデスクトップ(左から「Winsta0¥default」「Winsta0¥mydesktop」「Winsta0¥winlogon」)があるとします。 「Winsta0¥winlogon」が太線で囲われているのは、現在、ディスプレイに表示されているデスクトップが「Winsta0¥winlogon」であること を表します。(「Winsta0¥default」「Winsta0¥mydesktop」は裏に隠れています)



desktop を「Winsta0¥winlogon」にしてスクリプトを起動すると、次のように「Winsta0¥winlogon」にスクリプトのウインドウが 表示されます。





Winsta0¥winlogon



desktopを「Winsta0¥default」にしてスクリプトを起動すると、次のように「Winsta0¥default」にスクリプトのウインドウが表示されます。 ただし、ディディスプレイに表示されているデスクトップは「Winsta0¥winlogon」であるため、スクリプトのウインドウが表示されても 利用者には気づかれません。表示するデスクトップを変更するには、「active\_desktop」を設定してください。



desktop を「user」にしてスクリプトを起動すると、「WinstaO¥winlogon」に切り替わる前に表示されていたデスクトップにスクリプト が起動されます。例えば、「WinstaO¥mydesktop」を表示中に、Ctrl+Alt+Del キーを押して option ダイアログを表示したときには、 スクリプトは「WinstaO¥mydesktop」に起動されます。

利用者のデスクトップにプロセスを起動したい場合には user を指定するようにしてください。



#### 以下に、それぞれのダイアログで設定できる値とデフォルトの値を示します。

場所	設定できる値	デフォルト値	意味	例
option	< ウインドウステーション名¥	WinstaO¥	デスクトップ名を直接指定す	Winsta0¥winlogon
locknotice	デスクトップ名 >	winlogon	る。	
locklogon	user		直前に利用者が表示していたデ	user
_			スクトップ。	
notice	(指定できない)	Winsta0¥	Winsta0¥winlogon にしか起動	(なし)
logon		winlogon	できません。	

notice, logon ダイアログでは「desktop」を指定することはできません。

option, locknotice, locklogon ダイアログでは、「desktop」の値として「< ウインドウステーション名 >¥< デスクトップ名 >」もしくは「user」 を指定できます。「< ウインドウステーション名 >¥< デスクトップ名 >」では、「Winsta0¥default」のようにデスクトップ名を指定します。 ウインドウステーション名を省略したときには、自動的に先頭に「Winsta0¥」が付け足されます。ただし、「Winsta0¥user」を指定する ときにはウインドウステーション名は省略できません。

#### active\_desktop

active\_desktop は、スクリプト実行中にどのデスクトップを表示するかを指定します。指定できる値は winlogon か user のみです。 「winlogon」は CO-GINA が利用するデスクトップ (Winsta0¥winlogon)を表します。「user」は直前に表示されていた利用者のデスクトップ を表します。

例えば、「WinstaO¥mydesktop」を表示中に、Ctrl+Alt+Delキーを押して option ダイアログを表示したときには、「user」は「WinstaO¥mydesktop」を表します。「active\_desktop」に「user」を指定すると、スクリプトの起動中には「WinstaO¥mydesktop」が 表示されます。



スクリプトの実行が終了すると、表示するウインドウは自動的に「WinstaO¥winlogon」に戻ります。

以下に、それぞれのダイアログで設定できる値とデフォルトの値を示します。

場所	設定できる値	デフォルト値	意味	例
option	winlogon もしくは user	winlogon	user か winlogon かを指定します。	winlogon
locknotice				
locklogon				
notice	(指定できない)	winlogon	winlogon にしか表示することができ	(なし)
logon			ません。	

#### 📕 timeout

スクリプトが応答しなくなった場合に強制終了させるためのタイムアウトを指定します。

timeout を指定しないと、万が一スクリプトが応答しなくなったときに、電源を切るしかなくなります。timeout の設定があると そのような状況を回避でき、timeout の時間が経過したあとにスクリプトは強制終了され、ダイアログに戻ります。

timeout のデフォルトの値は -1 です。-1 に設定するとタイムアウトしません。

「timeout」が長すぎた場合、スクリプトが応答しなくなった場合に不必要に待機しなければならないことがあります。また短すぎると、 スクリプトを実行し終えるまでにタイムアウトしてしまう可能性があります。

タイムアウトの時間が経過すると、スクリプトは強制終了されます。

スクリプトが時間内に終了せずにタイムアウトしたときに、表示されるメッセージの内容を timeout.message で指定することができます。



timeout.message を指定しない場合、message が指定されていればそれを表示します。どちらも指定されていない場合には、何も表示 されません。

#### debug

yesを指定すると、スクリプトの標準出力と標準エラーをダイアログで表示します。同時に、標準出力はログにレベル5で、標準エラー はログにレベル5で出力されます。これにより、スクリプトのデバッグを行いやすくなります。ただし、yesにするのはスクリプトの 検証中だけにしてください。最悪の場合、パスワードなどの重要な情報がが画面やログに出力されてしまうことに留意してください。 「debug」を no にしている場合でも、標準エラーはログにレベル2で出力されます。

#### action

action を指定すると、イベント駆動スクリプトのデフォルトの処理を制御できます。 script が設定されていない場合、イベントが発生すると action に設定された action を実行します。 たとえば、以下のように設定すると、ログオン画面で OK ボタンが押された時にログオンします。(デフォルトの挙動と同じです)

[logon.ok]			
action=logon			

アクションの種類については、「3.4.5. スクリプトの出力」の「アクションの設定」をご覧ください。 action と script が両方設定されている場合、まず、script で指定されたプログラムを実行します。 [locknotice.timeout] script=C:¥perl¥bin¥perl.exe C:¥GINA¥admin¥lock-timeout.pl timeout=10m action=cancel message= 正常終了しました。 timeout.message= タイムアウトしました。

その後の処理は、スクリプトの実行結果によって異なります。

	正常終了時	タイムアウト	失敗
アクション	1. スクリプトが出力する action	1. 設定ファイルの timeout.action	1. 設定ファイルの error.action
	2. 設定ファイルの action	2. 設定ファイルの action	2. 設定ファイルの action
	3. 何も実行しない	3. 何も実行しない	3. 何も実行しない

スクリプトが正常終了した場合、実行結果としてスクリプトが出力したアクションを利用します。スクリプトがアクションを返さなかった 場合は、設定ファイルに記載された action の内容を利用します。action も設定されていないときには、特別な処理は実施しません。

プログラムがタイムアウトで指定された時間内に終了しない場合、timeout.action で指定されたアクションを行います。timeout.action が設定されていない場合、action で指定されたアクションを実行します、action も設定されていない場合は、特別な処理は実施しません。 スクリプトの実行に失敗した場合、error.action で指定されたアクションを実行します。error.action が設定さにはれていない場合、 action で指定されたアクションを実行します。action も設定されていない場合には、特別な処理は実施しません。

それぞれの場合について、設定ファイルで表示するメッセージを指定することができます。

	正常終了時	タイムアウト	失敗
アクション	1. 設定ファイルの message 2. スクリプトが出力する メッセージ (message と同じ優先度)	1. 設定ファイルの timeout.message 2. 設定ファイルの message	1. 設定ファイルの error.message 2. 設定ファイルの message

以下に示した例の場合、タイムアウト時には timeout.message を表示します。正常終了時、失敗時には message を表示します。

```
[locknotice.timeout]
script=C:¥perl¥bin¥perl.exe C:¥GINA¥admin¥lock-timeout.pl
timeout=10m
message= 終了しました。
timeout.message= タイムアウトしました。
```

#### 3.4.3. 起動方法設定の例

例えば、次のように設定すると、標準入力としてユーザー名、ドメイン名、パスワードを与えます。スクリプトが 10 秒たっても処理 を返さない場合は強制終了されます。

script=C:¥GINA¥admin¥auth.pl stdin=%u¥n%d¥n%p timeout=10s

## 3.4.4. 特殊変数について

特殊変数として利用できるのは次のような値です。

- name を設定されたウインドウの値
- タイムアウトまでの秒数(%tまたは%T)
- ・現在のステータスを表す値(「2.3.9.特殊変数」の「■全ての状況で利用できる特殊変数」で指定されているもの)

ダイアログごとにどのような名前が割り当てられているかは「3.3.4. MSGINA と機能互換にするための名前」をご覧ください。

# 3.4.5. スクリプトの出力

スクリプトは処理の結果を、標準出力を用いて CO-GINA に返します。

スクリプトの出力は INI ファイルの形式で行い、「メッセージ」と「アクション」の二つの値を指定します。項目名の大文字小文字は 区別しません。ただし、「logon」ダイアログはメッセージとアクション以外に設定可能な項目があります。詳しくは「4.2.2. logon ダイアログ特有のイベント駆動スクリプトの仕様」をご覧ください。

例えば、アクションとして none、メッセージとして「エラー」を設定するには次のように出力します。

action=none
message= エラー

このように出力するスクリプトとして、一番単純な Perl スクリプトは次のようなものでしょう。

print "action=none¥nmessage= エラー";

CO-GINA はスクリプトの出力を解釈するときに、まずメッセージを解釈して全てのメッセージを表示したあと、アクションに従って 処理を行います。この流れの図は「3.1. 概要」の「■ GUI マクロとイベント駆動スクリプトの関係」をご覧ください。

#### 📕 アクション

アクションは、スクリプトが終了したあとににどのような動作を行うかを指定します。

例えば、「action=shutdown」と出力すると、スクリプトの終了後にダイアログを終了し、シャットダウンを行います。

アクションはダイアログの種類によって使えるものと使えないものがあります。以下に、アクションの種類と、それを使えるダイアログの名前を列挙します。

アクション名	動作の内容	notice	logon(-mini)	option	chpwd	locknotice	locklogon , lock-mini
none	何も行わずダイアログを表示し続けます。action が指定されていない場合や 解釈できないときは、none が指定されたものとして扱われます。		•	•		•	•
cancel	ダイアログを終了します。						
logon	ログオンします。						
redo	ダイアログをもう一度表示します。						
chpwd	パスワード変更ダイアログを表示します。						
lock	端末をロックします。						
unlock	ロックを解除します。						
tasklist	タスクマネージャを表示します。						
logoff	端末からログオフします。						
forcelogoff	強制的にログオフを行います。						
shutdown	端末をシャットダウンします。						
poweroff	端末をシャットダウンし、自動的に電源を切断します。						
reboot	端末を再起動します。						
sleep	スリープ状態に入ります。						
sleep2	スリープ状態に入ります。(復帰するのに電源ボタンを押す必要があります)						
hibernate	ハイバネート状態に入ります。ただし、端末の設定でハイバネーションが可能 でなければなりません。	•	•	•		•	•

locknotice と locklogon ダイアログで、forcelogoffを出力したときは、利用者への確認は行われず、全てのプロセスが終了されます。また、 co-script.dll の prelogoff スクリプトも呼び出されません。(logoff スクリプトは呼び出されます)

locknotice と locklogon ダイアログで、shutdown, poweroff, reboot, sleep1, sleep2, hibernate を設定したときは、まず強制ログオフし、 その後 action で設定した処理が行われます。そのため、co-script.dll の prelogoff スクリプトは呼び出されず、co-script.dll の logoff スクリプト が呼ばれた後に、再度 shutdown スクリプトが呼び出されます。



メッセージではスクリプトが終了した後に表示するダイアログを指定します。これによってスクリプトからエラーダイアログや警告の ダイアログを表示できるようになります。複数のメッセージを順番に表示したり、また、OK ボタンを表示しないようにもできます。

以下に指定できる値を示します。

名前	值
message [メッセージリテラル]	ダイアログに表示するメッセージを指定します。デフォルトではメッセージなしです。 複数のメッセージを指定するには、message を複数列挙してください。その際、title, timeout, script, button, id, bitmap の値は、次の message までに指定された値を参照します。
title	ダイアログのタイトルを指定します。デフォルトの値は「警告」です。
timeout [時間リテラル]	ダイアログを表示する時間を指定します。デフォルトは「-1」(無限時間)です。タイムアウト を指定しないときは、ボタンを押さなければダイアログは消えません。
script [実行リテラル]	ボタンを押したときに起動するスクリプトを指定します。スクリプトの終了は待機されません。
button [真偽リテラル]	「OK」ボタンを表示するかどうかを指定します。デフォルトは「yes」です。「timeout」が「-1」 のときに button を表示しない設定にはできません。
msg-id	メッセージの ID を表す文字列を指定します。この ID は「script」で「% i」として参照する ことができます。
bitmap	アイコン名もしくは画像のパスを指定します。デフォルト値はアイコンなしです。

#### bitmap

bitmap のアイコン名には以下のものを設定できます。

画像	名前	詳細
	warning	警告のアイコン (!マーク)
(i)	information	情報のアイコン (i マーク)
?	question	質問のアイコン(?マーク)
$\mathbf{x}$	error	エラー時のアイコン (×マーク)
	application	実行ファイルのアイコン

それ以外の場合はパスとみなし、その場所にあるビットマップを表示します。ビットマップファイルがない場合や、ビットマップとして 読み取れない場合は、bitmap が設定されていなかったものとして扱われます。

以下に bitmap を設定したときの例を示します。図の左が正しく指定したとき、右が間違えていたときです。

警告 🔀
間違ってinfonationにしてしまった。
ОК
ssage= 間違って infonation にしてしまった。
e

次の例では、2つのダイアログを表示します。

message=0K ボタンを押すとメッセージは消えます。 title= これは警告です。 msg-id=id-00001
message=OK ボタンがないですが、3秒後にダイアログは消えます。 title= ボタンがない例 msg-id=id-00002 button=no timeout=3s script=perl xxx.pl %i

2つめのダイアログでは、ボタンを押すことができず、3秒経つとダイアログは自動的に閉じます。このときに、スクリプト「xxx.pl」 が起動されます。%iは msg-id の値(id-00002)を表します。

これは警告です。 🛛 🔀	赤タンがない 例	
OKボタンを押すとメッセージは消えます。 OK	OKボタンがないですが、3	杉後にダイアログは消えます。

# 3.4.6. イベント駆動スクリプトの例

各プラグインに掲載されている例をご覧ください。例えば、「4.2.3. ユーザー名とパスワードが同じだと許可する例」には logon ダイア ログにおいて認証を行うイベント駆動スクリプトの例が掲載されています。

# 4. プラグインの設定

# 4.1. co-notice.dll

co-notice.dll を利用するには、co-gina.ini において notice プラグインとして co-notice.dll を利用するように指定してください。(詳細は「2.1.2. co-gina.ini」をご覧ください)

[plugin] notice=C:¥GINA¥|ib¥co-notice.d||

```
■ 機能
```

- 案内画面のデザインを変更可能です。
- 一定時間キーボードに触れないとシャットダウン / ハイバネート / スリープするようにできます。(自動タイムアウト機能)

#### ■ 初期設定

#-

<u>#</u>-

<u>#</u>\_

初期状態では co-notice.dll の設定は、DLL と同じフォルダにある co-notice.ini で行います。co-notice.ini の初期設定は以下のようになっています。

```
# ダイアログにビットマップを表示するときは以下のように設定する
#-----
[path]
# BITMAP のパス [パスリテラル]
# 指定されていない場合は、デフォルトダイアログを表示する
#bitmap=C:¥GINA¥gui_macro¥notice.bmp
```

# ダイアログのデザインやレイアウトをカスタマイズ

```
[dialog]
# notice ダイアログをカスタマイズするときはマクロのパスを指定する
#notice=C:¥GINA¥gui_macro¥notice_macro.xml
```

```
# notice ダイアログのタイトル文字を指定する。
#notice.label=Windows へようこそ
```

```
# 無入力タイムアウトの時間 [時間リテラル]
# キーボード・マウスを触らない時間がこの時間を超過すると、イベント駆動
# スクリプト (notice.timeout_noinput) が発生する
notice.timeout_noinput=10m
```

# 無入力タイムアウトが発生したときの処理を変更するには以下を設定する

```
[notice.timeout_noinput]
```

```
# 無入力タイムアウトが発生したときのアクション
# 指定可能なのは none, cancel, shutdown, poweroff, reboot, sleep, sleep2, hibernate
```

```
# タイムアウト時に押されたときにスクリプトを実行するには Action をコメントアウトし、
# script のコメントをはずす
Action=poweroff
```

```
# 実行するスクリプト [実行リテラル][特殊変数 %u, %p, %d, %s, %S]
#script=C:¥perl¥bin¥perl.exe C:¥GINA¥admin¥timeout_noinput.pl
```

```
# 標準入力 [メッセージリテラル][特殊変数 %u, %p, %d, %s, %S]
# デフォルト値は空文字列
# メッセージリテラル、特殊変数の順に解釈されるため、パスワードなどに
# ¥ が 入っていても、誤動作することはない
#stdin=%u¥n%d¥n%p
# スクリプトの終了を待つ時間。[時間リテラル]
#-1 だと、スクリプトが終了するまで処理が帰らない。そのため
# スクリプトが途中で止まった場合は、電源を切らないといけなくなる
#timeout=-1
# スクリプトのデバッグ出力を行うか。[真偽リテラル]
# 有効にすると、標準出力や標準エラーを表示する
#debug=yes
#-
# cancel ボタンが押されたときの処理を変更するには以下を設定する
#_
# 無入力タイムアウトが発生したときのアクション
# 指定可能なのは none, cancel, shutdown, poweroff, reboot, sleep, sleep2, hibernate。
# cancel が押されたときにスクリプトを実行するには Action をコメントアウトし、
# script のコメントをはずす
#Action=cancel
# 実行するスクリプト [実行リテラル][特殊変数 %u, %p, %d, %s, %S]
#script=C:\u00e4perl\u00e4bin\u00e4perl.exe C:\u00e4GINA\u00e4admin\u00e4timeout noinput.pl
# 標準入力 [メッセージリテラル][特殊変数 %u, %p, %d, %s, %S]
# デフォルト値は空文字列
#stdin=
# スクリプトの終了を待つ時間。[時間リテラル]
#timeout=-1
# スクリプトのデバッグ出力を行うか。[真偽リテラル]
# 有効にすると、標準出力や標準エラーを表示する
#debug=yes
```

#### ■ 設定の詳細

path	
bitmap [ パスリテラル ]	notice ダイアログに、代わりに表示するビットマップファイルのパスを指定します。
dialog	
notice [パスリテラル]	notice ダイアログのデザインやレイアウトを変更するための GUI マクロを指定します。

通常の Ctrl+Alt+Del を押してくださいの画面が表示されます。以下のスクリーンショットは Windows XP の場合です。



この場合にも、自動タイムアウト機能を用いることができます。

bitmap を設定すると、ダイアログに表示する内容と画像で指定できます。bitmap のパスでビットマップを指定することで、例えば 次のようなダイアログを表示することができます。ビットマップのサイズに合わせてダイアログのサイズも変わります。また、title を 設定することで、ダイアログのタイトルに表示する文字列も変更することができます。



ビットマップはダイアログを表示するたびに読み込まれます。また、ビットマップを指定する部分はパスリテラルですので、最新版を Web からダウンロードして、管理の手間を省くことができます。詳しくは、「2.3.5. パスリテラル」をご覧ください。

GUI マクロを利用することはできません。ただし、タイムアウトイベントに対するイベント駆動スクリプトは設定できます。 「dialog」セクションの notice には GUI マクロを設定します。GUI マクロの詳しい作成方法は「3.2. GUI マクロ作成の手順」をご覧ください。 notice ダイアログ用の GUI マクロのサンプルは「C:¥GINA¥Sample¥gui\_macro」フォルダにある「notice\_macro.xml」です。これを編集して、 独自のダイアログを作成することができます。

notice ダイアログで指定しなければならない名前については「3.3.4. MSGINAと機能互換にするための名前」をご覧ください。これらの名前のうち「必須」と指定されているウインドウが存在しないときは、GUIマクロは不正なものとして処理されてダイアログを表示することはできません。このとき、ログファイルにレベル2で警告が出力されます。

# ご注意ください

Windows XP では起動直後の状態でネットワークが有効になっていない場合があります。これはネットワーク の準備と co-notice.dll の表示が同時に行われるためです。これを無効にするためには、グループポリシーで 「コンピュータの構成→管理用テンプレート→システム→ログオン」にある「コンピュータの起動およびログオン で常にネットワークを待つ」を有効にしてください。詳しくはマイクロソフトサポート技術情報 305293(英語) をご覧ください。

# 4.2. co-logon.dll

ログオン画面を表示するライブラリは、co-logon.dll と co-logon-mini.dll の 2 種類があります。co-logon.dll は GUI マクロやイベント 駆動スクリプトを利用することができます。co-logon-mini.dll ではダイアログの表示を外部の実行ファイルで行います。co-logon.dll と co-logon-mini.dll を同時に利用することはできません。co-logon-mini.dll については「4.3. co-logon-mini.dll」を参照してください。

# ご注意ください 自動ログオン機能が有効になっているときには、co-logon.dll や co-logon-mini.dll を利用するように設定されていても、標準のログオンダイアログが表示されます。これは自動ログオン機能を有効にするためです。

co-logon.dll を利用するには、co-gina.ini において logon プラグインとして co-logon.dll を利用するように指定してください。(詳細は「2.1.2. co-gina.ini」をご覧ください)

[plugin] logon=C:¥GINA¥lib¥co-logon.dll

#### ■ 機能

- ログオンダイアログのデザインやレイアウトを変更できます。
- ダイアログに追加されたボタンが押されたときに、利用者がパスワードを知らなくても特定のユーザーとしてログオンさせる ことができます。
- 認証を行うスクリプトを任意に設定できます。
- 認証時に利用者が入力したユーザー名と実際にログオンするユーザー名を異なるものにできます。
- 認証結果により、シャットダウンや再起動を行うことができます。
- 認証の可否にかかわらず、認証結果などの文字列を利用者に通知できます。

#### ■ 初期設定

初期状態では co-logon.dll の設定は、DLL と同じフォルダにある co-logon.ini で行います。co-logon.ini の初期設定は以下のように なっています。

```
# ダイアログのデザインやレイアウトをカスタマイズ
[dialog]
# ログオン ダイアログをカスタマイズするときはマクロのパスを指定する
#logon=C:\u00e4GINA\u00e4gui_macro\u00e4logon_macro.xml
# logon ダイアログで ok ボタンが押されたときの処理を変更するには以下を設定する
#-
[logon.ok]
# スクリプトに渡すかデフォルトの処理を行うかどうかを決めるアカウント名一覧。
# 大文字小文字は区別しない。複数を指定するときは, で区切る。
#*は全てのユーザーを表す。
# スクリプトを利用しないときは、以下はのように設定する。
msgina_account=*
cogina_account=
# スクリプトを利用するときは、以下のように設定する。
#msgina_account=Administrator
#cogina_account=*
```

```
# 認証を行うスクリプト [実行リテラル] [特殊変数:%u, %d, %p, %s, %S]
# 実行できないときは、エラーメッセージを表示する( ログインできない )
# デフォルト値はから文字列となり、実行できずにエラーとなる。
#script=C:\u00e4perl\u00e4bin\u00e4perl.exe C:\u00e4GINA\u00e4admin\u00e4auth.pl
# 標準入力から与える文字列 [メッセージリテラル][特殊変数:%u, %d, %p, %s, %S]
# デフォルト値は空文字列
# メッセージリテラル、特殊変数の順に解釈されるため、パスワードなどに
# ¥ が 入っていても、誤動作することはない
stdin=%u¥n%d¥n%p
# 認証スクリプトの終了を待つ時間。[時間リテラル]
#-1 だと、スクリプトが終了するまで処理が帰らない。そのため
# スクリプトが途中で止まった場合は、電源を切らないといけなくなる。
timeout=15s
# スクリプトのデバッグ出力を行うか。 [ 真偽リテラル ]
# 有効にすると、標準出力や標準エラーを表示する
#debug=yes
#---
# logon ダイアログで GuestLogon ボタンが押されたときの処理を変更するには
# 以下を設定する
#-
[logon. GuestLogon]
# GuestLogon ボタンが押されたときに実行されるスクリプト
#
                      [実行リテラル][特殊変数:%u, %d, %p, %s, %S]
#script=C:\u00efwindows\u00efsystem32\u00efccscript.exe /nologo C:\u00efGINA\u00efadmin\u00efGuestLogon.vbs
# 標準入力から与える文字列 [メッセージリテラル][特殊変数:%u, %d, %p, %s, %S]
# デフォルト値は空文字列
# メッセージリテラル、特殊変数の順に解釈されるため、パスワードなどに
# ¥ が 入っていても、誤動作することはない
stdin=
# 認証スクリプトの終了を待つ時間。[時間リテラル]
#-1 だと、スクリプトが終了するまで処理が帰らない。そのため
# スクリプトが途中で止まった場合は、電源を切らないといけなくなる。
timeout=15s
```

```
    # スクリプトのデバッグ出力を行うか。[真偽リテラル]
    # 有効にすると、標準出力や標準エラーを表示する
    #debug=yes
```

# 4.2.1. ログオンダイアログのカスタマイズ

dialog	
logon[パスリテラル]	logon ダイアログのデザインやレイアウトを変更するための GUI マクロを指定し
	ます

「dialog」セクションの logon に logon ダイアログのマクロを設定します。GUI マクロの詳しい作成方法は「3.2. GUI マクロ作成の手順」 をご覧ください。

logon ダイアログ用の GUI マクロのサンプルは「C:¥GINA¥Sample¥gui\_macro」フォルダにある「logon\_macro.xml」です。これを編集 して、独自のダイアログを作成することができます。

ログオンダイアログで指定しなければならない名前については「3.3.4. MSGINA と機能互換にするための名前」をご覧ください。 これらの名前のうち「必須」と指定されているウインドウが存在しないときは、GUI マクロは不正なものとして処理されてダイアログを 表示することはできません。このとき、ログファイルにレベル 2 で警告が出力されます。

以下に、マクロをカスタマイズした例を示します。

標準のダイアログは以下のようなものです。

Windows ∧ 🗗	りオン	
Copyright © : Microsoft Cor	1985-2001 postbori	Microsoft
ユーザー名(U): パスワード( <u>P</u> ):	administrator   	

これをカスタマイズすることで、次のようにすることができます。

Windows ∧□	ダオン	
市立(	○△図書館	
アカウントを持って	こいない方は、右下の「一時利用」ボタンを押してください。	
ユーザー名(型):		
パスワード(P):		
ドメイン名(①):	MYHOST (このコンピュータ)	
	OK キャンセル シャットダウン(S) (	→時利用①

※ただし、ドメインに参加していないときは、ドメイン名のコンボボックスは表示されません。



Windows では起動したプロセスのコマンドライン引数を知ることが比較的簡単に行えるため、イベント駆動 スクリプトのコマンドライン引数にパスワードを与えると、パスワードが外部に漏れてしまう可能性があります。 パスワードなどは認証スクリプトに標準入力として与えるのが望ましいでしょう。

# ご注意ください

イベント駆動スクリプトを用いて独自の認証体系を提供する場合は、利用者がロック解除・パスワード変更を 正しく行えるように、必ず lock プラグインおよび option プラグインの設定を行ってください。

# 4.2.2. logon ダイアログ特有のイベント駆動スクリプトの仕様

イベント駆動スクリプトの詳細は「3.4. イベント駆動スクリプト」をご覧ください。 ここでは、logon ダイアログに特有の仕様およびイベント駆動スクリプトの例を挙げます。

#### ■ 特殊変数の置き換え

特殊変数のうち、いくつかの値は簡単な別名が与えられています。別名が与えられている変数には、次のようなものがあります。

利用法	変数名	別名	置換文字列
%u	u	LogonUsername	ログオン ダイアログの「ユーザー名」に入力された値。
%р	р	LogonPassword	ログオン ダイアログの「パスワード」に入力された値。
%d	d	LogonDomain	ログオン ダイアログの「ドメイン」に入力された値。

#### ■ ユーザー情報の書き換え

logon ダイアログでは、「3.4.5. スクリプトの出力」で説明した出力の他に、アクションが logon のときはスクリプトの出力で実際に ログオンするユーザー情報を出力できます。これによって、利用者が logon ダイアログに入力したユーザー名やパスワード、ドメイン名 とは別のアカウントで端末にログオンさせることができます。

次のような項目で指定します。名前の大文字と小文字は区別されません。

名前	值
MsUsername	実際にログオンするアカウントのユーザー名を指定します。
MsPassword	実際にログオンするアカウントのパスワードを指定します。
MsDomain	実際にログオンするアカウントのドメイン名を指定します。

実際の使い方は次に示す例をご覧ください

#### ■ スクリプトの出力例

ログオンを拒絶する場合は以下のように出力してください。

#### action=none

ただし、このままだとエラーダイアログが出ないので、利用者が入力した文字をログオンスクリプトが適切でないと判断したこと、 つまりユーザー名かパスワードが適切でないことが伝わりません。そのため、次のようにするのがよいでしょう。

Action=none	
message=ログオンできません。ユーザー名とログオン先を確認して、もう一度パスワードを入力してください。パスワー	_
ドでは大文字と小文字が区別されます。	
bitmap=error	
title= ログオン メッセージ	



60

ログオンを許す場合は次のようにしてください。

#### action=logon

利用者が入力したユーザー名やパスワードとは別のアカウントでログオンさせる場合には、以下のようにします。

#### action=logon

MsUsername=test MsPassword=test この例では、ユーザー名:test、パスワード:test でログオンします。ただし、この場合は、ログオンする端末にユーザー名:test、 パスワード:test というアカウントが存在していなければなりません。

次のようにすると、ログオンが成功したときにもメッセージを表示することができます。

action=logon		
message= 個人で使えるディ	スク容量が制限に近づいています。注意してください。	
	>         >	
	個人で使えるディスク容量が制限に近づいています。注意してください。	
	ОК	

# 4.2.3. ユーザー名とパスワードが同じだと許可する例

まずは簡単なスクリプトの例として、ユーザー名とパスワードが同じだったらログオンできるような認証スクリプトを考えてみます。 この例は、実運用には適しませんが、スクリプトを理解するための例としては参考になるでしょう。 初めに認証スクリプトを利用するために、co-logon.ini を次のようにしてください。



このようにすると、Administrator 以外のユーザー名でログオンするときは、スクリプト「auth.pl」を起動するようになります。

次のスクリプトを「C:¥GINA¥admin¥auth.pl」として保存してください。(より詳細にエラーチェックを行うスクリプト が「C:¥GINA¥Sample¥admin¥auth.pl」にあります)このスクリプトは、ユーザー名とパスワードが同じ場合のみログオンを許可 します。ただし、ダイアログに入力したユーザー名とパスワードが同じアカウントがドメイン上やローカルに存在しなければ実際には ログオンできません。

```
# 読みとり
$username = <STDIN>;
$domain = <STDIN>;
$password = <STDIN>;
chomp $username;
chomp $domain;
chomp $password;
# 認証を行う
# Windows の認証(ドメインに参加している場合はドメインに)を行う
if(password_check($username, $domain, $password))
{
     print "message= パスワードが正しくないですよ。¥¥n 再入力してね。¥n";
     exit(0);
}
# ここまでくれば認証成功
print "Action=logon¥n";
# パスワードをチェックする
# ユーザー名とパスワードが等しい場合に許可する
sub password_ check {
 my ($username, $domain, $password) =@_;
 return ($username eq $password? 0 : 1);
}
```

#### 4.2.4. 独自の認証方法を使う場合の例

「4.2.3 ユーザー名とパスワードが同じだと許可する例」の password\_check 関数を適宜変更してください。

#### 4.2.5. Active Directory を使う場合の例

ここでは、認証に ActiveDirectory やローカルのアカウントを使う場合のスクリプトの例を示します。このスクリプトを単独で使用 するだけでは従来の認証と変わりはありませんが、このスクリプトを改造して、ActiveDirectory での認証後にどこかのサーバーに通知を 行うことや、利用者ごとに異なる警告メッセージを出すことが可能です。

以下のようなスクリプトを C:¥GINA¥admin¥auth\_win.pl として保存してください (より詳細にエラーチェックを行うスクリプトが 「C:¥GINA¥Sample¥admin¥auth\_win.pl」にあります)。設定ファイル co-logon.ini は「4.2.3. ユーザー名とパスワードが同じだと許可する 例」と同じです。

```
# chkpwd.exe の場所
 $chkpwd = 'C:\GINA\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\tools\
# 読みとり
$username = <STDIN>;
$domain = <STDIN>;
$password = <STDIN>;
chomp $username;
chomp $domain;
chomp $password;
# 認証を行う
# Windows の認証(ドメインに参加している場合はドメインに)を行う
if(password_check($username, $domain, $password))
 {
                  print "message= パスワードが正しくないですよ。¥¥n 再入力してね。¥n";
                  print "MsPassword=¥n";
                  exit(0);
}
# ここまでくれば認証成功
print "Action=logon¥n";
# 認証成功時にメッセージを出したいときには以下の行のコメントをはずす
#print "message= 認証成功しました ¥¥n$username ¥¥¥¥$domain さんがログオンします。¥n":
#---
# パスワードをチェックする
#----
 sub password_check {
                 my ($username, $domain, $password) =@_;
                  die $chkpwd. " doesn't exist" unless(-f $chkpwd);
                  open(CHKPWD, "|$chkpwd -s");
                  print CHKPWD "$username¥n$domain¥n$password¥n";
                  close CHKPWD or return 0;
                 return 1;
}
#-
# パスワードをチェックする
#-
sub password_check {
                 my ($username, $domain, $password) =@_;
                  die $chkpwd." doesn't exist" unless(-f $chkpwd);
                  open(CHKPWD, "|$chkpwd -s");
                  print CHKPWD "$username¥n$domain¥n$password¥n";
                  close CHKPWD or return 0;
                  return 1;
```

}

# 4.3. co-logon-mini.dll

co-logon-mini.dll を利用するには、co-gina.ini において logon プラグインとして co- logon-mini.dll を利用するように指定してください。 詳細は「2.1.2. co-gina.ini」をご覧ください。

[plugin]

logon=C:¥GINA¥|ib¥co-logon-mini.d||

#### ■ 機能

任意のプログラムをログオン画面として利用できます。

#### ■ 初期設定

初期状態では co-logon-mini.dll の設定は、DLL と同じフォルダにある co-logon-mini.ini で行います。co-logon-mini.ini の初期設定は 以下のようになっています。

```
[mini]
# ダイアログ表示を行うスクリプト [実行リテラル][特殊変数:%s, %S]
# デフォルト値はから文字列となり、実行できずにエラーとなる。
# !!! 必ず設定を行わなければならない !!!
script=C:\u00e4GINA\u00e4admin\u00e4logon-mini-script.exe
# 標準入力から与える文字列 [メッセージリテラル][特殊変数:%s, %S]
# デフォルト値は空文字列となる。
stdin=
# スクリプトを表示するかどうか。[ 真偽リテラル ]
# デフォルトは no だが、yes にすることが望ましい。
display=yes
# mini スクリプトの終了を待つ時間。[時間リテラル]
timeout=-1
# スクリプトのデバッグ出力を行うか。[真偽リテラル]
# 有効にすると、スクリプトの標準出力や標準エラーを表示する
#debug=yes
```

#### ■ 設定の詳細

スクリプトの起動方法は「mini」セクションに指定します。 このセクションでのスクリプトの起動方法の指定は、「3.4.2. 起動方法の設定」と同じですので、そちらを参照してください。

#### 4.3.1. ダイアログ表示スクリプト

スクリプトを開発するために必要な情報をまとめています。

#### ■ スクリプトの仕様

ダイアログ表示スクリプトは、下記の仕様を満たすならば、どのような言語でかかれていても問題ありません。(Perl, VB スクリプト,バイナリファイルなど)

・ コマンドライン引数や標準入力を読み取って、ユーザーインターフェースを表示後、標準出力で CO-GINA に動作を指示するもの。

#### 📕 出力

スクリプトは、認証の結果を標準出力に出力することで co-logon-mini.dll に返します。 この出力は INI ファイルの形式で行います。以下にその形式を記します。項目名の大文字小文字や出力する順番は問いません。

```
# 認証の結果 [logon|cance||shutdown|poweroff|reboot|sleep|sleep2| hibernate]
action=logon
# ダイアログ表示スクリプトに入力されたユーザー名
# 省略された場合は、MsUsername と同じ値を用いる
LogonUsername=Administrator
# ダイアログ表示スクリプトに入力されたドメイン名
# 省略された場合は、MsDomain と同じ値を用いる
LogonDomain=adminpass
# Windows にログオンさせるときのアカウント名
MsUsername=Administrator
# Windows にログオンするアカウントのパスワード
MsPassword=adminpass
# ログオンするドメイン
# ログオンするドメイン
# ログオンするドメイン
# ログオンするドメイン
# ログオンするにログオンするときは「.」(ピリオド)を用いる
MsDomain=.
```

「action」はスクリプトの結果としてどのような動作を行うかを表します。「action」に指定できる値については、「3.4.5.スクリプトの出力」の「アクションの設定」をご覧ください。

「action」で「logon」を指定した場合は、ログオンするアカウントの情報を「MsUsername」および「MsDomain」「MsPassword」と して出力しなければなりません。

また、ダイアログに実際に入力されたユーザー名とドメイン名を「LogonUsername」および「LogonDomain」として出力することが できます。

これらの「MsUsername」「MsDomain」「LogonUsername」「LogonDomain」に設定された値はログオン成功時にステータスに反映 されます。

# 4.3.2. Active Directory を使う場合の例

認証に ActiveDirectory やローカルのアカウントを使うダイアログ表示スクリプトの例を示します。

サンプルとなるスクリプトは「C:¥GINA¥Sample¥admin¥logon-mini-script.exe」にあり、次のような独自のログオンダイアログを表示 します。

Windows ヘログオン		X
ユーザー名(世):		
バスワード(P):		
ドメイン(10): [		
ОК	キャンセル	

ソースコードは C++ で書かれており、Visual C++ でビルドされています。ソースコードは「C:¥GINA¥Sample¥admin¥logon-mini-script」 フォルダにありますので、ご自由に利用ください。なお、この実行ファイルは LogonUser 関数を用いています。通常の Windows では 管理者権限を持っていなければこの関数を呼ぶことができません。そのためデバッグは管理者権限のアカウントで行うか、runas サービス を利用して行ってください。

また、実際のログオンダイアログに近いサンプルが「C:¥GINA¥Sample¥admin¥logon-mini-script2」フォルダにあります。こちらは、 logon-mini-script に比べてドメイン名のリストを読み取る機能や、シャットダウンボタンが追加されています。

# 4.4. co-option.dll

co-option.dll を利用するには、co-gina.ini において option プラグインとして co-option.dll を利用するように指定してください。詳細は「2.1.2. co-gina.ini」をご覧ください。

[plugin] logon=C:¥GINA¥lib¥co-option.dll

#### ■ 機能

- オプションダイアログのデザインやレイアウトを変更可能です。
- パスワードの変更を任意の外部スクリプトで実行できます。
- •「ロック」「パスワードの変更」などのボタンを押したときの挙動を変更できます。

#### ■ 設定

co-option.dllの設定は、DLLと同じフォルダにある co-option.ini で行います。co-option.ini の初期設定は以下のようになっています。

# # ダイアログのデザインやレイアウトをカスタマイズ #
" [dialog]
# option ダイアログをカスタマイズするときはマクロのパスを指定する
#option=C:¥G1NA¥gui_macro¥option_macro.xml
# option ダイアログのユーザー情報を変更するには以下を設定する
#option.userinfo.label=%u は %d¥¥%u としてログオンしています。
# chpwd ダイアログをカスタマイズするときはマクロのパスを指定する
#chpwd=C:¥GINA¥gui_macro¥chpwd_macro.xml
#
# option ダイアログで「パスワードを変更する」ボタンが押されたときの処理を
# 変更するには以下を設定する #
" [option.chpwd]
#「パスワードを変更する」ボタン駆動スクリプト [ 実行リテラル ] [ 特殊変数 :%u, %d, %s, %S]
# デフォルト値はから文字列となり、実行できずにエラーとなる。
#script=C:¥windows¥system32¥cscript.exe /nologo C:¥GINA¥tools¥run.vbs C:¥Program Files¥Interne Explorer¥iexplore.exe″ http://www.xxxxy.ee.in/
LAPIOI 61 + 16APIOI 6. 6A6 11LLP-//WWW. AAAAA. 60. JP/
# 標準入力から与える文字列 [メッセージリテラル]
stdin=

# スクリプトのタイムアウト [時間リテラル] timeout=10s # スクリプトのウインドウを表示するか [ 真偽リテラル ] display=no # 表示するデスクトップ desktop=default # 実行する権限 exec user=user # デバッグを行うかどうか [ 真偽リテラル ] #debug=yes #-# chpwd ダイアログで OK が押されたときの処理を変更するには以下を設定する #---[chpwd.ok] # スクリプトに渡すかデフォルトの処理を行うかどうかを決めるアカウント名一覧。 # 大文字小文字は区別しない。複数を指定するときは, で区切る。 #\*は全てのユーザーを表す。 # スクリプトを利用しないときは、以下はのように設定する。 msgina\_account=\* cogina\_account= # スクリプトを利用するときは、以下のように設定する。 #msgina\_account=Administrator #cogina\_account=\* # パスワードの変更を行うスクリプト [実行リテラル][特殊変数:%u, %d, %s, %S] # デフォルト値はから文字列となり、実行できずにエラーとなる。 #script=C:\u00e4perl\u00e4bin\u00e4perl.exe C:\u00e4GINA\u00e4admin\u00e4chpwd.pl # 標準入力から与えるもの [メッセージリテラル][特殊変数:%u, %d, %p, %P, %o, %s, %S] # デフォルト値は空文字列 # メッセージリテラル、特殊変数の順に解釈されるため、パスワードなどに # ¥ が 入っていても、誤動作することはない stdin=%u¥n%d¥n%o¥n%p¥n%P # スクリプトのタイムアウト [時間リテラル] timeout=-1 # デバッグを行うかどうか [ 真偽リテラル ] #debug=yes

設定の詳細

dialog	
option [パスリテラル]	option ダイアログのデザインやレイアウトを変更する GUI マクロを指定します。
chpwd [パスリテラル]	chpwd ダイアログのデザインやレイアウトを変更する GUI マクロを指定します。

GUI マクロを設定します。マクロの詳しい作成方法は「3.2. GUI マクロ作成の手順」をご覧ください。

option ダイアログ用の GUI マクロのサンプルは「C:¥GINA¥Sample¥gui\_macro」フォルダにある「notice\_macro.xml」です。 chpwd ダイアログ用の GUI マクロのサンプルは「C:¥GINA¥Sample¥gui\_macro」フォルダにある「chpwd\_macro.xml」です。これを 編集して、独自のダイアログを作成することができます。(編集用のツールやマクロの仕様は今後提供する予定です) option ダイアログと chpwd ダイアログで指定しなければならない名前については「3.3.4.MSGINA と機能互換にするための名前」を ご覧ください。これらの名前のうち「必須」と指定されているウインドウが存在しないときは、GUI マクロは不正なものとして処理され てダイアログを表示することはできません。このとき、ログファイルにレベル2で警告が出力されます。

# 4.4.1. option ダイアログ特有のイベント駆動スクリプトの仕様

イベント駆動スクリプトの詳細は「3.4. イベント駆動スクリプト」をご覧ください。 ここでは、option ダイアログに特有の仕様およびイベント駆動スクリプトの例を挙げます。

#### ■ 特殊変数の置き換え

特殊変数のうち、いくつかの値は簡単な別名が与えられています。別名が与えられている変数には、次のようなものがあります。

利用法	支数名         別名         置換文字列		置換文字列
% u	u	LogonUsername	logon ダイアログの「ユーザー名」に入力された値
% d	d	LogonDomain	logon ダイアログの「ドメイン」に入力された値

#### 4.4.2. chpwd ダイアログ特有のイベント駆動スクリプトの仕様

イベント駆動スクリプトの詳細は「3.4. イベント駆動スクリプト」をご覧ください。 ここでは、chpwd ダイアログに特有の仕様およびイベント駆動スクリプトの例を挙げます。

#### ■ 特殊変数の置き換え

特殊変数のうち、いくつかの値は簡単な別名が与えられています。別名が与えられている変数には、次のようなものがあります。

利用法	変数名	別名	置換文字列
% u	u	Username	chpwd ダイアログの「ユーザー名」に入力された値
% d	d	Domain	chpwd ダイアログの「ドメイン」に入力された値
% о	0	Oldpwd	chpwd ダイアログの「古いパスワード」に入力された値
% p	р	Newpwd1	chpwd ダイアログの「新しいパスワード」に入力された値
% P	Р	Newpwd2	chpwd ダイアログの「新しいパスワードの確認入力」に入力された値

# 4.4.3. chpwd ダイアログのイベント駆動スクリプトの出力例

以下の例はパスワード変更に成功したときの出力例です。

```
# アクション
Action=cancel
# ダイアログで表示させるメッセージ [メッセージリテラル]
# 改行するには ¥n を使う
# デフォルトはメッセージなし。
message= パスワードを正しく設定できました。
```

Action が cancel のときは、パスワード変更ダイアログは閉じられます。none のときは、パスワード変更ダイアログは閉じられません。 Action が cancel でも none のいずれの場合においても、message が出力されればその内容をダイアログで表示します。

# 4.4.4. パスワード変更のソリューション

co-option.dll を利用すると、パスワード変更を次のような3通りの方法で行うことができます。

- option ダイアログで利用者のデスクトップにプロセスを起動する:
   option ダイアログで「パスワードを変更する」ボタンを押したときに、option ダイアログを終了し、利用者のデスクトップに 外部プロセスを起動します。外部プロセスの例として、インターネットエクスプローラでパスワード変更を行うページを開く 利用法が考えられます。詳しい設定方法は、以下をご覧ください。
- option ダイアログの手前にパスワード変更を行うプロセスがダイアログを表示する:
   option ダイアログで「パスワードを変更する」ボタンを押したときに、option ダイアログの手前に新たにダイアログを作成する 外部プロセスを起動します。
- chpwd ダイアログのイベント駆動スクリプトを利用する:
   通常のパスワード変更ダイアログを利用してパスワードの変更を行いますが、OK ボタンを押したときのパスワード変更処理を 独自のスクリプトで実装します。独自の認証を利用する場合に有用です。

#### option ダイアログで利用者のデスクトップに外部プロセスを起動する

CO-GINA はイベント駆動スクリプトの終了を待機し、タイムアウトが経過するとスクリプトを強制終了します。そのため、script に 実行ファイル名を指定しても option ダイアログは終了しません。これを実現するには、C:¥GINA¥tools¥run.vbs を利用してください。 C:¥GINA¥tools¥run.vbs は外部プロセスを作成し、標準出力に「Action=cancel」と出力します。

次のように設定すると、option ダイアログで「パスワードを変更する」ボタンを押したときに、インターネットエクスプローラで任意 のページを開き、option ダイアログは閉じられます。

```
# option ダイアログで「パスワードを変更する」ボタンが押されたときの処理を
#変更するには以下を設定する
#---
[option.chpwd]
#「パスワードを変更する」ボタン駆動スクリプト [実行リテラル][特殊変数:%u, %d, %s, %S]
# デフォルト値はから文字列となり、実行できずにエラーとなる。
script=C:\u00e4windows\u00e4system32\u00e4cscript.exe /nologo C:\u00e4GINA\u00e4tools\u00e4run.vbs "C:\u00e4Program Files\u00e4Internet
Explorer¥iexplore.exe" http://www.xxxxx.co.jp/
#標準入力から与える文字列「メッセージリテラル]
stdin=
# スクリプトのタイムアウト [時間リテラル]
timeout=10s
# スクリプトのウインドウを表示するか [ 真偽リテラル ]
display=no
# 表示するデスクトップ
desktop=default
# 実行する権限
exec_user=user
# デバッグを行うかどうか [ 真偽リテラル ]
#debug=yes
```

#### option ダイアログの手前にパスワード変更を行うプロセスがダイアログを表示する

#\_\_\_\_

パスワード変更を行うダイアログを option ダイアログの手前に表示するには、タイムアウトを無限にして、デスクトップ「winlogon」 に起動する必要があります。

# option ダイアログで「パスワードを変更する」ボタンが押されたときの処理を #変更するには以下を設定する #--[option.chpwd] #「パスワードを変更する」ボタン駆動スクリプト [実行リテラル][特殊変数:%u, %d, %s, %S] # デフォルト値はから文字列となり、実行できずにエラーとなる。 script=C:¥GINA¥admin¥chpwd.exe #標準入力から与える文字列「メッセージリテラル] stdin= # スクリプトのタイムアウト [時間リテラル] timeout=-1 # スクリプトのウインドウを表示するか [ 真偽リテラル ] display=yes # 表示するデスクトップ desktop=winlogon # 実行する権限 exec\_user=system # デバッグを行うかどうか [ 真偽リテラル ] #debug=yes

#### ■ chpwd ダイアログのイベント駆動スクリプトを利用する

chpwd ダイアログのイベント駆動スクリプトを利用するには以下のように設定してください。

```
[chpwd.ok]
#スクリプトに渡すかデフォルトの処理を行うかどうかを決めるアカウント名一覧。
#大文字小文字は区別しない。複数を指定するときは、で区切る。
msgina_account=Administrator
cogina_account=*
#パスワードの変更を行うスクリプト [実行リテラル][特殊変数:%u, %d, %s, %S]
#デフォルト値はから文字列となり、実行できずにエラーとなる。
script=C:¥perl¥bin¥perl.exe C:¥GINA¥admin¥chpwd.pl
#標準入力から与えるもの [メッセージリテラル][特殊変数:%u, %d, %p, %P, %o, %s, %S]
#デフォルト値は空文字列
#メッセージリテラル、特殊変数の順に解釈されるため、パスワードなどに
#¥が入っていても、誤動作することはない
stdin=%u¥n%d¥n%o¥n%p¥n%P
#スクリプトのタイムアウト[時間リテラル]
timeout=-1
# デバッグを行うかどうか [ 真偽リテラル ]
#debug=yes
```

# 4.5. co-lock.dll

co-lock.dll を利用するには、co-gina.ini において lock プラグインとして co-lock.dll を利用するように指定してください。詳細は 「2.1.2. co-gina.ini」をご覧ください。

[plugin] lock=C:¥gina¥lib¥co-lock.dll

#### ■ 概要

- ロック通知ダイアログやロック解除ダイアログのデザインやレイアウトを変更できます。
- ロック状態のまま一定時間が経過した際にログオフを行えます。
- ロック解除の認証を行うためのスクリプトを任意に設定できます。

```
🛛 設定
```

co-lock.ini で次の項目を設定できます。

```
# ダイアログのデザインやレイアウトをカスタマイズ
<u>#</u>_
[dialog]
# locknotice ダイアログをカスタマイズするときはマクロのパスを指定する
#locknotice=C:\u00efGINA\u00efgui_macro\u00eflocknotice_macro.xml
# locklogon ダイアログをカスタマイズするときはマクロのパスを指定する
#locklogon=C:\GINA\gui_macro\locklogon_macro.xml
# locknotice ダイアログでタイムアウトまでの秒数。 [時間リテラル]
# デフォルトは -1。タイムアウト時の動作は locknotice.timeout セクションで設定する。
#locknotice.timeout=10m
# locknotice ダイアログのユーザー情報を変更するには以下を設定する
# Windows 2000 では以下のようにするとよい。
#locknotice.userinfo.label= このコンピューターは %u によってロックされています。¥n¥n%u か管理者のみ解除
することができます。¥n¥n あと %t 後に自動的にログオフします。
# Windows XP では以下のようにするとよい。
#locknotice.userinfo.label=%uか管理者のみ解除することができます。¥n¥nあと %t 後に自動的にログオフします。
# locklogon ダイアログのユーザー情報を変更するには以下を設定する
# Windows 2000 では以下のようにするとよい。
#locklogon.userinfo.label=このコンピューターは %u によってロックされています。
# Windows XP では以下のようにするとよい。
#locklogon.userinfo.label=%uか管理者のみ解除することができます。
#-
# locknotice ダイアログで タイムアウトしたときの処理を変更するには以下を
# 設定する
[locknotice.timeout]
# タイムアウト時のアクション。
# 指定可能なのは none, unlock, cancel, forcelogoff, shutdown, poweroff, reboot, sleep, sleep2, hibernate。
# タイムアウト時にスクリプトを実行するには Action をコメントアウトし、script
# のコメントをはずす
Action=forcelogoff
# 実行するスクリプト [実行リテラル][特殊変数 %s, %S]
#script=C:\u00e4perl\u00e4bin\u00e4per.exe C:\u00e4GINA\u00e4admin\u00e4lock-timeout.pl
```

```
# 標準入力 [メッセージリテラル][特殊変数 %s, %S]
# デフォルト値は空文字列
#stdin=
# スクリプトの終了を待つ時間。[時間リテラル]
#timeout=-1
# スクリプトのデバッグ出力を行うか。 [ 真偽リテラル ]
#debug=yes
#-
# locklogon ダイアログで OK ボタンが押されたときの処理を変更するには以下を
# 設定する
#---
[locklogon.ok]
# スクリプトに渡すかデフォルトの処理を行うかどうかを決めるアカウント名一覧。
# 大文字小文字は区別しない。複数を指定するときは, で区切る。
#*は全てのユーザーを表す。
# スクリプトを利用しないときは、以下はのように設定する。
msgina_account=*
cogina account=
# スクリプトを利用するときは、以下のように設定する。
#msgina_account=Administrator
#cogina_account=*
# 実行するスクリプト [実行リテラル][特殊変数 %u, %p, %d, %s, %S]
script=C:¥perl¥bin¥perl.exe C:¥GINA¥admin¥lock_auth.pl
# 標準入力 [メッセージリテラル][特殊変数 %u, %p, %d, %s, %S]
# デフォルト値は空文字列
# メッセージリテラル、特殊変数の順に解釈されるため、パスワードなどに
# ¥ が 入っていても、誤動作することはない
stdin=%u¥n%d¥n%p
# スクリプトの終了を待つ時間。[時間リテラル]
#-1 だと、スクリプトが終了するまで処理が帰らない。そのため
# スクリプトが途中で止まった場合は、電源を切らないといけなくなる。
timeout=-1
```

```
    # スクリプトのデバッグ出力を行うか。[真偽リテラル]
    # 有効にすると、標準出力や標準エラーを表示する
    #debug=yes
```

# ご注意ください

locknotice ダイアログは「ユーザーがログオンする前には必ず Ctrl+Alt+Del キーを押す」が有効になっていないと 機能しません。これを有効にする方法は「1.2.1. インストールの準備」の「Windows の設定変更」をご覧くださ い
設定の詳細

dialog	
locknotice [パスリテラル]	locknotice ダイアログのデザインやレイアウトを変更する GUI マクロを指定します。
locklogon [ パスリテラル ]	locklogon ダイアログのデザインやレイアウトを変更する GUI マクロを指定します。

GUIマクロを設定します。マクロの詳しい作成方法は「3.2.GUIマクロ作成の手順」をご覧ください。

locknotice ダイアログ用の GUI マクロのサンプルは「C:¥GINA¥Sample¥gui\_macro」フォルダにある「locknotice\_macro.xml」です。 locklogon ダイアログ用の GUI マクロのサンプルは「C:¥GINA¥Sample¥gui\_macro」フォルダにある「locklogon\_macro.xml」です。

locknotice ダイアログと locklogon ダイアログで指定しなければならない名前については「3.3.4. MSGINA と機能互換にするための名前」 をご覧ください。これらの名前のうち「必須」と指定されているウインドウが存在しないときは、GUI マクロは不正なものとして処理 されてダイアログを表示することはできません。このとき、ログファイルにレベル2で警告が出力されます。

# 4.5.1. locknotice ダイアログ特有のイベント駆動スクリプトの仕様

イベント駆動スクリプトの詳細は「3.4. イベント駆動スクリプト」をご覧ください。 ここでは、locknotice ダイアログに特有の仕様およびイベント駆動スクリプトの例を挙げます。

## ■ 特殊変数の置き換え

特殊変数のうち、いくつかの値は簡単な別名が与えられています。別名が与えられている変数には、次のようなものがあります。

参照するとき	変数名	本来の変数名	意味
% u	u	LogonUsername	logon ダイアログの「ユーザー名」に入力された値
% d	d	LogonDomain	logon ダイアログの「ドメイン」に入力された値

## ■ スクリプトの出力例

ロックを解除するには次のように出力します。

# アクション Action=unlock

強制ログオフする場合は次のように出力します。

# アクション Action=forcelogoff

# 4.5.2. locklogon ダイアログ特有のイベント駆動スクリプトの仕様

イベント駆動スクリプトの詳細は「3.4 イベント駆動スクリプト」をご覧ください。 ここでは、locklogon ダイアログに特有の仕様およびイベント駆動スクリプトの例を挙げます。

#### 特殊変数の置き換え

特殊変数のうち、いくつかの値は簡単な別名が与えられています。別名が与えられている変数には、次のようなものがあります。

参照するとき	変数名	本来の変数名	意味
% u	u	Username	locklogon ダイアログの「ユーザー名」に入力された値
% d	d	Domain	locklogon ダイアログの「ドメイン」に入力された値
% p	р	Password	locklogon ダイアログの「パスワード」に入力された値

## ■ スクリプトの出力例

ロックを解除するには次のように出力します。

# アクション Action=unlock

ロック解除せずに locknotice に戻るときには次のように出力します。

# アクション Action=cancel

強制ログオフする場合は次のように出力します。

# アクション Action=forcelogoff

# 4.5.3. locknotice ダイアログでのタイムアウトの設定

locknotice ダイアログで一定時間経過したときに強制ログオフするには、次のように設定します。

```
[dialog]
# locknotice ダイアログでタイムアウトまでの秒数。[時間リテラル]
# デフォルトは -1
locknotice.timeout=10m
[locknotice.timeout]
# タイムアウト時のアクション。
# タイムアウト時にスクリプトを実行するにはコメントアウトする
Action=forcelogoff
```

この例では10分経過したときに強制ログオフします。

Action を変更することで、強制シャットダウンやハイバネートすることができます。詳しくは、「3.4.5. スクリプトの出力」の 「アクションの設定」をご覧ください。

タイムアウトまでの残り時間を表示するには次のようにします。

```
# locknotice ダイアログのユーザー情報を変更するには以下を設定する
locknotice.userinfo.label=%u か管理者のみ解除することができます。¥n¥n あと %t 後に自動的にログオフします。
```

locknotice ダイアログで無入力時間が一定時間以上経過すると強制ログオフするには、次のように「無入力タイムアウト」を設定します。

```
[dialog]
# locknotice ダイアログで無入力タイムアウトまでの秒数。[時間リテラル]
# デフォルトは -1
locknotice.timeout_noinput=10m
[locknotice.timeout_noinput]
# タイムアウト時のアクション。
# タイムアウト時にスクリプトを実行するにはコメントアウトする
Action=forcelogoff
```

Action を変更することで、強制シャットダウンやハイバネートすることができます。詳しくは、「3.4.5. スクリプトの出力」の 「■ アクション」をご覧ください。

無入力タイムアウトまでの残り時間を表示するには次のようにします。

```
# locknotice ダイアログのユーザー情報を変更するには以下を設定する
locknotice.userinfo.label=%u か管理者のみ解除することができます。¥n¥n あと %T 後に自動的にログオフします。
```

# 4.6. co-lock-mini.dll

co-lock-mini.dll を利用するには、co-gina.ini において lock プラグインとして co-lock-mini.dll を利用するよう指定してください。 詳細は「2.1.2. co-gina.ini」をご覧ください。

[plugin]

lock=C:¥gina¥lib¥co-lock-mini.dll

### ■ 機能

任意のプログラムをロック解除画面として利用できます。

#### ■ 初期設定

初期状態では co-lock-mini.dll の設定は、DLL と同じフォルダにある co-lock-mini.ini で行います。co-lock-mini.ini の初期設定は以下の ようになっています。

```
[mini]
```

```
# ダイアログ表示を行うスクリプト [実行リテラル][特殊変数:%s, %S]
# デフォルト値はから文字列となり、実行できずにエラーとなる。
# !!! 必ず設定を行わなければならない !!!
script=C:¥GINA¥admin¥lock-mini-script.exe
# 標準入力から与える文字列 [メッセージリテラル][特殊変数:%s, %S]
# デフォルト値は空文字列となる。
stdin=
# スクリプトを表示するかどうか。[ 真偽リテラル]
# デフォルトは no だが、yes にすることが望ましい。
display=yes
# mini スクリプトの終了を待つ時間。[ 時間リテラル]
timeout=-1
# スクリプトのデバッグ出力を行うか。 [ 真偽リテラル]
# 有効にすると、スクリプトの標準出力や標準エラーを表示する
#debug=yes
```

## ■ 設定の詳細

スクリプトの起動方法は「mini」セクションに指定します。このセクションでのスクリプトの起動方法の指定は、「3.4.2. 起動方法の設定」 と同じですので、そちらを参照してください。

## ■ スクリプトの仕様

ダイアログ表示スクリプトは、下記の仕様を満たすならば、どのような言語でかかれていても問題ありません。(Perl, VB スクリプト,バイナリファイルなど)

・コマンドライン引数や標準入力を読み取って、ユーザーインターフェースを表示後、標準出力で CO-GINA に動作を指示するもの。

#### 📕 出力

スクリプトは、認証の結果を標準出力に出力することで co-lock-mini.dll に返します。この出力は INI ファイルの形式で行います。以下 にその形式を記します。項目名の大文字小文字や出力する順番は問いません。

# 認証の結果 action=unlock

「action」はスクリプトの結果としてどのような動作を行うかを表します。「action」に指定できる値については、「3.4.5. スクリプトの出力」 の「■アクション」をご覧ください。

# 4.7. co-script.dll

co-script.dll を利用するには、co-gina.ini において script プラグインとして co-script.dll を利用するように指定してください。 詳細は「2.1.2. co-gina.ini」をご覧ください。

[plugin] Script=C:¥gina¥lib¥co-script.dll

# ■ 機能

- 運用上重要なタイミングで管理スクリプトを実行できます。
- 複数のスクリプトが設定された際、実行順序を指定できます。
- スクリプトの実行時に画面やタスクバーに表示を出さないようにできます。
- スクリプトを管理者の権限で実行できます。

#### 🛛 設定

初期状態では co-script.dll の設定は、DLL と同じフォルダにある co-script.ini で行います。co-script.ini の初期設定は以下のように なっています。

#
# # 起動時スクリプトの設定 #
" [anhoat]
teorint1-C:YnorlYhinYnorl ava C:YGINAYadminYeorint nl %s %f
#Limeouti
#01501291-110
#
# # ログオン時スクリプトの設定
# ロッカンボバア ジンド ひひと 世
" [onlogon]
#script1=C:¥perl¥bin¥perl.exe C:¥GINA¥user¥script.pl %s
#timeout1=0s
#display1=ves
#exec user1=user
#
# ログオフ前スクリプト(ログオフを行う直前)の設定
#
[prelogoff]
#script1=C:¥windows¥svstem32¥cscript.exe /nologo C:¥GINA¥user¥logoff ask.vbs
timeout1=-1
display1=no
deskton1=default
#
" # ログオフスクリプト(ログオフした後)の設定
#
" [logoff]
teorint1-C.YnorlYbinYnorl ovo C.YGINAYadminYeorint pl
#Soripti-o-tperitpilltperi.exe o-toinataumintsoript.pi
#Limeoulii
#01Sp1ay1=no

```
# シャットダウンスクリプト(ログオフした後、シャットダウンする前)の設定
```

```
[shutdown]
```

#-

#\_

<u>#</u>–

#script1=C:\u00e4perl\u00e4bin\u00e4perl.exe C:\u00e4GINA\u00e4admin\u00e4script.pl
#timeout1=-1
#display1=no

# スクリーンセーバー前スクリプトの設定

" [screensaver] #script1=C:¥perl¥bin¥perl.exe C:¥GINA¥admin¥script.pl #timeout1=-1 #display1=no

# ■ スクリプトの種類

co-script.dll でスクリプトが起動されるタイミングには以下のようなものがあります。

onboot	• 電源投入時に呼び出されます。具体的には Ctrl+Alt+Del をおしてくださいのダイアログが表示される 直前です。
onlogon	<ul> <li>ログオン時、シェルが起動される前に呼び出されます。</li> </ul>
prelogoff	<ul> <li>ログオフ処理が行われる前に呼び出されます。</li> <li>スクリプトから標準出力の1文字目に1を出力すると、ログオフ処理を中断することができます。</li> </ul>
	<ul> <li>ブロファイルの転送が行われる前です。</li> </ul>
logoff	<ul> <li>ログオフ処理が行われた後に呼び出されます。</li> <li>通常のログオフスクリプトと同じ場所です。</li> </ul>
shutdown	<ul> <li>シャットダウン・リブート命令後のシャットダウンの歳に呼び出されます。</li> <li>通常のログオフスクリプトと同じ場所です。</li> </ul>
screensaver	<ul> <li>スクリーンセーバーが呼ばれる前に呼び出されます。</li> <li>スクリプトから標準出力の1文字目に1を出力すると、スクリーンセーバーの呼び出しを中断 することができます。</li> </ul>

# ご注意ください

Windows XP では起動直後の状態でネットワークが有効になっていない場合があります。これはネットワーク の準備と notice 画面の準備が同時に行われるためです。そのため onboot スクリプトでネットワークを利用できない ことがあります。これを無効にするためには、グループポリシーで「コンピュータの構成→管理用テンプレート →システム→ログオン」にある「コンピュータの起動およびログオンで常にネットワークを待つ」を有効にして ください。詳しくはマイクロソフトサポート技術情報 KB305293(英語)をご覧ください

# ■ 管理スクリプト実行タイミングの詳細な図

スクリプトが実行されるタイミングは「2.1.6. ステータス」の「CO-GINA 状態遷移図」を参照してください。この図のうち、ピンク色の四角の場所で「管理起動スクリプト」が起動されます。

# ■ 管理スクリプトの指定方法

設定ファイル co-script.ini にそれぞれの[] で囲った部分で、そのタイミングで起動するスクリプトを列挙します。スクリプトは1つのタイミングにつき、複数起動することができます。

それぞれのスクリプトの起動方法は script, timeout, display などのパラメータ名に起動する順番を表す番号を各パラメータの後ろに つけて指定します。例えば、次のように設定すると「onlogon」のタイミングで「mount.pl」「printer.pl」「userset.pl」の順に起動します。 [onlogon] script1=C:¥perl¥bin¥perl.exe C:¥GINA¥admin¥mount.pl timeout1=10s display1=no script2= C:¥perl¥bin¥perl.exe C:¥GINA¥admin¥printer.pl timeout2=10s display2=no script3= C:¥perl¥bin¥perl.exe C:¥GINA¥admin¥userset.pl timeout3=10s display3=no

複数のスクリプトを同時に起動するには、同じ番号の script を複数書いてください。このとき同じ番号の script は、stdin, timeout, display などその他の設定は共通の値を用います。

次のようにすると、「mount.pl」「printer.pl」を同時に起動し、2つのスクリプトの終了を待ってから、「userset.pl」を起動します。

# [onlogon] script1=C:¥perl¥bin¥perl.exe C:¥GINA¥admin¥mount.pl script1=C:¥perl¥bin¥perl.exe C:¥GINA¥admin¥printer.pl timeout1=10s display1=no

script2= C:¥perl¥bin¥perl.exe C:¥GINA¥admin¥userset.pl
timeout2=10s
display2=no

## ■ 特殊変数

特殊変数として使える値は、「2.3.9. 特殊変数」の「全ての状況で利用できる特殊変数」で指定されているものと「% f」です。 例えば、logoff と shutdown の path を次のように設定して、スクリプトの側でコマンドライン引数を解析すれば、1 つのスクリプトを いくつかのタイミングで用いることができます。

[logoff]	4.c		
Scripti-peri exit.pr	/05		
[shutdown]			
script1=perl exit.pl	%s		

%fは onbootのみ利用できる特殊変数で、前回の電源切断が正常なものだったかを知ることができます(電源投入時のステータスが「logoff」が0のとき、%fは1となります)

0	前回が正常シャットダウンの時。
1	前回が強制シャットダウンの時

#### ■ パラメータの意味

パラメータの意味については下表を参照してください。

script<番号 >	• 実行するアプリケーションのパスを指定します。
[実行リテラル]	• 実行リテラルであるため、Webから最新版のファイルをダウンロードできます。
[特殊変数:% s, % S,% f]	詳しくは「2.3.6 実行リテラル」をご覧ください。
stdin<番号 >	• 認証スクリプトに標準入力として与える文字列を記述してください。
[メッセージリテラル]	<ul> <li>メッセージリテラル、特殊変数の順に解釈されます。</li> </ul>
[特殊変数:% s, % S,% f]	

timeout<番号 > [時間リテラル]	<ul> <li>・終了を待って次のアプリケーションを起動するかどうかを指定します。デフォルトは -1(無限時間)です。</li> <li>・指定された時間だけ待ちます。指定時間内にアプリケーションが終了すると、次の アプリケーションを起動します。指定時間内に終わらなかった場合は、アプリケーション を強制終了して、次のアプリケーションを起動します。</li> <li>・1を指定すると、アプリケーションの終了まで待機します。0を指定すると待たずに 次に行きます。</li> </ul>
display< 番号 > [ 真偽リテラル ]	<ul> <li>ウインドウを表示させるかを表します。デフォルトは yes です。</li> <li>perl や cscript(WSH) などは、通常は MS-DOS プロンプトが表示さますが、no を指定 することでウインドウが表示されなくなります。</li> <li>no に設定してもアプリケーションによってはウインドウが表示されてしまうことが あります。</li> </ul>
desktop<番号>	<ul> <li>・ スクリプトを起動するデスクトップを指定します。</li> <li>・ デフォルトの値は状況依存です。</li> </ul>
active_desktop<番号> [userもしくは winlogon]	<ul> <li>スクリプトを起動時に表示するデスクトップを指定します。</li> <li>「winlogon」もしくは「user」を指定できます。</li> <li>デフォルトの値は状況依存です。</li> </ul>
exec_user<番号> [userもしくは system]	<ul> <li>実行者の権限を指定します。system は端末に関する全ての権限をもちます。user は ログオンしたユーザーの実行権限をあらわします。</li> <li>user は onlogon および prelogoff でのみ利用できます。</li> <li>デフォルトは system です。</li> </ul>
debug< 番号 > [ 真偽リテラル ]	<ul> <li>・実行するプロセスの標準出力をログに出力するかどうか。デフォルトは no です。</li> <li>・ yes にすると、レベル3でログに標準出力を出力します。</li> <li>・ no でも標準エラーはレベル2でログに出力されます。</li> </ul>

#### desktop

スクリプトを実行するデスクトップの名前を指定します。下記に示したデフォルト値以外のデスクトップに起動したいときに、desktop を適切に設定する必要があります。なお、スクリプトの起動中にデスクトップを切り替えるためには、active\_desktop を設定する必要が あります。

以下に、それぞれのダイアログで設定できる値とデフォルトの値を示します。

場所	設定できる値	デフォルト値	意味	例
onlogon	< ウインドウステーション	user	実際のデスクトップ名。	Winsta0¥
prelogoff	名¥デスクトップ名 >			winlogon
	user		onlogon では利用者のために準備され	user
			たデスクトップ、prelogoff では直前に	
			表示されていたデスクトップ。	
onboot	(指定できない)	Winsta0¥	WinstaO¥winlogon にしか起動できま	(なし)
logoff		winlogon	せん。	
shutdown				
screensaver				

onboot, logoff, shutdown, screensaver スクリプトでは「desktop」を設定することができません。

onlogon, prelogoff スクリプトでは、「desktop」の値として「< ウインドウステーション名 >¥< デスクトップ名 >」という形と「user」 という値を指定できます。

「< ウインドウステーション名 >¥< デスクトップ名 >」という形では、「WinstaO¥default」のようにデスクトップ名を指定します。 ウインドウステーション名は省略できますが、「user」というデスクトップ名のときはウインドウステーション名は省略できません。

「user」という値は、onlogon スクリプトではログオンしたユーザーのために用意されたデスクトップ(通常は WinstaO¥default)、 prelogoff では直前に利用者が表示していたデスクトップ名を表します。通常、利用者は「WinstaO¥default」というデスクトップを表示 していますが、利用者自身がデスクトップを作成した場合や、別のデスクトップを利用している特殊な場合を想定するときは、「user」を 指定してください。

## active\_desktop

active\_desktop は、スクリプト実行中にどのデスクトップを表示するかを指定します。指定できる値は winlogon か user のみです。 「winlogon」は CO-GINA が利用するデスクトップ (WinstaO¥winlogon) を表します。「user」は直前に表示されていた利用者のデスクトップ を表します。

ŝ	、てれてれのメイアロアで成とてきるがで値とアフォルトの値を示します。						
	場所	設定できる値	デフォルト値	意味	例		
	onlogon	winlogon もしくは user	user	user か winlogon かを指定します。	winlogon		
	prelogoff						
	onboot	(指定できない)	winlogon	winlogon にしか表示することができ	(なし)		
	logoff			ません。			
	shutdown						
	screensaver						

以下に、それぞれのダイアログで設定できるかを値とデフォルトの値を示します。

#### ■ 補足

- prelogoff スクリプトで利用者に確認を求める場合は、タイムアウトを設けるのが好ましでしょう。なぜなら、利用者は ダイアログに答えないまま、作業を続行することが可能になってしまうからです。スクリプトを作成する言語が、システム モーダルなメッセージボックスをサポートしているならば、それを使うのが好ましでしょう。(システムモーダルなメッセージ ボックスが表示中には、他のウインドウにフォーカスを移すことができません)
- prelogoff と screensaver では、スクリプトの中で標準出力の1文字目に1を出力すると、それぞれログオフ、スクリーン セーバーの起動を中断します。スクリプトが強制終了して標準出力になにも出力されていなかった場合には中断しません。標準 出力のチェックは、同じ番号で起動した同時実行中のスクリプトが全て終了した後に行われます。中断するときは、それ以降の 番号のスクリプトは起動しません。

# ご注意ください

screensaver スクリプトで1を出力すると次のような問題が発生してしまうため、通常は1を出力しないで ください。

- スクリプトが1を出力すると、すぐさま screensaver スクリプトが再度呼び出される。キーボード・マウス入力が行われるまで、screensaver スクリプトは呼び出され続ける。
- また、Windowsの詳細なバージョンによっては、次のような問題も発生します。
- 短時間に繰り返しスクリーンセーバーを起動しようとするとブルースクリーンとなることがある。
- ・時間のかかる処理をした後に1を出力すると、Windows が応答しなくなりブルースクリーンになる。

# 4.7.1. prelogoff スクリプトの例

■ ログオフしてよいか確認するスクリプト

ログオフしてよいか確認するスクリプトは次の通りです。

```
Set WSHShell=Wscript.CreateObject("Wscript.Shell")
Dim intMsg
intMsg = WSHShell.Popup("ログオフしてよろしいですか? ", 0, " 確認 ", vbYesNo)
if intMsg = vbNo then
Wscript.echo("1")
end if
```

このスクリプトは、C:¥GINA¥sample¥user¥logoff\_ask.vbs にも置いてあります。

co-script.ini では次のように設定します。display を no にしないと、cscript のコンソールが表示されてしまいます。

[prelogoff]
script1=cscript /NoLogo C:¥GINA¥logoff\_ask.vbs
timeout1=-1
display1=no
exec\_user1=user
timeout1=30s

これにより、ログオフ時に次のように質問されます。「はい」を押すとログオフできますが、「いいえ」を押すとログオフがキャンセル されます。

確認 🛛 🕅					
ログオフしてよろしいですか?					
<u>(北い奴)</u> いいえ(N)					

# 5. 付録

# 5.1. 過去の変更履歴

改変履歴は C:¥GINA¥doc¥changes.txt を参照してください。

CO-GINA 設定マニュアル

株式会社 CO-CONV

第12版 (Version 2.4.1.0 対応)

2009年1月

※ Microsoft,Windows は、米国 Microsoft 社の米国及びその他の国における登録商標です。

※ Windows Server は、米国 Microsoft 社の米国及びその他の国における商標です。

<sup>※</sup>その他の会社名、製品名は、各社の登録商標または商標です。