

CO-GINA

設定マニュアル

有限会社 CO-CONV

CO-GINA ドキュメント

目次

1.	CO-GINA概要	3
1.1.	CO-GINA とは	3
1.2.	インストール	4
1.2.1.	インストールの準備	4
1.2.2.	インストール作業	5
1.2.3.	インストール失敗時のトラブルシューティング	7
1.3.	万が一の場合からの復帰方法	9
2.	CO-GINA 詳細	10
2.1.	CO-GINA 構成	10
2.1.1.	co-gina.dll	10
2.1.2.	co-gina.ini	11
2.1.3.	ライセンスファイル	12
2.1.4.	ステータス	13
2.1.5.	ログ	17
2.2.	プラグイン仕様	18
2.2.1.	代表的なプラグイン	18
2.3.	設定ファイルの記法	19
2.3.1.	文字コード	19
2.3.2.	コメント・空行	19
2.3.3.	値の設定方法	19
2.3.4.	時間リテラル	20
2.3.5.	パスリテラル	20
2.3.6.	URLリテラル	21
2.3.7.	実行リテラル	21
2.3.8.	メッセージリテラル	22
2.3.9.	真偽リテラル	22
2.3.10.	特殊変数	23
3.	プラグインの設定	24
3.1.	co-notice.dll	24
3.1.1.	通常のログインダイアログ	26
3.1.2.	ビットマップをダイアログに用いる場合	26
3.2.	co-logon	27

3.2.1.	認証スクリプト	31
3.2.2.	ユーザー名とパスワードが同じだと許可する例	35
3.2.3.	独自の認証方法を使う場合の例	36
3.2.4.	Active Directory を使う場合の例	36
3.3.	co-script.dll	37
3.3.1.	prelogoffスクリプトの例	41
3.4.	co-option.dll	42
3.4.1.	パスワード変更スクリプトの仕様	45
3.5.	co-lock.dll	46
3.5.1.	ロック解除スクリプトの仕様	48
4.	付録.....	49
4.1.	過去の変更履歴	49

1. CO-GINA 概要

1.1. CO-GINA とは

■ GINA とは

GINA は **Graphical Identification aNd Authentication** の略であり、“ギナ”と発音します。GINA 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。

インストール直後の 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、通常は `system32` フォルダの中にあります。

GINA には「認証」以外に次のような機能があります。

- 認証
- 認証前の画面・ログイン作業中の画面
- パスワード変更
- スクリーンセーバー・スクリーンロック
- ログイン・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・ギナと読む）は `msgina.dll` が提供している機能を拡張するための開発用ライブラリです。

設定ファイルを何も書いていない状態(インストール直後の状態)では、`msgina.dll` と完全に同じ動作をしますが、各種設定ファイルを容易することで、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。

CO-GINA にはいくつかのプラグインが用意されており、`Ctrl+Alt+Del` を押してごさいのダイアログをカスタマイズできる、`co-notice.dll` や、ログオンダイアログをカスタマイズできる `co-logon.dll`、いろいろなタイミングでスクリプトを実行できる `co-script.dll` などがあります。

例えば、`co-logon.dll` を用いると、従来の Windows の認証だけではなく、認証の部分を置き換えることができます。これにより、独自の認証体系を使ったり UNIX 系の認証との同期を取ることが可能になります。

1.2. インストール

1.2.1. インストールの準備

■ 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。しかしながら、co-gina に付属する各種のサンプル スクリプトの多くは 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活用するためには Perl を使ってスクリプトを作成することを推奨します。

Perl を導入するには次の手順に従ってください。

- ① <http://www.activestate.com/Products/ActivePerl/> に接続し、"Download" を選択する。
- ② ActivePerl-5.6.1 を選択してダウンロードする。(zip パッケージを選択した場合には、zip の解凍ツールを導入する必要があります。MSI パッケージを選択した場合には、サービスパック等をあてて Windows の状態を最新にしておく必要がある場合があります。)
- ③ installer.bat を起動して、質問にはすべてを標準設定で展開するよう回答してください。

■ Windows の設定変更

(WindowsXP 利用者のみ)

- ① CO-GINA を使う際には、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。
- 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を利用することができません。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。

(Windows2000 利用者のみ)

- ① 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。

1.2.2. インストール作業

Administrator で作業してください。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 co-gina-YYYYMMDD.lzh を展開します。
2. 展開した際に現れる GINA というフォルダを C:¥GINA となるように移動します。
3. 別途配布されるライセンスファイルを C:¥GINA¥license.txt に置きます。

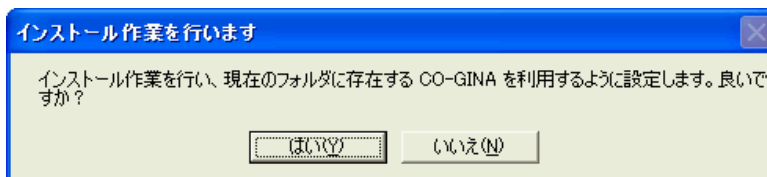
ライセンスの入手方法

- 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

4. C:¥GINA 中にある gina-install.vbs をダブルクリックして実行します。



5. 「CO-GINA をインストールしますか?」の問いに「はい」と答えてください。



6. co-gina.ini 中で必要な機能を選択し、各機能の設定ファイルを c:¥GINA に置きます。(設定ファイルのサンプルは C:¥GINA¥Sample¥*.ini にあります)
7. 一般ユーザーが C:¥GINA にアクセスできないようにします。

Windows XP Professional では次のようにします。

- C:¥GINA のプロパティ → 「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

8. 再起動してください。
9. Administrator でログインし、C:¥GINA¥gina.log が作成されているか確認します。
10. 念のため、一般ユーザーでも従来通りログインできることを確認してください。

ご注意ください

独自ログイン認証 (co-logon.dll) を有効にするときには、万一の場合においても administrator だけはログインできるようにしてください。

- co-login.ini において msgina_account=Administrator を消さないようにする。
→ 詳細は「3.2 co-logon」参照のことしてください。
- co-login.ini のタイムアウトを 0 にしないようにする。
→ 詳細は「3.2 co-logon」参照してください。

ご注意ください

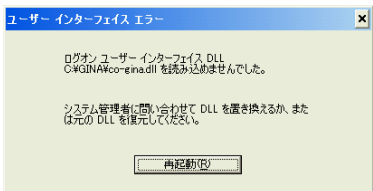
セキュリティーの観点から、c:\¥GINA にはユーザー権限でのアクセスを認めないようにしてください。

- co-script を利用してユーザー権限で起動するスクリプトは、C:\¥GINA の下には置けません。

1.2.3. インストール失敗時のトラブルシューティング

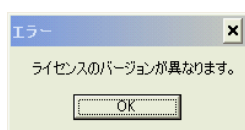
CO-GINA をインストールして再起動したときに、エラーダイアログが表示されたときには、この賞を参考にして設定を修正してください。

■ CO-GINA をインストールできていない場合

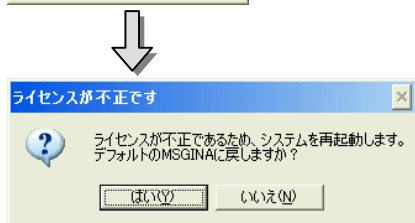
ダイアログ	解決方法
ユーザーインターフェース DLL C:\¥GINA¥co-gina.dll を読み込めませんでした。 	「C:\¥GINA¥co-gina.dll」に弊社製の co-gina.dll がおかれているか確認してください。

■ ライセンスファイルが正しく認識されない場合

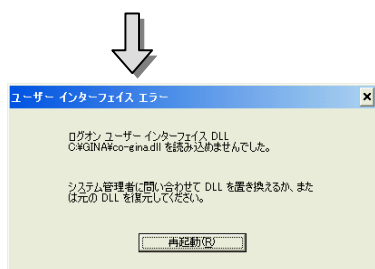
ライセンスファイルが正しく認識されないとき、次のような順番でダイアログが表示されます。



ライセンスが正しく認識されない理由が表示されます。



「はい」を押すと MSGINA に戻すことができます。この状態で復旧することは困難なので、通常は「はい」をおしてください。



co-gina.dll 読み込めなかったというエラーが出ますが、これはライセンスファイルを正しく認識できなかったためです。再起動ボタンを押して再起動してください。

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に対策方法を示します。
ご参考にしてください。

ダイアログ	解決方法
<p>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。</p>	<p>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。</p> <ul style="list-style-type: none"> ● 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？
<p>ライセンスのバージョンが異なります。</p>	<p>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。最新のバージョンのライセンスの入手方法につ いては、別途お問い合わせください。</p>
<p>ライセンスファイルの形式が不正です。 弊社から提供されたものかどうかご確 認ください。</p>	<p>ライセンスファイルの記述形式が規定のものとは異な るときにこのエラーが表示されます。</p> <ul style="list-style-type: none"> ● 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しょうか？
<p>ライセンスの有効期限が過ぎているた め、CO-GINA を利用できません。</p>	<p>ライセンスの有効期限が過ぎてしまったため、 CO-GINA を利用できません。</p> <p>→ 最新版のライセンスを入手してください。</p>
<p>ライセンスが正しくないため、 CO-GINA を利用できません。</p>	<p>ライセンスが正しくありません。</p> <ul style="list-style-type: none"> ● 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？

1.3. 万が一の場合からの復帰方法













- ログイ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`)を開く。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\GinaDll` を `msgina.dll` に変更してください。
 3. ログイ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、`gina-uninstall.vbs` を実行してください。
 3. 再起動してください。

2. CO-GINA 詳細

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ていきます。

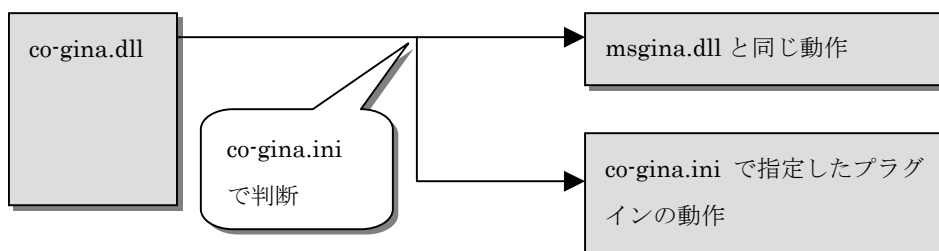
 GINA	
 doc	ドキュメントがあるフォルダ
 Sample	設定ファイルのサンプルがあるフォルダ
 Scripts	サンプルのスクリプトや実行ファイルがあるフォルダ
 co-gina.dll	CO-GINA の根幹となる DLL ファイル
 co-lock.dll	lock プラグイン
 co-logon.dll	logon プラグイン
 co-notice.dll	notice プラグイン
 co-option.dll	option プラグイン
 co-script.dll	script プラグイン
 gina-install.vbs	インストール用 VBS ファイル
 gina-uninstall.vbs	アンインストール用 VBS ファイル

2.1.1. co-gina.dll

■ 概要

co-gina.dll は Microsoft 社製の msgina.dll に代わる弊社製の GINA です。

CO-GINA をインストールした直後の ini ファイルが初期設定の状態では、msgina.dll と同じ挙動を示します。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.dll の設定は、co-gina.dll と同じフォルダにある co-gina.ini で行います。co-gina.ini は co-gina.dll の設定ファイルです。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するために用います。

■ 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。

```
[license]
# ライセンスファイルへのパスをここに記述する
# 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¥license.txt

# 起動時にスプラッシュ ウィンドウを表示するかどうか
# 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 [デフォルト:yes]
splash=yes
#splash=no

[status]
# GINA の状態を出力するファイルをここに記述する
# デフォルトでは、レジストリに状態を出力する
#path=c:\¥GINA¥status.txt

[plugin]
# 通知ダイアログを独自のものにする時には、コメントをはずす
#notice=c:\¥GINA¥co-notice.dll

# 認証を独自のものにする時には、コメントをはずす
#logon=c:\¥GINA¥co-logon.dll

# オプション ダイアログ、パスワード変更ダイアログを独自のものに
# するときには、コメントをはずす
#option=c:\¥GINA¥co-option.dll

# タイムリミットつきロックダイアログを利用するときには
# コメントをはずす
#lock=c:\¥GINA¥co-lock.dll

# スクリプト（ログオフ後、電源投入時など）を利用するときには
# コメントをはずす
#script=c:\¥GINA¥co-script.dll

[log]
# ログを出力するファイル名を指定する
# 指定がない場合は、DLL と同じ場所の co-gina.log
path=c:\¥GINA¥gina.log

# ログの詳しさ。
# 0: 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
level=3
```

■ プラグインの設定

co-gina.ini において、どのプラグインを呼び出すかどうかはを選択することができます。プラグインの1つに logon プラグインというものがありますが、これを変更する方法を例にとって説明します。

まず、初期設定においては次のようになっています。

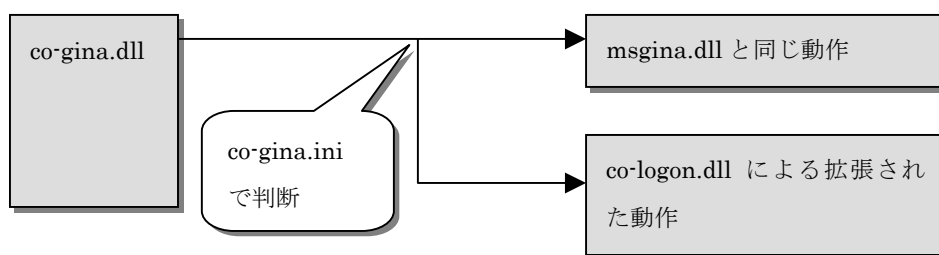
```
[plugin]
logon=
```

これは、logon プラグインが設定されていない状態を表し、通常の Windows と同じ動作になります。

弊社が提供している logon プラグイン「co-logon.dll」を用いる場合は次のようにします。

```
[plugin]
logon=c:\¥gina¥co-logon.dll
```

このようにすれば、logon プラグインとして co-logon.dll が用いられます。co-logon.dll の設定は co-logon.dll と同じフォルダにある co-logon.ini です。



プラグインにはいくつかの種類がありますが、これは 2.2 プラグイン仕様でリストアップします。

2.1.3. 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。

ライセンスファイルが正しくない場合や試用期限が過ぎた場合は、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。

2.1.4. 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

co-gina.ini の[status]にある path を設定しなかった場合、ステータス情報はレジストリに記録されます。

```
[status]
# GINA の状態を出力するファイルをここに記述する
# デフォルトでは、レジストリに状態を出力する
#path=c:¥GINA¥status.txt
```

出力先は「HKEY_LOCALMACHINE¥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。

co-gina.ini において status を設定した場合は、指定したファイルにステータス情報を INI ファイルの形式で出力します。この場合は、レジストリの情報は出力されません。

```
[status]
# GINA の状態を出力するファイルをここに記述する
# デフォルトでは、レジストリに状態を出力する
path=c:¥GINA¥status.txt
```

出力先を変更を有効にするためには再起動してください。

■ 出力データ

ステータスは次のような値が出力されます。

状態 情報	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。
	logoff	現在ログオフしているかどうか。 0 がログオフした状態で、1 がログオンした状態を表します。 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。
	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 カ ウ ン ト 情 報	UserName	利用者がログオン ダイアログに入力したユーザー名。
	Domain	利用者がログオン ダイアログに入力したドメイン。
	LogonUserName	利用者が実際に Windows にログオンしたときのユーザー名。
	LogonDomain	利用者が実際に 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。

電源起動時に `script` プラグインの電源投入時スクリプトを呼び終わるまでは `logoff` の情報は保持されません。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。この情報は特に、`script` プラグインの電源投入時スクリプトで「前回のログオフが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ることができます。

アカウント情報はログオフすると初期化されます。

■ 出力の詳細

ステータスには以下のような数字が用いられます。

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)
	111	auth_script (認証スクリプト実行時)
ログオン中	200	onlogon (co-script のログオン スクリプトを実行中)
	201	loggedon (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)
	211	chpwd_script (パスワード変更スクリプト実行中)
	220	locknotice (ロック通知ウインドウを表示中)
	221	lock (ロック ウインドウを表示中)
	222	unlock_script (ロック解除スクリプトを実行中)
	230	prelogoff (co-script のロック前 スクリプトを実行中)
ログオフへの移行	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)
	302	shutdown (co-script のシャットダウン スクリプトを実行中。パワーお富士)
	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)

ただし、スクリーンセーバーを表示しているときの PrevStatus は 400 (screensaver) となりますが、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。これは、GINA がスクリーンセーバーの表示が終了した後にはどのような状態に移行するかを知ることができないためです。

■ 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
```

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
```

2.1.5. ログ

ログを出力する場所を `co-gina.ini` の `path` で指定してください。指定がない場合は DLL と同じディレクトリの `co-gina.log` に出力します。

また、ログには詳しきのレベルがあります。レベルは0～4まであり、

0	ログをはかない
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

となっております。CO-GINA の設定を行うときは レベル3 で確認し、運用開始時にはレベル2にするのが望ましいでしょう。

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

補足

`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 を利用中に `path` を変更した場合、`co-gina.dll` は、再起動されて初めて再読み込みさ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読み込まれるので、直ちにログの場所が変わることになります。これは大きな混乱を招いてしまうため、`co-gina.ini` の `level` や `path` を編集したときは、再起動を行うことを推奨しています。

2.2. プラグイン仕様

CO-GINA は、プラグインは、DLL の形で提供されています。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	ログイ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
lock	ロックの解除ダイアログ。
script	設定ファイルに記述された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。このマニュアルの次章では、それぞれのプラグインの使い方を解説を行います。

co-notice.dll	<ul style="list-style-type: none">● 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。● 一定時間キーボードを触らないと、電源切断を行う。
co-logon.dll	<ul style="list-style-type: none">● ログインダイアログの表示およびカスタマイズ。● 特殊変数つきで外部の認証スクリプトの呼び出し。● スクリプトからの任意のエラーメッセージを返す。● スクリプトからどのようなユーザーでログインするかを設定する。
co-option.dll	<ul style="list-style-type: none">● ボタンの配置、有効・無効の設定を行える。● 各ボタンを押したときに、どの権限で何を実行するかを設定できる。
co-lock.dll	<ul style="list-style-type: none">● タイムリミットつきロックダイアログの表示。
co-script.dll	<ul style="list-style-type: none">● 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。

各プラグインは、DLL と同じ場所にある プラグイン名と同じ INI ファイルを読み取ります。プラグインのカスタマイズは、この設定ファイルを編集することで行います。

ご注意ください

弊社製のプラグインのうち、co-logon.dll 以外のプラグインを導入する場合は、必ず co-logon.dll とともに導入してください。そうしなければ、意図どおりの動作を行わない場合があります。なお、co-logon.dll は初期設定では MSGINA と同じ動作を行うので問題はありません。

2.3. 設定ファイルの記法

設定ファイルは `co-gina.ini` と各種プラグインの設定ファイルで使われます。ここでは、これらの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` にしてください。改行コードは `CRLF` にしてください。メモ帳で編集する限りは、`Shift JIS`・`CRLF` で記述することができますが、その他のエディタを使用する場合は注意が必要です。

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できるためのコメントであり、意味をなしません。空行も同様に無視されます。

```
# これはコメント
# すぐ上に空行があるが、これも意味をなさない
```

2.3.3. 値の設定方法

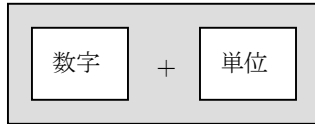
値を設定するには次のようにします。

```
[フィールド名]
設定項目=値
```

設定ファイルごとに、どのフィールド・項目にどのような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、「3. 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。



数字の部分は整数で記述します。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ます。

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。

```
# 10 ミリ秒
10ms

# 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 s
aaa
```

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。パスの中にスペースが含まれていても問題ありません。デフォルト値の解釈され方は用いられる場所に依存します。

以下に例を挙げます。

```
C:\Program Files\Internet Explorer\iexplore.exe
C:\windows\system32\eula.txt
```

2.3.6. URL リテラル

パスの代わりに Web サーバー上のファイルを指定することができます。Web サーバー上のファイルをダウンロードできた場合は、テンポラリ フォルダにある一時ファイルへのパスに変換されます。URL リテラルはパスリテラルの特別な場合であり、ローカルのパスを指定することも可能です。

以下に例を示します。単に URL を指定する場合は次のようにします。一時ファイルはスクリプトが動的に呼び出されている間は存在することが保証されています。ダウンロードできなかった場合は、空文字列が指定されていないものと解釈されます。

```
HTTP(url=http://xxxxxx/xxxx.bmp)
```

URL の他に属性を用いることができます。指定できるのは `timeout`、`alt` です。

`timeout` はダウンロードするまでのタイムアウト時間であり、この時間がたってもダウンロードが終了しない場合は、ダウンロードに失敗します。タイムアウトの時間は、時間リテラルと同じ記法を用います。指定されていないときは `-1`(無限時間)が指定されたものとして解釈されます。無限時間を指定しても、Web サーバー上にファイルがないときにはダウンロードは失敗します。

`alt` はダウンロードに失敗したときに、代わりに指定するローカルのパスです。

```
HTTP(url=http://xxxxxx/xxxx.bmp, alt=C:¥xxxxx.bmp)
HTTP(url=http://xxxxxx/xxxx.bmp, timeout=5s,)
HTTP(url=http://xxxxxx/xxxx.bmp, timeout=5s, alt=C:¥xxxxx.bmp)
```

補足

- URL リテラルでは `url` 属性は必ず指定する必要があります。 `url` 属性のみを指定するときは、`url=` を省略して `HTTP(http://xxxxxx/xxx.bmp)` のように書くことができます。
- `url`、`timeout`、`alt` などの属性はカンマ (,) で区切ります。カンマは特別な区切り文字として解釈されるため、`url` や `timeout`、`alt` の値に含めることはできません。`url` にカンマが含まれている場合は、`%2d` と HEX エンコードしてください。

2.3.7. 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るが、明記されていない場合は何も実行しない。

```
notepad.exe
notepad.exe c:¥hoge.txt
notepad.exe "c:¥Documents and settings¥hoge.txt"
"C:¥Program Files¥Internet Explorer¥iexplore.exe" http://hoge.jp/
```

2.3.8. メッセージリテラル

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 test¥¥domain さんです。¥n 以上です。
```

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 test¥domain さんです。  
以上です。
```

という内容になります。

2.3.9. 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ります。

yes	真。
no	偽。

2.3.10. 特殊変数

特殊変数を指定できる設定項目の文字列は、解釈前に次のように文字列が置き換えられます。

例えば、%u が ユーザー名 として解釈されます。test というユーザー名でログインした場合は、%u は test と置換されます。

以下に、特殊変数の一覧を示します。いろいろな特殊変数がありますが、状況によって使えるものと使えないものがあります。

状態	意味	logon	script		option		lock
		auth_script	onboot	onboot 以外	message	chpwd_script	unlock_script
%%	%記号	○	○	○	○	○	○
%u	ユーザー名	○			○	○	○
%p	パスワード	○				○	○
%d	ドメイン名	○			○	○	○
%t	時間				○		
%P	入力しなおしのパスワード					○	
%o	古いパスワード					○	
%s	現在のステータス。	○	○	○		○	○
%S	1つ前のステータス。	○	○	○		○	○
%f	強制終了かどうか。		○				

ステータスの番号は、2.1.4 ステータス を参照してください。

複数のコマンドライン引数を渡す場合、特殊変数の中で変換後に空白を含む可能性のあるかどうか注意してください。このようなことを回避するためには、ダブルクォーテーション（"）で囲んで"%u" というようにするとよいでしょう。

補足

Windows の標準ではユーザー名にダブルクォーテーション(")を含めるとはできないため、コマンドライン引数を "%u" として渡すことが可能です。しかし、利用者がログオンダイアログのユーザー名にダブルクォーテーションを含んだユーザー名を入力すると、%u にダブルクォーテーションが含まれてしまいます。このとき、コマンドラインの解釈が予期せぬものになってしまいますが、意図しないプロセスが起動されてしまうなどの問題は起こりません。

ただし、コマンドライン引数のエスケープは複雑なため、標準入力経由でパラメータを渡すことを推奨しています。

3. プラグインの設定

3.1. co-notice.dll

■ 機能

- 「Ctrl+Alt+Del を押してください」の画面に任意のビットマップ ファイルを表示できます。
- 一定時間キーボードを触らないと、電源切断を行うことができます。

■ 初期設定

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。

```
[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=

# タイトル文字。
# 指定されていない場合は、"Windows へようこそ" となる。
title=

[timeout]
# 自動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、
# exec で指定した動作をする
time=-1

# タイムアウト時の動作
# 0: なにもしない (デフォルト)
# 1: シャットダウン
# 2: 電源切断
# 3: リブート
# 4: ハイバネーション
exec=0
```

■ 設定の詳細

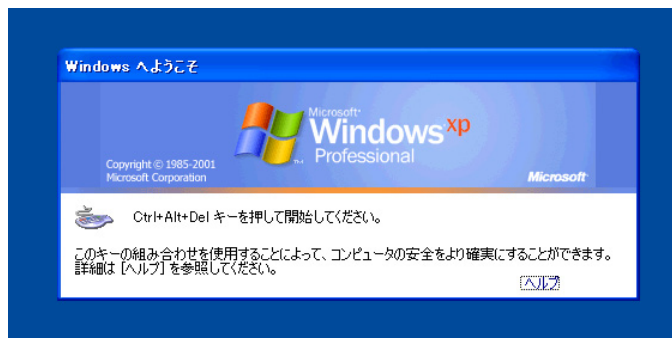
path											
bitmap [URL リテラル]	<ul style="list-style-type: none"> ● 「Ctrl+Alt+Del を押してください」の画面に、代わりに表示するビットマップファイルのパスを指定します。 ● 指定されていない場合は、デフォルトのダイアログを表示します。 										
title	<ul style="list-style-type: none"> ● ダイアログのタイトル文字列を指定します。 ● 指定されていない場合は「Windows へようこそ」というタイトルになります。 										
timeout											
time [時間リテラル]	<ul style="list-style-type: none"> ● 自動タイムアウトまでの時間を指定します。 ● 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、execで指定した動作をします。 ● [timeout] の time が -1 か 0 の場合は、自動タイムアウトがおりません。 										
exec	<ul style="list-style-type: none"> ● タイムアウト時にどのような処理をするかを指定します。 <table border="1" data-bbox="684 987 1217 1283"> <tbody> <tr> <td>0</td> <td>何もしない [デフォルト]</td> </tr> <tr> <td>1</td> <td>シャットダウン (電源切断を行うことはない。)</td> </tr> <tr> <td>2</td> <td>電源切断。</td> </tr> <tr> <td>3</td> <td>リブート</td> </tr> <tr> <td>4</td> <td>ハイバネーション</td> </tr> </tbody> </table> ● 4 のハイバネーションをするためには、端末の設定でハイバネーションが可能な設定になっていなければなりません。 	0	何もしない [デフォルト]	1	シャットダウン (電源切断を行うことはない。)	2	電源切断。	3	リブート	4	ハイバネーション
0	何もしない [デフォルト]										
1	シャットダウン (電源切断を行うことはない。)										
2	電源切断。										
3	リブート										
4	ハイバネーション										

ご注意ください

- 自動タイムアウトを用いるためには、co-notice.dll のほかに co-logon.dll も有効にする必要があります。
- 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。

3.1.1. 通常のログインダイアログ

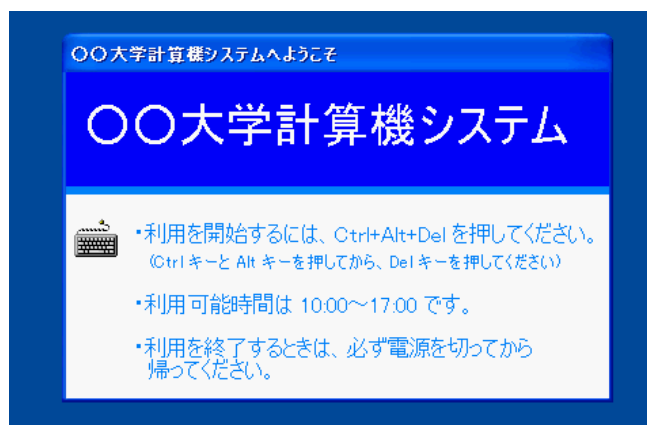
bitmap のパスを指定しないと通常の Ctrl+Alt+Del を押してくださいの画面が表示されます。以下のスクリーンショットは Windows XP の場合です。



この場合にも、自動タイムアウト機能は用いることができます。

3.1.2. ビットマップをダイアログに用いる場合

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。

3.2. co-logon

■ 機能

- 特殊変数つきで外部の認証スクリプトの呼び出すことができます。
- スクリプトから任意のメッセージを返すことができます。(ログインに成功した場合も、警告を出すことができます)
- スクリプトからどのようなユーザーでログインするかを設定することができます。

■ 初期設定

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。

```
[auth_script]
# 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=*

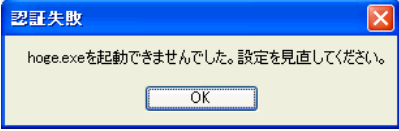
# 認証を行うスクリプト [実行リテラル][特殊変数:%u, %d, %p, %s, %S]
# 実行できないときは、エラーメッセージを表示する (ログインできない)
# デフォルト値はから文字列となり、実行できずにエラーとなる。
script=perl C:\¥GINA¥auth.pl "%u" %d

# 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=%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
```

■ 設定の詳細

auth_script	
msgina_account	<ul style="list-style-type: none"> ● MSGINA と同様の認証を行うアカウント名を記入して下さい。 ● 複数のアカウントを区切るには , (カンマ) を使って下さい。全てのアカウントにマッチさせるには * (アスタリスク) を使って下さい。
cogina_account	<ul style="list-style-type: none"> ● 認証スクリプトを使って認証を行うアカウント名を記入して下さい。 ● 複数のアカウントを区切るには , (カンマ) を使って下さい。全てのアカウントにマッチさせるには * (アスタリスク) を使って下さい。
script [実行リテラル] [特殊変数:%u,%d,%p]	<ul style="list-style-type: none"> ● 認証を行うスクリプトのパスを記述してください。 ● 実行できない場合は、エラーのダイアログが出力されます。  <ul style="list-style-type: none"> ● 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 2 (警告) で出力します。
stdin [メッセージリテラル] [特殊変数:%u,%d,%p]	<ul style="list-style-type: none"> ● 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、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ありません。
timeout [時間リテラル]	<ul style="list-style-type: none"> ● 認証スクリプトの終了を待つまでの時間です。
debug [真偽リテラル]	<ul style="list-style-type: none"> ● yes に設定すると、認証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。スクリプトのデバッグを行いやすくなります。

認証スクリプトを利用しない場合は、次のようにしてください。

```
[auth_script]
msgina_account=*
cogina_account=
```

認証スクリプトを利用する場合は、次のようにすると問題なさそうです。

```
[auth_script]
msgina_account=
cogina_account=* XXXXXXXXXX
```

しかし、このようにしてしまうと、全てのユーザー名でログインするときに認証スクリプトを利用してしまう。認証スクリプトを利用します。スクリプトに不具合があり実行できないような状況に陥ってしまうと、認証スクリプトを利用したログインはできなくなってしまいます。そのような万一の場合に備えるために、Administrator だけはログオンできるように設定することをお勧めします。

そこで、認証スクリプトを利用する場合は次のようにして下さい。

```
[auth_script]
msgina_account=Administrator
cogina_account=*
```

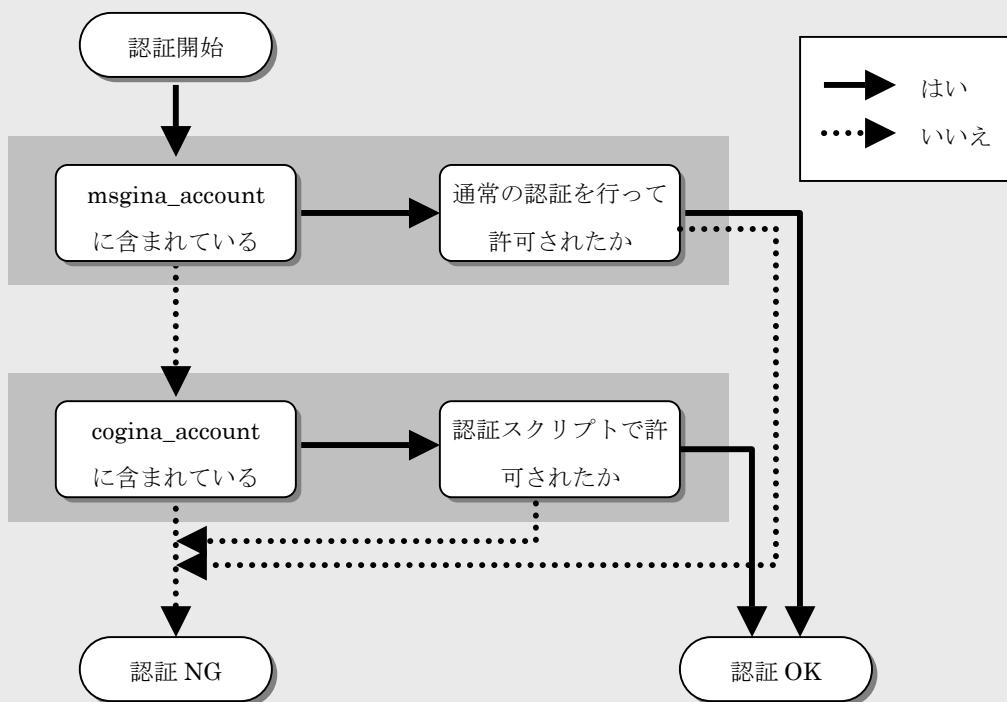
補足

ここでは、msgina_account と cogina_account の解釈順序について説明します。

利用者がログオン ダイアログにアカウント名とパスワードを入力して ログオン ボタンを押したとき、認証が開始します。

最初に、co-logon.dll は、アカウント名が msgina_account に含まれているかを確認します。含まれていた場合、通常の MSGINA と同様の認証を行います。もし、ここで認証が許可された場合は、認証 OK となります。認証が許可されなかった場合は、認証 NG となります。

msgina_account にアカウント名がなかった場合は、cogina_account にユーザー名が含まれているかを確認します。含まれている場合は認証スクリプトで認証を行い、許可された場合は認証 OK となります。認証スクリプトで認証が認められなかった場合や cogina_account に含まれていなかった場合は、認証 NG となります。



「script」に認証スクリプトのパスを指定します。認証スクリプトのスクリプトを実行できない場合は、エラーとなりログインできません。パスの指定では特殊変数の %u, %d, %p を使うことができます。

「stdin」に指定した文字列は、認証スクリプトに標準入力として与えることができます。例えば、「stdin=%d¥n%p」とした場合に、ユーザー名:test、パスワード:testpasswd、ドメイン名:mydomain としてログインすると、認証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
tesetpasswd
```

パスワードが「test¥np」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

```
mydomain
teset¥np
```

が標準入力として与えられます。

「timeout」は認証スクリプトが終了しなくなった場合に対処するためです。もし認証スクリプトから処理が帰ってこなくなってしまうと、電源を切るしかなくなります。timeout を設定す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ログオン ダイアログに戻ることができます。

かといって、「timeout」が短ければよいというわけではなく、短すぎると認証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い、中途半端な標準出力しかえることができません。

また、タイムアウトはプロセスの終了を待つじかんであり、タイムアウトの時間が経過してもプロセスは強制終了されず、実行され続けることに注意してください。

「debug」で yes を指定すると、認証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。これにより、認証スクリプトのデバッグをしやすくなります。ただし、yes にするのはあくまで認証スクリプトの検証中だけにしてください。最悪の場合、パスワードなどが表示されてしまう可能性があることに留意してください。

また、「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。

ご注意ください

認証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、遠隔から起動中のプロセス名を知る機能によって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

認証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。

3.2.1. 認証スクリプト

認証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。

■ 必要要件

認証スクリプトは、下記の必要要件を満たすならば、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ジャンルは問いません)

- co-logon.ini で与えられるコマンドライン引数や標準入力を読み取って、認証結果を規定の出力形態にしたがって標準出力に出力する。

■ 出力

認証スクリプトは、認証の結果を標準出力に出力することで co-logon.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。

```
# 認証に成功したか [真偽リテラル]
# yes : ログインさせる
# no  : エラー (デフォルト)
result=no

# Windows にログオンさせるときのアカウント名
# 省略した場合は、ユーザーがダイアログに入力したアカウント名が使用される
# ローカル or Active Directory 上にユーザーがいる必要がある。
# いない場合は、エラーとなる。
# user=

# Windows にログオンするアカウントのパスワード
# 省略した場合は、ユーザーがダイアログに入力したパスワードが使用される
# passwd=

# ログオンするドメイン
# 省略した場合は、ユーザーがダイアログに入力したパスワードが使用される
# domain=

# 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=アカウント名が間違っているかパスワードが間違っています。¥n もう一度確認してください。

# ダイアログのタイトル
# デフォルトは 警告
# title=警告

# アイコン [application|error|warning|information|question|パスリテラル]
#bitmap=warning

# メッセージを識別するための ID。
# script で渡すことができる。
# msg-id=id-xxxxx






# ダイアログをタイムアウトするまでの時間 [時間リテラル]
# -1 のときは、タイムアウトなし
# デフォルトは -1
# timeout=-1

# OK ボタンを押したときかタイムアウトになったときに実行するパス
# [特殊変数:%u,%d,%p,%i]
# %i で、msg-id を参照できる
# デフォルトは何も実行しない
#script=perl xxx.pl %i

# ボタンを表示するか [真偽リテラル]
# timeout が -1 でないときのみボタンを消すことができる。デフォルトは yes
button=yes
```

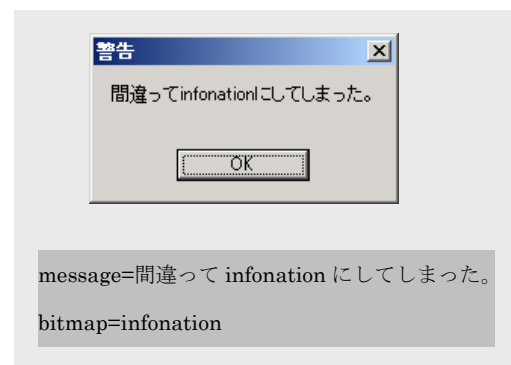
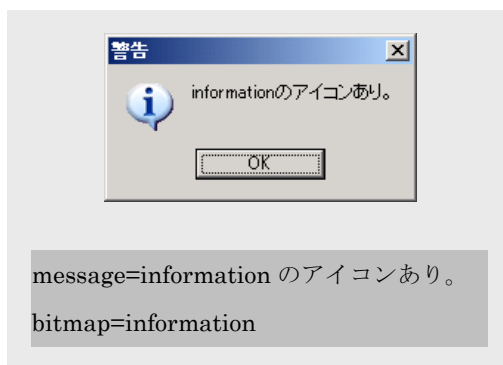
■ bitmap について

bitmap に以下のものを設定した場合は Windows 標準のアイコンとなります。

画像	名前	詳細
	warning	警告のアイコン (!マーク)
	information	情報のアイコン (i マーク)
	question	質問のアイコン (?マーク)
	error	エラー時のアイコン (X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、アイコンがなかったものとして扱われます。

図の左が正しく指定した例で、右が間違えていた場合の例です。



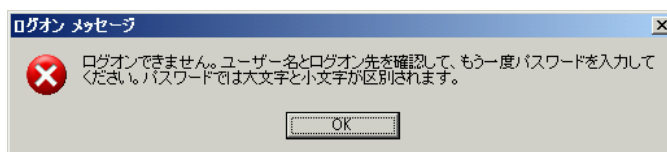
■ 認証スクリプトの出力例

ログインを拒絶する場合は以下のように出力すればよいです。

```
result=no
```

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にパスワードが間違っていることが伝わりません。そのため、次のようにしてエラーダイアログを出すのがよいでしょう。

```
result=no  
bitmap=error  
title=ログイン メッセージ  
message=ログインできません。ユーザー名とログイン先を確認して、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。
```



ログインを許す場合は次のようにしてください。

```
result=yes
```

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インさせる場合には、以下のようになります。次の例では、ユーザー名: test、パスワード: test でログインします。ただし、この場合は、ログインさせ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。

```
result=yes  
  
user=test  
passwd=test
```

次のようにすると、ログイ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。

```
result=yes  
  
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。
```

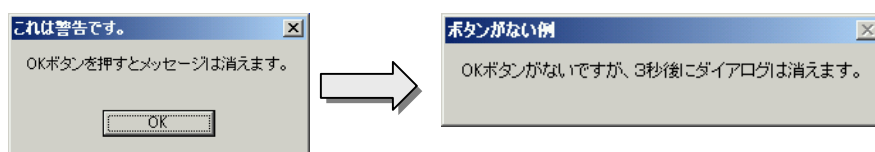


ログオン時に表示するメッセージは `message` を複数並べることで複数表示することができます。そのメッセージの内容は、次の `message` までの `title`, `timeout`, `script`, `button`, `msg-id`, `bitmap` で設定できます。

この例では、2つのダイアログを表示します。

```
message=OK 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ませんが3秒経つとタイムアウトします。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。

3.2.2. ユーザー名とパスワードが同じだと許可する例

まずは簡単な認証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インできるような認証スクリプトを考えてみます。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。

```
[auth_script]
# 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。
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]
script=perl C:\¥GINA¥auth.pl %u %d

# 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]
stdin=%p

# 認証スクリプトの終了を待つ時間。[時間リテラル]
timeout=15s

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

このようにすると、Administrator 以外のユーザー名でログインするときは、認証スクリプト「auth.pl」を起動するようになります。スクリプトが起動しないなどの状況に陥ってしまうと、認証スクリプトを利用したログインはできなくなってしまいます。そのような万一の場合でも administrator だけはログインできるようにするため、msgina_account に Administrator は入れておいてください。

次のスクリプトを `c:\¥gina¥auth.pl` として保存してください。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に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ログオンできませんので注意してください。

```
# 読みとり
$username = $ARGV[0];
$domain = $ARGV[1];
$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ありませんよ。¥¥n 再入力してね。¥n";
    exit(0);
}

# ここまでくれば認証成功
print "result=yes¥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 ($username,$password)=@_;
    return ($username eq $password? 0 : 1);
}
```

3.2.3. 独自の認証方法を使う場合の例

「3.2.2 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。

3.2.4. Active Directory を使う場合の例

ここでは、認証に ActiveDirectory やローカルアカウントを使う場合の認証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ったり、利用者ごとに異なる警告メッセージを出すことができます。

以下のようなスクリプトを `c:\¥GINA¥auth.pl` として保存してください。このスクリプトは `C:\¥GINA¥scripts¥auth.pl` にもおいてあります。また、`C:\¥GINA¥tools¥chkpwd.exe` を `C:\¥GINA` に移動しておいてください。設定ファイル `co-logon.ini` は「3.2.2 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¥chkpwd.exe';

# 読みとり
$username = $ARGV[0];
$domain = $ARGV[1];
$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ありません。¥¥n 再入力してね。¥n";
    exit(0);
}

# ここまでくれば認証成功
print "result=yes¥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);

    print "#"; # chkpwd の出力をコメントにする
    system("perl -e ¥¥print '$password'¥¥ | C:\¥¥GINA¥¥chkpwd.exe $username $domain");
    print "¥n";

    # return code が 0 の場合は認証成功
    return $? != 0;
}
```

3.3. co-script.dll

■ 機能

- 設定ファイルに記述されたログオンスクリプトを実行する。
- ユーザー権限・システム権限のいずれでも実行可能。
- 起動の順番、同期・非同期の設定も可能。

■ 設定

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
[onboot]
#path=perl c:\%gina%\script.pl %s %f
#wait=-1
#display=no

# ログイン時スクリプトを利用するときにはコメントをはずす
[onlogon]
#path=notepad.exe
#wait=0s
#display=yes
#exec_user=user

# ログオフ前スクリプト (ログオフを開始する前)
[prelogoff]
#path=cscript /nologo c:\%GINA%\logoff_ask.vbs
#wait=-1
#display=no
#exec_user=user

# ログオフ時スクリプト (ログオフ完了後)
[logoff]
#path=perl c:\%gina%\script.pl %s
#wait=-1
#display=no

# シャットダウン時スクリプト (ログオフ完了後、シャットダウンする場合)
[shutdown]
#path=perl c:\%gina%\script.pl %s
#wait=-1
#display=no

# スクリーンセーバー前スクリプト
[screensaver]
#path=
#wait=-1
#display=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	<ul style="list-style-type: none">● 電源投入時に呼び出されます。具体的には Ctrl+Alt+Del をおしよてくたさいのダイアログが表示される直前です。
onlogon	<ul style="list-style-type: none">● ログオン時。シェルが起動される前に呼ばれます。
prelogoff	<ul style="list-style-type: none">● ログオフ処理が行われる前に呼び出されます。● 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。● プロファイルの転送が行われる前です。
logoff	<ul style="list-style-type: none">● ログオフ処理が行われた後に呼び出されます。● 通常のログオフスクリプトと同じ場所です。
shutdown	<ul style="list-style-type: none">● シャットダウン・リブート時に、呼び出されます。● 通常のログオフスクリプトと同じ場所です。
screensaver	<ul style="list-style-type: none">● スクリーンセーバーが呼ばれる前に呼び出されます。● 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。

ご注意ください

- Windows 2000 の場合、onboot 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。有効にするには、コントロールパネルの「ユーザーとパスワード」の「詳細」から「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。
- 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。

■ 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。

それぞれのスクリプトの起動方法は `path`, `wait`, `display`, `exec_user` のパラメータで設定します。これらのパラメータの意味については下表を参照してください。

`path` を並べることで、複数のアプリケーションを順番に立ち上げることができます。起動方法は、次の `path` かセクションの最後まで `wait`, `display`, `exec_user` で設定できます。

path [実行リテラル] [特殊変数:%s, %S, %f]	<ul style="list-style-type: none"> ● 実行するアプリケーションのパスを指定します。
wait [時間リテラル]	<ul style="list-style-type: none"> ● 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。 ● 指定された時間だけ待ちます。待っている間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ずなかった場合、次のアプリケーションを起動します。 ● 1つ前のアプリケーションの終了しか待機しないことに注意してください。 ● -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。
display [真偽リテラル]	<ul style="list-style-type: none"> ● ウィンドウを表示させるかどうかを表します。デフォルトは <code>yes</code> です。 ● <code>perl</code> や <code>csript(WSH)</code> などは、MS-DOS プロンプトが表示されるため、<code>no</code> を指定することでウィンドウが表示されなくなります。
exec_user [user もしくは system]	<ul style="list-style-type: none"> ● 実行者の権限を指定します。<code>system</code> は端末に関する全ての権限をもちます。<code>user</code> はログオンしたユーザーの実行権限をあらわします。 ➔ インタラクティブなアプリケーションをシステム権限で実行しないようにしてください。利用者が <code>system</code> 権限で任意のアプリケーションを (ファイルを開くダイアログなどから) 起動できるようになってしまう可能性があります。 ● <code>user</code> は <code>onlogon</code> および <code>prelogoff</code> でのみ利用できます。<code>user</code> を利用するためには、<code>co-logon.dll</code> プラグインを有効にしなければなりません。 ● デフォルトは <code>system</code> です。
debug [真偽リテラル]	<ul style="list-style-type: none"> ● 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 <code>no</code> です。 ● <code>yes</code> にすると、レベル 3 でログに標準出力を出力します。 ● <code>no</code> でも標準エラーはレベル 2 でログに出力されます。

スクリプトがどのタイミングで呼び出されたかは特殊変数 `%s` で知ることができます。`%s` の値については、「2.1.4 ステータス」を参照してください。

例えば、`logoff` と `shutdown` の `path` を次のように設定することができます。

```
[logoff]
path=perl exit.pl %s

[shutdown]
path=perl exit.pl %s
```

このとき、スクリプトの側でコマンドライン引数として与えられる `%s` を解析することで、1 つのスクリプトで複数のタイミングの処理を行うことができます。

`%f` は、`onboot` のみ利用できます。

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

これにより、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)

■ 補足

- ログオフ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いでしょう。なぜならば、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。起動するスクリプトが、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いでしょう。(システムモーダルなメッセージボックスが表示中には、他のウィ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することができます。

3.3.1. 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。

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

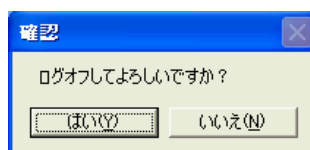
```
Set WSHShell=Wscript.CreateObject("Wscript.Shell")
Dim intMsg
intMsg = WSHShell.Popup("ログオフしてよろしいですか?", 0, "確認", vbYesNo)

if intMsg = vbNo then
    Wscript.echo("1")
end if
```

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]
path=cscript /nolog C:\¥GINA¥logoff_ask.vbs
wait=-1
display=no
exec_user=user
```

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。



3.4. co-option.dll

■ 機能

- パスワード変更のタイミングでプロセスを起動したり、パスワード変更スクリプトを起動できます。

■ 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。

```
[passwd]
# パスワードの変更で実行するプログラム [真偽リテラル]
# デフォルトは no
# use_hook と同時には yes にできない
#use_passwd_exec=yes

# パスワードのフックを行うかどうか [真偽リテラル]
# デフォルトは no
# use_passwdexec と同時には yes にできない
#use_chpwd_script=yes

[passwd_exec]
# パスワードの実行を行うプログラム
#path="C:¥Program Files¥Internet Explorer¥iexplore" https://www.hoge.co.jp/password/

[chpwd_script]
# フックを行うスクリプト [実行リテラル][特殊変数:%u, %d, %p, %s, %S]
# デフォルト値は空文字列となり、実行できずにエラーとなる。
script=perl C:¥tmp¥passwd.pl %u

# 標準入力から与えるもの [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=%p¥n%P¥n%o

# スクリプトのタイムアウト [時間リテラル]
timeout=-1
```

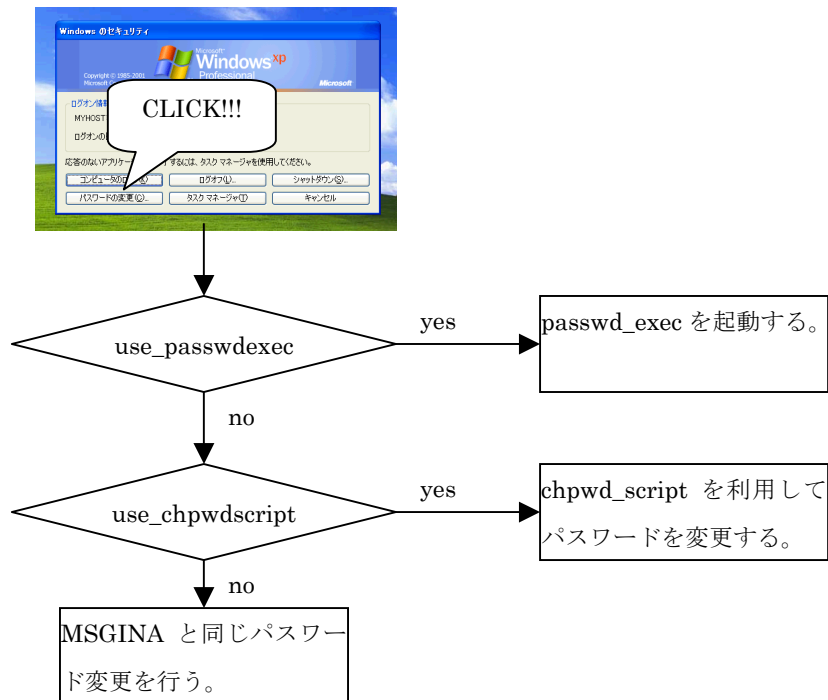
■ 設定の詳細

passwd	
use_passwd_exec [真偽リテラル]	<ul style="list-style-type: none"> ● 「パスワードの変更」ボタンを押したときにプロセスを起動する場合は「yes」に設定して下さい。デフォルトは「no」です。
use_chpwd_script [真偽リテラル]	<ul style="list-style-type: none"> ● 「パスワードの変更」を行う部分で独自スクリプトを利用する場合は「yes」に設定して下さい。デフォルトは「no」です。
passwd_exec	
path [実行リテラル]	<ul style="list-style-type: none"> ● 「パスワードの変更」ボタンを押したときに実行するプロセスを記述して下さい。 ● 現在のところ、特殊変数は利用できません。
chpwd_script	
msgina_account	<ul style="list-style-type: none"> ● MSGINA と同じパスワード変更を行うアカウント名を記入します。 ● 複数のアカウントを区切るには、(カンマ) を使って下さい。全てのアカウントにマッチさせるには * (アスタリスク) を使って下さい。
cogina_account	<ul style="list-style-type: none"> ● 認証スクリプトを使ってパスワード変更を行うアカウント名を記入して下さい。 ● 複数のアカウントを区切るには、(カンマ) を使って下さい。全てのアカウントにマッチさせるには * (アスタリスク) を使って下さい。
script [実行リテラル] [特殊変数:%u,%d,%p, %s, %S]	<ul style="list-style-type: none"> ● パスワードの変更を行う独自スクリプトのパスを設定して下さい。 ● 実行できない場合はエラーが表示されます。
timeout [時間リテラル]	<ul style="list-style-type: none"> ● パスワード変更を行うスクリプトの終了を待つ時間です。
stdin [メッセージリテラル] [特殊変数:%u,%d,%p, %P,%o, %s, %S]	<ul style="list-style-type: none"> ● パスワード変更を行うスクリプトの標準入力として設定する文字列です。 ● 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ありません。

co-option.dll では、パスワード変更を行う方法は次のような3通りの方法が用意されています。

- プロセスを起動する(passwd_exec) : passwd_exec はプロセスを利用者権限で起動して、ダイアログの表示を終了します。例えば、Web 上でパスワード変更用のインターフェースを用意する場合には、利用者をそのページに誘導できます。
- パスワード変更スクリプト(chpwd_script)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、変更部分の実装を独自のスクリプトで行うことができます。独自の認証体系を利用する場合に有用です。
- MSGINA と同じパスワード変更を行う : 通常の Windows と同じ方法でパスワードの変更を行います。Windows 標準の認証方法を用いる場合は、通常と同じ方法でパスワード変更を行ってください。

パスワード変更時の流れは以下のようになっています。



「passwd_exec」を利用する場合は、起動するプロセスを path に記述してください。path に記述されたアプリケーションを起動すると、利用者のデスクトップに処理が戻ります。

「chpwd_script」はオプションダイアログでパスワード変更をクリックしたときに呼び出されます。パスワード変更スクリプトの詳細については「3.4.1 パスワード変更スクリプトの仕様」をご覧ください。msgina_account および cogina_account については、

3.4.1. パスワード変更スクリプトの仕様

■ 概要

- パスワードの変更を行うスクリプト

■ 出力

パスワード変更スクリプトは、その結果を標準出力を用いて `co-option.dll` に返す。

出力は `ini` ファイルの形式にのっとり行う。以下にその形式を記す。

```
# パスワード変更できたか [真偽リテラル]
# yes : 変更された
# no  : エラー (デフォルト)
result=no

# 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。
```

`result` が `yes` のときは、パスワード変更ダイアログは閉じられます。`no` のときは、パスワード変更ダイアログは閉じられません。

いずれの場合も、`message` があればその内容をダイアログで表示します。

3.5. co-lock.dll

■ 概要

- タイムリミットつきロックダイアログの表示。

■ 設定

co-lock.ini で次の項目を設定できる。

```
[gui]
# 自動ログオフまでの秒数。 [時間リテラル]
# -1 なら自動ログオフしない
# 0 ならすぐに自動ログオフ。
# 最大値は....
timeout=300s

# メッセージの内容 [メッセージリテラル,特殊変数:%u,%d]
# %u: ログオン時のアカウント名 (co-logon.dll を使用したときのみ有効)
# %t: タイムアウトまでの時間
# デフォルトは、msgina.dll が表示するメッセージのまま。
message=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 が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。

[unlock_script]
# 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。
msgina_account=*
cogina_account=

# 実行するスクリプト [特殊変数 %u, %p, %d, %s, %S]
script=

# 標準入力 [特殊変数 %u, %p, %d, %s, %S]
stdin=%p

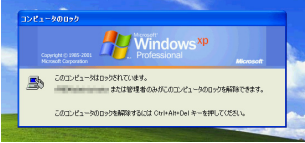
# 認証スクリプトの終了を待つ時間。 [時間リテラル]
# 0 だと、スクリプトが終了するまで処理が帰らない。そのため
# スクリプトが途中で止まった場合は、電源を切らないといけなくなる。
timeout=-1

# 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

- Windows 2000 の場合、ロックのタイムアウトは「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。有効にするには、コントロールパネルの「ユーザーとパスワード」の「詳細」から「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。

■ 設定の詳細

gui	
timeout [時間リテラル]	<ul style="list-style-type: none"> ● ロックの自動タイムアウトまでの時間を指定します。 ● -1 では自動ログオフしません。デフォルトは-1 です。
message [メッセージリテラル] [特殊変数: %u, %d, %t]	<ul style="list-style-type: none"> ● ロック通知ダイアログに表示するメッセージの内容です。  <ul style="list-style-type: none"> ● デフォルトは、msgina.dll が表示するメッセージのままです。
unlock_script	
msgina_account	<ul style="list-style-type: none"> ● MSGINA と同じロック解除を行うアカウント名を記入します。 ● 複数のアカウントを区切るには , (カンマ) を使って下さい。全てのアカウントにマッチさせるには * (アスタリスク) を使って下さい。
cogina_account	<ul style="list-style-type: none"> ● 認証スクリプトを使ってロック解除を行うアカウント名を記入して下さい。 ● 複数のアカウントを区切るには , (カンマ) を使って下さい。全てのアカウントにマッチさせるには * (アスタリスク) を使って下さい。
script [実行リテラル] [特殊変数:%u,%d,%p, %s, %S]	<ul style="list-style-type: none"> ● ロック解除を行う独自スクリプトのパスを設定してください。 ● 実行できない場合はエラーが表示されます。
timeout [時間リテラル]	<ul style="list-style-type: none"> ● ロック解除を行うスクリプトの終了を待つ時間です。
stdin [メッセージリテラル] [特殊変数%u,%p,%d %s, %S]	<ul style="list-style-type: none"> ● ロック解除を行うスクリプトの標準入力として設定する文字列です。 ● 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ありません。

3.5.1. ロック解除スクリプトの仕様

ロック解除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。

■ 必要要件

ロック解除スクリプトは、下記の必要要件を満たすならば、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ジャンルは問いません)

- `co-lock.ini` で指定されたコマンドライン引数や標準入力を読み取って、認証結果を規定の出力形態にしたがって標準出力に出力する。

■ 出力

ロック解除スクリプトは、結果を標準出力に出力することで `co-lock.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。

```
# パスワード変更できたか [真偽リテラル]
# yes : 変更された
# no  : エラー (デフォルト)
result=no

# 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=パスワードが異なります
```

■ 補足

`result` が `yes` のときは、ロックが解除されます。`no` のときは、ロックされたままとなり、ダイアログは閉じられません。なお、`message` の内容は `result` の結果によらず出力されます。

4. 付録

4.1. 過去の変更履歴

改変履歴は `/doc/changes.txt` を参照してください。

CO-GINA 設定マニュアル

有限会社 CO-CONV
第 3 版 (Version 2.1.0.2 対応)
2003 年 12 月 04 日
