



IceWall SSO

Version 10.0

UserExit ルーチン開発者マニュアル

日本ヒューレット・パッカード株式会社
2010 年 8 月

Printed in Japan

HP Part No. B2873-97009

Rev.100819A

ご注意

本書に記載されました内容は、予告なしに変更することがあります。

本書は内容について細心の注意をもって作成いたしましたが、万一ご不審な点や誤り、記載もれなど、お気づきの点がございましたら当社までお知らせください。

当社は、本書に記載されている誤り、あるいは備品、パフォーマンス、または本書の使用に関連して発生する偶然または必然的な損傷について責任は負いかねます。

本書の著作権は Hewlett-Packard Development Company, L.P. に帰属します。本書の一部あるいは全部についての無断複製、転載を禁じます。

本製品パッケージとして提供した本マニュアル、CD-ROM 等の媒体は本製品用だけにお使いください。

IceWall はヒューレット・パッカー社社の商標です。

Adobe は Adobe Systems Incorporated の米国及びその他の国における登録商標または商標です。

Java は、米国 Sun Microsystems, Inc. の米国およびその他の国における商標または登録商標です。

Microsoft および Windows は、米国 Microsoft Corporation の、米国、日本およびその他の国における登録商標または商標です。

Oracle は、Oracle Corporation 及びその子会社、関連会社の米国及びその他の国における登録商標です。

— 目次 —

1 はじめに	1
1.1 文中におけるバージョン表示記号について	1
2 UserExit ルーチンとは	2
2.1 オリジナル認証処理の追加が可能	3
2.2 画面制御が可能	3
3 開発の手順	4
3.1 仕様の決定	4
3.2 コーディング・ビルド	4
3.3 テスト・デバッグ	5
4 UserExit ルーチン仕様	7
4.1 フォワーダー用 UserExit ルーチン仕様	7
IW_ExDFWInterFace	8
4.1.1 インターフェイス関数内 API	15
IW_ExDFWChangeProtocol	16
IW_ExDFWGetMethod	17
IW_ExDFWModifyMethod	18
IW_ExDFWGetPostdata	19
IW_ExDFWModifyPostdata	20
IW_ExDFWGetBuffer	21
IW_ExDFWModifyBuffer	22
IW_ExDFWModifyBufferLength	23
IW_ExDFWModifyRequestPath	24
IW_ExDFWModifyPathinfo	26
IW_ExDFWGetUid	27
IW_ExDFWGetCertData	29
IW_ExDFWCtrlCertData	31
4.2 認証モジュール用 UserExit ルーチン仕様	33
IW_ExInterFace ⑩.0	34
4.2.1 インターフェイス関数内 API	43
IW_ExCSNoSelectDBClm	44
IW_ExCSGetCacheUser	45
IW_ExCSModifyCacheUser ⑩.0	47
IW_ExCSReleaseRecord	49
IW_ExCSGetICP	50
IW_ExCSAddICP	51
IW_ExCSModifyICP	52
IW_ExCSDeleteICP	54
IW_ExCSAddCacheUserGroup ⑩.0	56
IW_ExCSDeleteCacheUserGroup	58
IW_ExCSGetLoginUserCount ⑩.0	59
5 注意事項	60
5.1 UserExit ルーチンの影響範囲	60
5.2 パフォーマンスの低下	60

5.3 メモリリーク	60
5.4 従来バージョンからのバージョンアップについての注意事項	60
5.5 サポートについて	60
6 制限事項	61
6.1 モジュール共通の制限	61
6.1.1 リンクする標準ライブラリのバージョンについて	61
6.1.2 互換性について	61
6.2 フォワーダーの制限	61
6.2.1 OS 共通の制限	61
6.2.2 HP-UX 版 (Itanium)	61
6.2.3 Linux 版	62
6.3 認証モジュールの制限	62
6.3.1 OS 共通の制限	62
6.3.2 HP-UX 11i v3 版 (Itanium)	62
6.3.3 Linux 版	62
7 UserExit ルーチン開発のヒント	63
7.1 フォワーダー UserExit ルーチン	63
7.1.1 IceWall に対するリクエストの判別方法	63
7.1.2 バックエンド Web サーバーへのリクエストを制御する方法	63
7.1.3 バックエンド Web サーバーからのレスポンスを制御する方法	63
7.2 認証モジュール UserExit ルーチン	64
7.2.1 各処理後のエラー判別方法	64
8 サンプルソース	65
8.1 フォワーダー UserExit サンプル	65
8.1.1 GET メソッドを POST メソッドへ変更	65
8.1.2 リクエストの HTTP ヘッダーを変更	65
8.2 認証モジュール UserExit のサンプル	66
8.2.1 特定のユーザー ID を強制的に guest へ変更	66
8.2.2 特定の URL に対するリクエストをログイン要求に変更	66
9 参考	67
9.1 フォワーダー用 UserExit ルーチン開発キット	67
9.1.1 スケルトンソース (dfwinterface.c)	67
9.1.2 ヘッダーファイル (dfwinterface.h)	68
9.1.3 メイクファイル (HP-UX Itanium64Bit 版 : Makefile)	70
9.1.4 メイクファイル (Linux64Bit 版 : Makefile)	71
9.2 認証モジュール用 UserExit ルーチン開発キット	72
9.2.1 スケルトンソース (iwintface.c)	72
9.2.2 ヘッダーファイル (iwintface.h)	73
9.2.3 メイクファイル (HP-UX Itanium 64Bit 版 : Makefile)	75
9.2.4 メイクファイル (Linux64Bit 版 : Makefile)	76

1 はじめに

本書では、フォワーダーおよび認証モジュールに組み込む UserExit ルーチンの開発に必要な情報が記述してあります。

1.1 文中におけるバージョン表示記号について

記述中に付加されているバージョン表示記号はそれぞれ以下の意味を表します。

表示記号	意味
10.0	四角で囲まれているバージョンで追加された項目です。この場合、10.0 にて追加されたことを表します。
10.0	楕円で囲まれているバージョンで仕様変更もしくは機能追加された項目です。この場合、10.0 にて仕様変更または機能追加されたことを表します。

2 UserExit ルーチンとは

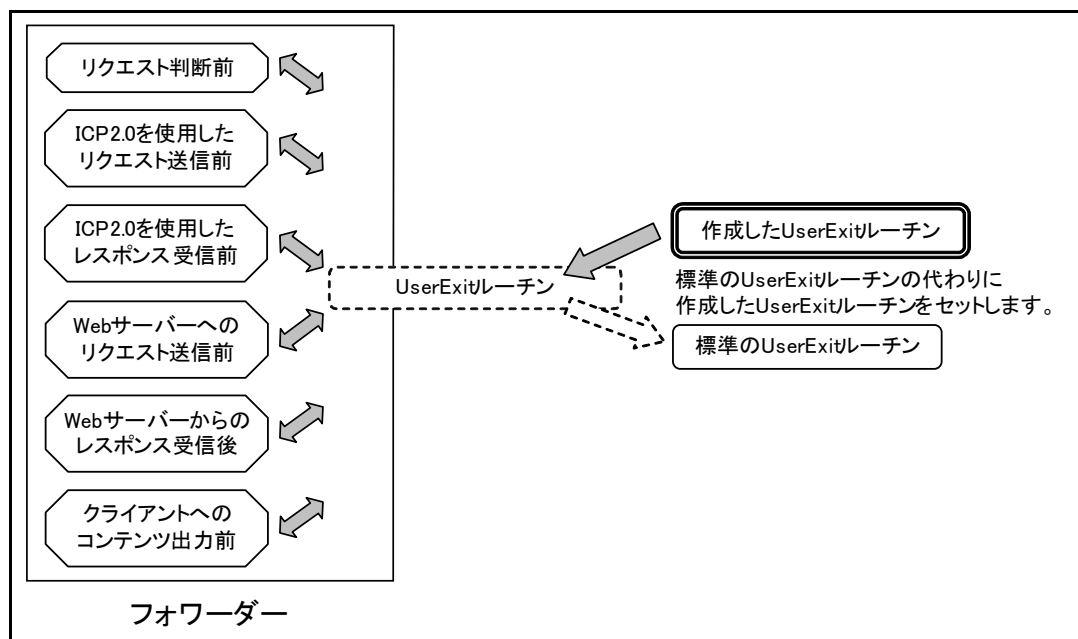
UserExit ルーチンとは、フォワーダーと認証モジュールの特定の処理に対して任意の処理を追加できる機能です。

各プロセスにて任意の処理を追加できる箇所は以下のとおりです。

プロセス	追加できる箇所
フォワーダー	クライアントからのリクエスト判断前
	ICP 2.0 を使用した認証モジュールへのログインリクエスト送信前
	ICP 2.0 を使用した認証モジュールからのログインレスポンス受信後
	ICP 2.0 を使用した認証モジュールへのログインリクエスト (Agent) 送信前
	ICP 2.0 を使用した認証モジュールからのログインレスポンス (Agent) 受信後
	ICP 2.0 を使用した認証モジュールへのアクセス制御リクエスト送信前
	ICP 2.0 を使用した認証モジュールからのアクセス制御レスポンス受信後
	ICP 2.0 を使用した認証モジュールへのログアウトリクエスト送信前
	ICP 2.0 を使用した認証モジュールからのログアウトレスポンス受信後
	ICP 2.0 を使用した認証モジュールへのパスワード変更リクエスト送信前
	ICP 2.0 を使用した認証モジュールからのパスワード変更レスポンス受信後
	バックエンド Web サーバーへのリクエスト送信前
	バックエンド Web サーバーからのレスポンス受信後
	クライアントへのコンテンツ出力前
認証モジュール	ユーザー認証前
	ユーザー認証後
	ユーザー認証エラー後
	アクセス制御前
	アクセス制御後
	アクセス制御エラー後
	パスワード変更前
	パスワード変更後
	パスワード変更エラー後

プロセス	追加できる箇所
認証モジュール	ログアウト前
	ログアウト後
	ログアウトエラー後

UserExit ルーチンは Shared Library として提供されています。機能を追加する場合は、新たにその目的用の Shared Library を作成する必要があります。



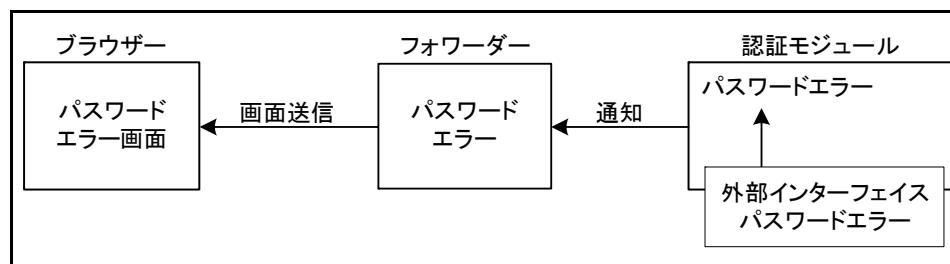
UserExit ルーチン概念図 (例：フォワーダー)

2.1 オリジナル認証処理の追加が可能

UserExit ルーチン内で、他のセキュリティ製品を使用した認証処理を組み込むことで、オリジナルかつ、より強力な認証機能を構築することが可能です。

2.2 画面制御が可能

UserExit ルーチン内の処理結果として用意されている値を返すことにより、ブラウザーに表示される画面をコントロールすることが可能です。



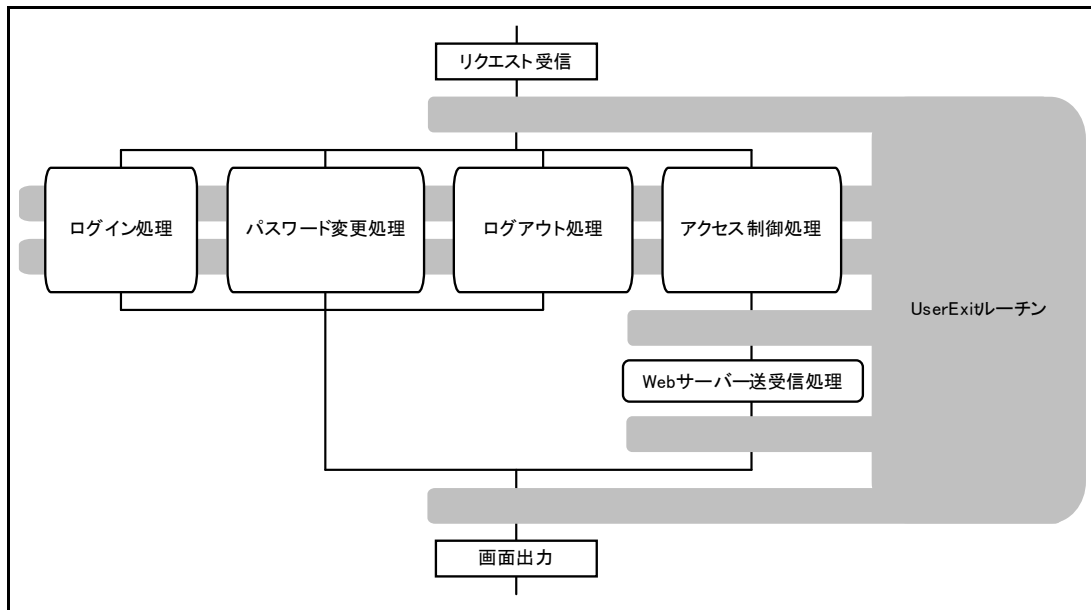
画面制御例 (認証モジュール：パスワードエラーの場合)

3 開発の手順

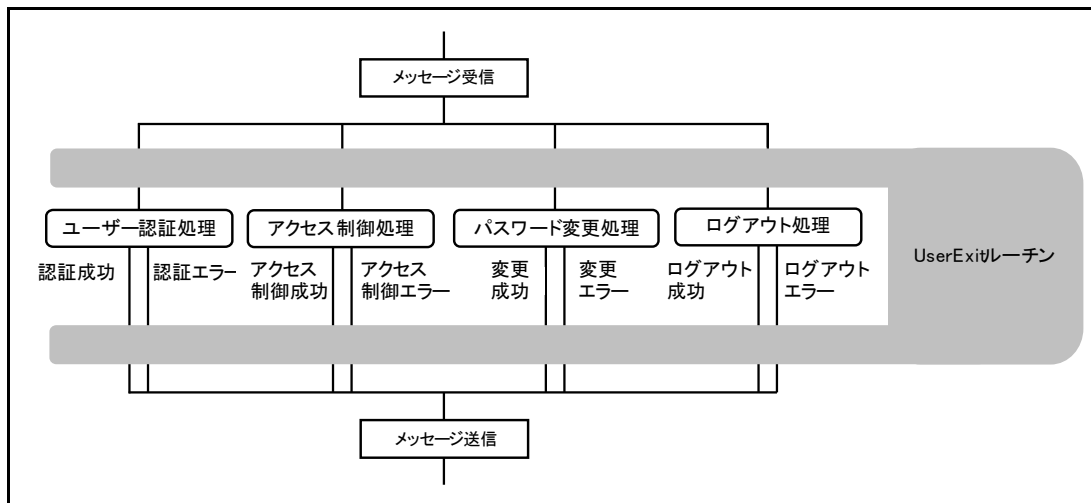
UserExit ルーチンの開発を行うための一般的な手順をご説明いたします。

3.1 仕様の決定

UserExit ルーチン内で行う処理、追加する箇所を決定します。



フォワーダーの UserExit ルーチン概要図



認証モジュールの UserExit ルーチン概要図

3.2 コーディング・ビルド

決定した仕様に基づき、UserExit ルーチンのソースコードをコーディングします。コーディング後に Shared Library としてビルドします。

コーディングは、開発キットに含まれるスケルトンソースを利用してください。
ビルドは開発キットに含まれる Makefile で行います。

フォワーダーの場合

```
$ cd /opt/icewall-ssso/developkit/dfw/DfwExit
$ make -f Makefile
```

認証モジュールの場合

```
$ cd /opt/icewall-ssso/developkit/certd/CertExit
$ make -f Makefile
```

3.3 テスト・デバッグ

テストプログラムなどにより、作成された **Shared Library** のテストおよびデバッグを行います。テストが終了した後に、ライブラリをフォワーダーまたは認証モジュールへ組み込み、動作テストを行います。

インストールは以下の手順で行います。

フォワーダーの場合

- (1) 標準でインストールされる **UserExit** ルーチンをバックアップします。

```
$ cd /opt/icewall-ssso/lib
$ cp -p libDfwExit.sl libDfwExit.sl.bak
```

- (2) 作成した **UserExit** ルーチンをコピーします。

```
$ cp -p /opt/icewall-ssso/developkit/dfw/DfwExit/libDfwExit.sl
/opt/icewall-ssso/lib
```

認証モジュールの場合

- (1) 認証モジュールを停止します。

```
$ /opt/icewall-ssso/certd/bin/end-cert
```

- (2) 標準でインストールされる **UserExit** ルーチンをバックアップします。

```
$ cd /opt/icewall-ssso/lib
$ cp -p libCertExit.sl libCertExit.sl.bak
```

- (3) 作成した UserExit ルーチンをコピーします。

```
$ cp -p /opt/icewall-ssso/developkit/certd/CertExit/  
libCertExit.sl /opt/icewall-ssso/lib
```

- (4) 認証モジュールを起動します。

```
$ /opt/icewall-ssso/certd/bin/start-cert
```

4 UserExit ルーチン仕様

フォワーダー用と認証モジュール用の UserExit ルーチンはそれぞれ仕様が異なります。
本章では各 UserExit ルーチンの仕様、および使用可能な API を解説します。

4.1 フォワーダー用 UserExit ルーチン仕様

フォワーダー用 UserExit ルーチンでは以下のものが定義されています。

- ・ フォワーダーよりコールされるインターフェイス関数
- ・ インターフェイス関数の戻り値としてクライアントに表示する画面を制御するコード
- ・ インターフェイス関数内で使用可能な API

次ページよりこれらの解説を行います。

IW_ExDFWInterFace

フォワーダー用 UserExit ルーチンのインターフェイス関数の仕様は以下のとおりです。

書式 **int IW_ExDFWInterFace(EX_DFW_REQ* req, int kind)**

引数 **req** : UserExit ルーチンリクエスト構造体です。この構造体にフォワーダーの情報が格納されています。この構造体の情報を操作するには API を使用します。

kind : UserExit ルーチンがコールされた位置情報が格納されます。以下の値が設定されます。

EX_DFW_KIND_REQ	: クライアントからのリクエスト判断前
EX_DFW_KIND_LOGIN_SEND	: 認証モジュールへのログインリクエスト送信前
EX_DFW_KIND_LOGIN_RECV	: 認証モジュールからのログインレスポンス受信後
EX_DFW_KIND_ALOGIN_SEND	: 認証モジュールへのログインリクエスト (Agent) 送信前
EX_DFW_KIND_ALOGIN_RECV	: 認証モジュールからのログインレスポンス (Agent) 受信後
EX_DFW_KIND_ACC_SEND	: 認証モジュールへのアクセス制御リクエスト送信前
EX_DFW_KIND_ACC_RECV	: 認証モジュールからのアクセス制御レスポンス受信後
EX_DFW_KIND_LOGOUT_SEND	: 認証モジュールへのログアウトリクエスト送信前
EX_DFW_KIND_LOGOUT_RECV	: 認証モジュールからのログアウトレスポンス受信後
EX_DFW_KIND_PWDCHG_SEND	: 認証モジュールへのパスワード変更リクエスト送信前
EX_DFW_KIND_PWDCHG_RECV	: 認証モジュールからのパスワード変更レスポンス受信後
EX_DFW_KIND_SEND	: Web サーバーへのリクエスト送信前
EX_DFW_KIND_RECV	: Web サーバーからのレスポンス受信後
EX_DFW_KIND_END	: コンテンツ出力前

戻り値 クライアントに表示するページを制御するコードを返します。
使用可能なコードについては後述の「フォワーダー UserExit 用クライアント画面制御コード」を参照してください。

制限事項 設定ファイルにて定義した内容により UserExit ルーチンが**実行されない**場合があります。**実行されない条件**とインターフェイス関数の位置については以下ようになります。

位置情報	実行されない条件
EX_DFW_KIND_REQ	・ 必ず実行されます
EX_DFW_KIND_LOGIN_SEND	・ ICP 2.0 を使用したログイン以外
EX_DFW_KIND_LOGIN_RECV	
EX_DFW_KIND_ALOGIN_SEND	・ ICP 2.0 を使用したエージェントからのログイン要求以外
EX_DFW_KIND_ALOGIN_RECV	
EX_DFW_KIND_ACC_SEND	・ ICP 2.0 を使用したアクセス制御以外
EX_DFW_KIND_ACC_RECV	
EX_DFW_KIND_LOGOUT_SEND	・ ICP 2.0 を使用したログアウト以外
EX_DFW_KIND_LOGOUT_RECV	
EX_DFW_KIND_PWDC_HG_SEND	・ ICP 2.0 を使用したパスワード変更以外
EX_DFW_KIND_PWDC_HG_RECV	
EX_DFW_KIND_SEND	・ アクセス制御以外
EX_DFW_KIND_RECV	・ アクセス制御処理で BUFFER 項目に 0 が設定されていてかつ、受信したコンテンツが CTYPE 項目で設定されていない Content-Type である。
EX_DFW_KIND_END	

この関数で渡される引数 req および kind の内容は変更できません。内容を変更した場合の動作は保障されません。

フォワーダー UserExit 用クライアント画面制御コード

フォワーダー用 UserExit ルーチンにて使用可能なクライアント画面制御コードは以下のとおりです。表示される画面については「IceWall SSO 標準 HTML カスタマイズガイド」および「IceWall SSO リファレンスマニュアル」を参照してください。

制御コード	内容	
EX_DFW_OK	内容	UserExit ルーチン内の処理終了後にフォワーダーの処理を続行する場合に使用します。
	表示画面	この制御コードを使用した場合に表示される画面は、その後のフォワーダーの動作に依存します。
EX_DFW_LOGIN	内容	クライアントにユーザー ID によるログイン要求を行う場合に使用します。
	表示画面	ユーザー ID とパスワード用ログインページ (login.html)
EX_DFW_LOGINCERT	内容	クライアントにクライアント証明書によるログイン要求を行う場合に使用します。この制御コードを使用する場合は、別途「IceWall SSO Client Certificates Option」が必要となります。
	表示画面	クライアント証明書とパスワード用ログインページ (login_cert.html)
EX_DFW_LOGINFORCE	内容	クライアントに強制ログイン要求を行う場合に使用します。
	表示画面	強制ログインページ (login_force.html)。ただし、リクエスト判断前以外で使用した場合、その後の動作は保証されません。
EX_DFW_LOGINUIDERR	内容	クライアントからのログイン要求をユーザー ID 入力エラーとする場合に使用します。
	表示画面	ユーザー ID エラーページ (login_userid_error.html)
EX_DFW_LOGINPWDERR	内容	クライアントからのログイン要求をパスワード入力エラーとする場合に使用します。
	表示画面	パスワードエラーページ (login_pwd_error.html)
EX_DFW_LOGINLOCKERR	内容	クライアントからのログイン要求をアカウントロックエラーとする場合に使用します。
	表示画面	アカウントロックエラーページ (login_lock_error.html)
EX_DFW_LOGINCERTERR	内容	クライアントからのログイン要求をクライアント証明書による初回認証済みエラーとする場合に使用します。この制御コードを使用する場合は、別途「IceWall SSO Client Certificates Option」が必要となります。
	表示画面	初回認証済みエラーページ (login_cert_error.html)

制御コード	内容	
EX_DFW_LOGINSERIALERR	内容	クライアントからのログイン要求をクライアント証明書シリアル No 不正エラーとする場合に使用します。この制御コードを使用する場合は、別途「IceWall SSO Client Certificates Option」が必要となります。
	表示画面	不正シリアル No エラーページ (login_error.html)
EX_DFW_LOGINGRPERR	内容	クライアントからのログイン要求をグループなしエラーとする場合に使用します。
	表示画面	所属グループなしエラーページ (login_group_error.html)
EX_DFW_LOGINSTOP	内容	クライアントからのログイン要求をログイン停止エラーとする場合に使用します。
	表示画面	ログイン停止エラーページ (login_stop.html)
EX_DFW_LOGINLIMITERR	内容	クライアントからのログイン要求をログイン上限エラーとする場合に使用します。 ※ 現状、認証モジュールの UserExit でログイン上限エラー画面は出せません。
	表示画面	ログイン上限エラーページ (login_limit_error.html)
EX_DFW_LOGOUT	内容	クライアントにログアウト要求を行う場合に使用します。
	表示画面	ログアウトページ (logout.html)
EX_DFW_LOGOUTOK	内容	クライアントからのログアウト要求をログアウト成功とする場合に使用します。
	表示画面	ログアウト成功ページ (logout_ok.html)
EX_DFW_LOGOUTNO	内容	クライアントからのログアウト要求をログアウト済みエラーとする場合に使用します。
	表示画面	ログアウト不可ページ (logout_no.html)
EX_DFW_LOGOUTERR	内容	クライアントからのログアウト要求をログアウト失敗エラーとさせる場合に使用します。
	表示画面	ログアウトエラーページ (logout_error.html)
EX_DFW_ACCESS	内容	クライアントからのアクセス制御要求をアクセス権限エラーとする場合に使用します。
	表示画面	アクセス権限エラーページ (access_error.html)
EX_DFW_SENERR	内容	クライアントよりデータを送信されたときにデータ送信エラーとする場合に使用します。
	表示画面	データ送信エラーページ (datasend_error.html)
EX_DFW_PWDCHG	内容	クライアントにパスワード変更要求を行う場合に使用します。
	表示画面	パスワード変更ページ (pwdchg.html)
EX_DFW_PWDCHGOK	内容	クライアントからのパスワード変更要求をパスワード変更成功とする場合に使用します。
	表示画面	パスワード変更成功ページ (pwdchg_ok.html)
EX_DFW_PWDOLDERR	内容	クライアントからのパスワード変更要求を現パスワードの入力エラーとする場合に使用します。
	表示画面	現パスワード入力エラーページ (pwdchg_oldpasserr.html)

制御コード	内容	
EX_DFW_PWDREERR	内容	クライアントからのパスワード変更要求を確認用パスワードの入力エラーとする場合に使用します。
	表示画面	確認用パスワード入力エラーページ (pwdchg_repaserr.html)
EX_DFW_PWDPCYERR	内容	クライアントからのパスワード変更要求をパスワードポリシーエラーとする場合に使用します。
	表示画面	パスワード変更ポリシーエラーページ (pwdchg_policy_err.html)
EX_DFW_PWDVIOERR	内容	クライアントからのパスワード変更要求をパスワード変更不可エラーとする場合に使用します。
	表示画面	パスワード変更権限なしエラーページ (pwdchg_pwdvioerr.html)
EX_DFW_PWDNOLOGIN	内容	クライアントからのパスワード変更要求をログアウト済みエラーとする場合に使用します。
	表示画面	パスワード変更未ログインエラーページ (pwdchg_nologin.html)
EX_DFW_PWDERR	内容	クライアントからのパスワード変更要求をパスワード変更エラーとする場合に使用します。
	表示画面	パスワード変更エラーページ (pwdchg_err.html)
EX_DFW_PWDWARNING	内容	クライアントにパスワード有効期限切れ警告を行う場合に使用します。
	表示画面	パスワード有効期限切れ警告ページ (pwdchg_warning.html)
EX_DFW_SYSERR	内容	クライアントからのリクエストをシステムエラーとする場合に使用します。
	表示画面	システムエラーページ (system_error.html)
EX_DFW_ALIASNO	内容	クライアントからのリクエストをエイリアスなしエラーとする場合に使用します。
	表示画面	エイリアスなしエラーページ (system_server_error.html)
EX_DFW_ALIASBAD	内容	クライアントからのリクエストを未定義エイリアスエラーとする場合に使用します。
	表示画面	未定義エイリアスエラーページ (system_alias_error.html)
EX_DFW_DOWNCERTD	内容	クライアントからのリクエストを認証モジュールダウンエラーとする場合に使用します。
	表示画面	認証モジュールダウンエラーページ (system_cert_error.html)
EX_DFW_DOWNDB	内容	クライアントからのリクエストを認証 DB ダウンエラーとする場合に使用します。
	表示画面	認証 DB ダウンエラーページ (system_ldap_error.html)
EX_DFW_DOWNBKEND	内容	クライアントからのリクエストをバックエンド Web サーバーダウンエラーとする場合に使用します。
	表示画面	バックエンド Web サーバーダウンエラーページ (system_backend_error.html)

制御コード	内容	
EX_DFW_TOUTCERTD	内容	クライアントからのリクエストを認証モジュール受信タイムアウトエラーとする場合に使用します。
	表示画面	認証モジュール受信タイムアウトエラーページ (system_timeout_certd.html)
EX_DFW_TOUTBKEND	内容	クライアントからのリクエストをバックエンド Web サーバー受信タイムアウトエラーとする場合に使用します。
	表示画面	バックエンド Web サーバー受信タイムアウトエラーページ (system_timeout_bkend.html)
EX_DFW_FILTERGE	内容	クライアントからのリクエストを GET フィルターエラーとする場合に使用します。
	表示画面	GET フィルターエラーページ (filter_get_error.html)
EX_DFW_FILTERPOST	内容	クライアントからのリクエストを POST フィルターエラーとする場合に使用します。
	表示画面	POST フィルターエラーページ (filter_post_error.html)
EX_DFW_FILTERHTML	内容	クライアントからのリクエストを HTML フィルターエラーとする場合に使用します。
	表示画面	HTML フィルターエラーページ (filter_html_error.html)
EX_DFW_FILTERHOST	内容	クライアントからのリクエストをホストフィルターエラーとする場合に使用します。
	表示画面	ホストフィルターエラーページ (filter_svr_error.html)
EX_DFW_LOGINTIMEERR	内容	パスワード変更時にフォワーダーへの POST 送信を行うまでの制限時間を越えた場合に使用します。
	表示画面	パスワード変更送信時間制限エラーページ (pwdchg_postlimit_err.html)
EX_DFW_PWDTIMEERR	内容	パスワード変更時にフォワーダーへの POST 送信を行うまでの制限時間を越えた場合に使用します。
	表示画面	パスワード変更送信時間制限エラーページ (pwdchg_postlimit_err.html)
EX_DFW_FILTERREQ 10.0	内容	リクエストを受け付けていないリクエスト種別がリクエストされた場合に使用します。
	表示画面	リクエストフィルターエラーページ (filter_request_error.html)
EX_DFW_DBBUSYERR	内容	認証モジュールでデータベース接続にアクセスが集中して処理ができない場合に使用します。
	表示画面	DB ビジーエラーページ (system_busy_database.html)
EX_DFW_REQACLERR 10.0	内容	認証モジュールへの接続が許可されなかった場合に使用します。
	表示画面	リクエスト ACL エラーページ (request_acl_error.html)

制御コード	内容	
EX_DFW_POSTERR	内容	クライアントからのリクエストを POST データ最大サイズエラーとする場合に使用します。
	表示画面	POST データ最大サイズエラーページ (max_postsize_error.html)
EX_DFW_USREX1	内容	クライアントからのリクエストをユーザー定義エラー 1 とする場合に使用します。
	表示画面	ユーザー定義エラー 1 ページ (usr_ext1.html)
EX_DFW_USREX2	内容	クライアントからのリクエストをユーザー定義エラー 2 とする場合に使用します。
	表示画面	ユーザー定義エラー 2 ページ (usr_ext2.html)
EX_DFW_USREX3	内容	クライアントからのリクエストをユーザー定義エラー 3 とする場合に使用します。
	表示画面	ユーザー定義エラー 3 ページ (usr_ext3.html)
EX_DFW_USREX4	内容	クライアントからのリクエストをユーザー定義エラー 4 とする場合に使用します。
	表示画面	ユーザー定義エラー 4 ページ (usr_ext4.html)
EX_DFW_USREX5	内容	クライアントからのリクエストをユーザー定義エラー 5 とする場合に使用します。
	表示画面	ユーザー定義エラー 5 ページ (usr_ext5.html)
EX_DFW_USREX6	内容	クライアントからのリクエストをユーザー定義エラー 6 とする場合に使用します。
	表示画面	ユーザー定義エラー 6 ページ (usr_ext6.html)

4.1.1 インターフェイス関数内 API

インターフェイス関数内で使用可能な API は以下のとおりです。

API 名	内容
IW_ExDFWChangeProtocol	フォワーダーの動作プロトコルを変更します。
IW_ExDFWGetMethod	フォワーダーにリクエストされたメソッドを取得します。
IW_ExDFWModifyMethod	フォワーダーにリクエストされたメソッドを変更します。
IW_ExDFWGetPostdata	フォワーダーに送信された POST データを取得します。
IW_ExDFWModifyPostdata	フォワーダーに送信された POST データを任意の POST データに変更します。
IW_ExDFWGetBuffer	バックエンド Web サーバーと送受信を行うデータバッファを取得します。
IW_ExDFWModifyBuffer	バックエンド Web サーバーと送受信を行うデータバッファを変更します。テキストデータ用です。
IW_ExDFWModifyBufferLength	バックエンド Web サーバーと送受信を行うデータバッファを変更します。バイナリデータ用です。
IW_ExDFWModifyRequestPath	バックエンド Web サーバーへのリクエストパスを変更します。
IW_ExDFWModifyPathinfo	フォワーダーにリクエストされた URL のエイリアス以降を変更します。
IW_ExDFWGetUid	ログインユーザーのユーザー ID を取得します。
IW_ExDFWGetCertData	認証モジュールと送受信を行うデータバッファを取得します。
IW_ExDFWCtrlCertData	認証モジュールと送受信を行うデータバッファを操作します。

以降ではこれらの API 使用の解説を行います。

IW_ExDFWChangeProtocol

書式 **int IW_ExDFWChangeProtocol(EX_DFW_REQ* req, int kind, int flg)**

機能 フォワーダーの動作プロトコルを変更します。
 フォワーダー設定ファイルでプロトコルを SSL に変更した場合
 (DFW_PROTOCOL=1) でも、本 API でプロトコルを HTTP に戻すことが可能です。

引数 **req** : UserExit ルーチンリクエスト構造体を使用します。インターフェイス関数より渡された req を設定してください。

kind : UserExit ルーチン位置情報フラグを使用します。インターフェイス関数の引数として渡された kind を設定してください。

flg : フォワーダーの動作プロトコルを設定するフラグです。以下の値を設定します。
 EX_DFW_HTTP : プロトコルを HTTP に設定します。
 EX_DFW_HTTPS : プロトコルを SSL に設定します。

戻り値 以下の値が返されます。
 EX_DFW_SUCCESS : 正常終了
 EX_DFW_ERROR : 異常終了
 EX_DFW_BADKIND : API を使用できないエントリー、または不正な kind 値

制限事項 インターフェイス関数の引数 kind が EX_DFW_KIND_REQ の場合のみ使用可能です。

IW_ExDFWGetMethod

書式 **char *IW_ExDFWGetMethod (EX_DFW_REQ* req, int kind)**

機能 フォワーダーにリクエストされたメソッドを取得します。

引数 **req** : UserExit ルーチンリクエスト構造体を使用します。インターフェイス関数より渡された req を設定してください。

kind : UserExit ルーチン位置情報フラグを使用します。インターフェイス関数の引数として渡された kind を設定してください。

戻り値 以下の値が返されます。
 NULL 以外 : メソッドを表す文字列のポインター。編集が必要な場合はデータを別の領域へコピーしてから使用してください。
 NULL : 異常終了、または API が使用できないエントリー

制限事項 インターフェイス関数の引数 kind が EX_DFW_KIND_REQ の場合のみ使用可能です。

戻り値が指す文字列の変更は禁止です。変更した場合の動作は保証されません。

IW_ExDFWModifyMethod

書式 **int IW_ExDFWModifyMethod (EX_DFW_REQ* req, int kind, int method)**

機能 フォワーダーにリクエストされたメソッドを変更します。

引数 **req** : UserExit ルーチンリクエスト構造体を使用します。インターフェイス関数より渡された req を設定してください。

kind : UserExit ルーチン位置情報フラグを使用します。インターフェイス関数の引数として渡された kind を設定してください。

method : 変更するメソッドを指定するフラグです。以下の値を設定します。

EX_DFW_METHOD_GET : メソッド GET

EX_DFW_METHOD_POST : メソッド POST

戻り値 以下の値が返されます。

EX_DFW_SUCCESS : 正常終了

EX_DFW_ERROR : 異常終了

EX_DFW_BADKIND : API が使用できないエントリー、または不正な kind 値

制限事項 インターフェイス関数の引数 kind が EX_DFW_KIND_REQ の場合のみ使用可能です。

IW_ExDFWGetPostdata

書式 **char* IW_ExDFWGetPostdata (EX_DFW_REQ* req, int kind)**

機能 フォワーダーに送信された POST データを取得します。
POST データは、ブラウザーからの送信時に URL エンコードされている場合は、本 API でもエンコードされた状態で取得されます。

引数 **req** : UserExit ルーチンリクエスト構造体を使用します。インターフェイス関数より渡された req を設定してください。

kind : UserExit ルーチン位置情報フラグを使用します。インターフェイス関数の引数として渡された kind を設定してください。

戻り値 以下の値が返されます。
 NULL 以外 : POST データが格納されているバッファのポインター。編集が必要な場合はデータを別の領域へコピーしてから使用してください。
 NULL : 異常終了、または API が使用できないエントリー

制限事項 インターフェイス関数の引数 kind が EX_DFW_KIND_REQ の場合のみ使用可能です。

戻り値が指す文字列の変更は禁止です。変更した場合の動作は保証されません。

FORCELOGIN_ENC 項目を使用して暗号化を行っている場合、強制ログインページから送信された POST データのユーザー ID とパスワード部分が暗号化されることに注意してください。

IW_ExDFWModifyPostdata

書式 **int IW_ExDFWModifyPostdata (EX_DFW_REQ* req, int kind, char* buf)**

機能 フォワーダーに送信された POST データを任意の POST データに変更します。
変更する POST データはブラウザからの送信時に URL エンコードされている場合は、本 API 使用時にも URL エンコードされた状態でなければなりません。
POST データの変更後は Content-Length ヘッダーが自動的に書き換えられます。

引数 **req** : UserExit ルーチンリクエスト構造体を使用します。インターフェイス関数より渡された req を設定してください。

kind : UserExit ルーチン位置情報フラグを使用します。インターフェイス関数の引数として渡された kind を設定してください。

buf : 変更する POST データが格納されているバッファのポインターを設定します。本 API を使用した後はバッファを破棄しても問題ありません。

戻り値 以下の値が返されます。
EX_DFW_SUCCESS : 正常終了
EX_DFW_ERROR : 異常終了
EX_DFW_BADKIND : API が使用できないエントリ、または不正な kind 値

制限事項 インターフェイス関数の引数 kind が EX_DFW_KIND_REQ の場合のみ使用可能です。

IW_ExDFWGetBuffer

書式 **char* IW_ExDFWGetBuffer (EX_DFW_REQ* req, int kind)**

機能 バックエンド Web サーバーへ送信するリクエストデータ、バックエンド Web サーバーより受信したレスポンスデータまたはブラウザへ出力する直前のコンテンツを取得します。

取得可能なデータは引数 **kind** により以下のものが取得されます。

EX_DFW_KIND_SEND : バックエンド Web サーバーへ送信するリクエストデータ

EX_DFW_KIND_RECV : バックエンド Web サーバーより受信したレスポンスデータ

EX_DFW_KIND_END : ブラウザーへ出力する直前のコンテンツ

引数 **req** : UserExit ルーチンリクエスト構造体を使用します。インターフェイス関数より渡された **req** を設定してください。

kind : UserExit ルーチン位置情報フラグを使用します。インターフェイス関数の引数として渡された **kind** を設定してください。

戻り値 以下の値が返されます。

NULL 以外 : データが格納されているバッファのポインター。編集が必要な場合はデータを別の領域へコピーしてから使用してください。

NULL : 異常終了、または API が使用できないエントリー

制限事項 インターフェイス関数の引数 **kind** が EX_DFW_KIND_SEND、EX_DFW_KIND_RECV、EX_DFW_KIND_END の場合のみ使用可能です。

戻り値が指す文字列の変更は禁止です。変更した場合の動作は保証されません。

IW_ExDFWModifyBuffer

書式 **int IW_ExDFWModifyBuffer (EX_DFW_REQ* req, int kind, char* buf)**

機能 バックエンド Web サーバーへ送信するリクエストデータ、バックエンド Web サーバーより受信したレスポンスデータまたはブラウザへ出力する直前のコンテンツを任意のデータへ変更します。

変更可能なデータは引数 **kind** により指定されます。

EX_DFW_KIND_SEND : バックエンド Web サーバーへ送信するリクエストデータ

EX_DFW_KIND_RECV : バックエンド Web サーバーより受信したレスポンスデータ

EX_DFW_KIND_END : ブラウザーへ出力する直前のコンテンツ

引数 **req** : UserExit ルーチンリクエスト構造体を使用します。インターフェイス関数より渡された **req** を設定してください。

kind : UserExit ルーチン位置情報フラグを使用します。インターフェイス関数の引数として渡された **kind** を設定してください。

buf : 変更する POST データが格納されているバッファのポインタを設定します。本 API を使用した後はバッファを破棄しても問題ありません。

戻り値 以下の値が返されます。

EX_DFW_SUCCESS : 正常終了

EX_DFW_ERROR : 異常終了、API が使用できないエントリ、または不正な **kind** 値

制限事項 インターフェイス関数の引数 **kind** が EX_DFW_KIND_SEND、EX_DFW_KIND_RECV、EX_DFW_KIND_END の場合のみ使用可能です。

リクエストデータおよびレスポンスデータを変更する場合は、そのデータを正しいフォーマットに設定してください。リクエストデータに対してレスポンスデータフォーマットのバッファへの変更などを行った場合の動作は保証されません。

※ 変更するバッファにバイナリーデータが含まれる場合は、本 API を使用せず、後述の IW_ExDFWModifyBufferLength0 を使用してください。

IW_ExDFWModifyBufferLength

書式 **int IW_ExDFWModifyBufferLength (EX_DFW_REQ* req, int kind, char* buf, int len)**

機能 バックエンド Web サーバーへ送信するリクエストデータ、バックエンド Web サーバーより受信したレスポンスデータまたはブラウザへ出力する直前のコンテンツを任意のデータへ変更します。

IW_ExDFWModifyBuffer() のバイナリーデータ対応版です。

変更可能なデータは引数 kind により指定されます。

EX_DFW_KIND_SEND : バックエンド Web サーバーへ送信するリクエストデータ

EX_DFW_KIND_RECV : バックエンド Web サーバーより受信したレスポンスデータ

EX_DFW_KIND_END : ブラウザーへ出力する直前のコンテンツ

引数 **req** : UserExit ルーチンリクエスト構造体を使用します。インターフェイス関数より渡された req を設定してください。

kind : UserExit ルーチン位置情報フラグを使用します。インターフェイス関数の引数として渡された kind を設定してください。

buf : 変更する POST データが格納されているバッファのポインターを設定します。本 API を使用した後はバッファを破棄しても問題ありません。

len : 変更を行うデータバッファの長さを指定します。

戻り値 以下の値が返されます。

EX_DFW_SUCCESS : 正常終了

EX_DFW_ERROR : 異常終了

EX_DFW_BADKIND : API が使用できないエントリ、または不正な kind 値

制限事項 インターフェイス関数の引数 kind が EX_DFW_KIND_SEND、EX_DFW_KIND_RECV、EX_DFW_KIND_END の場合のみ使用可能です。

リクエストデータおよびレスポンスデータを変更する場合は、そのデータを正しいフォーマットに設定してください。リクエストデータに対してレスポンスデータフォーマットのバッファへの変更などを行った場合の動作は保証されません。

IW_ExDFWModifyRequestPath

書式 **int IW_ExDFWModifyRequestPath(EX_DFW_REQ* req, int kind, char *path)**

機能 バックエンド Web サーバーに対するコンテンツのパスを変更します。

引数 **req** : UserExit ルーチンリクエスト構造体を使用します。インターフェイス関数より渡された req を設定してください。

kind : UserExit ルーチン位置情報フラグを使用します。インターフェイス関数の引数として渡された kind を設定してください。

path : 新たに設定するコンテンツへのパスを設定します。

戻り値 以下の値が返されます。
 EX_DFW_SUCCESS : 正常終了
 EX_DFW_ERROR : 異常終了

制限事項 本 API にてコンテンツへのパスを変更した場合、変更位置により以下のように動作が変わります。

位置情報	変更後の動作
EX_DFW_KIND_REQ	・変更されたコンテンツパスにてアクセス制御が行われます。 ・変更したパスより Set-Cookie ヘッダーが path 属性なしで送信された場合、変更前のパスに対してセットされます。 ・認証モジュールのアクセスログには変更されたパスが記録されます。 ・フォワーダーのアクセスログには変更されたパスが記録されます。
EX_DFW_KIND_LOGIN_SEND	・変更されたコンテンツパスにログイン後リダイレクトされます。
EX_DFW_KIND_LOGIN_RECV	・変更されたコンテンツパスにログイン後リダイレクトされます。
EX_DFW_KIND_ALOGIN_SEND	・変更は可能ですが動作に影響しません。
EX_DFW_KIND_ALOGIN_RECV	・変更は可能ですが動作に影響しません。

位置情報	変更後の動作
EX_DFW_KIND_ACC_SEND	・ 変更される前のコンテンツパスにてアクセス制御が行われますが、実際に取得されるコンテンツは変更されたパスになります。 ・ 変更したパスより Set-Cookie ヘッダーが path 属性なしで送信された場合、変更前のパスに対してセットされます。 ・ 認証モジュールのアクセスログには変更される前のパスが記録されます。 ・ フォワーダーのアクセスログには変更されたパスが記録されます。
EX_DFW_KIND_ACC_RECV	・ 変更される前のコンテンツパスにてアクセス制御が行われますが、実際に取得されるコンテンツは変更されたパスになります。 ・ 変更したパスより Set-Cookie ヘッダーが path 属性なしで送信された場合、変更前のパスに対してセットされます。 ・ 認証モジュールのアクセスログには変更される前のパスが記録されます。 ・ フォワーダーのアクセスログには変更されたパスが記録されます。
EX_DFW_KIND_LOGOUT_SEND	・ SESSION=0 リクエストパスがないため EX_DFW_ERROR が返ります。 ・ SESSION=1 リクエストパスが変更されますが動作に影響しません。
EX_DFW_KIND_LOGOUT_RECV	・ SESSION=0 リクエストパスがないため EX_DFW_ERROR が返ります。 ・ SESSION=1 リクエストパスが変更されますが動作に影響しません。
EX_DFW_KIND_PWDCHG_SEND	・ SESSION=0 リクエストパスがないため EX_DFW_ERROR が返ります。 ・ SESSION=1 リクエストパスが変更されますが動作に影響しません。
EX_DFW_KIND_PWDCHG_RECV	・ SESSION=0 リクエストパスがないため EX_DFW_ERROR が返ります。 ・ SESSION=1 リクエストパスが変更されますが動作に影響しません。
EX_DFW_KIND_SEND	・ 変更される前のコンテンツパスにてアクセス制御が行われますが、実際に取得されるコンテンツは変更されたパスになります。 ・ 変更したパスより Set-Cookie ヘッダーが path 属性なしで送信された場合、変更前のパスに対してセットされます。 ・ 認証モジュールのアクセスログには変更される前のパスが記録されます。 ・ フォワーダーのアクセスログには変更されたパスが記録されます。
EX_DFW_KIND_RECV	・ フォワーダーのアクセスログには変更されたパスが記録されます。
EX_DFW_KIND_END	・ フォワーダーのアクセスログには変更されたパスが記録されます。

IW_ExDFWModifyPathinfo

書式 **int IW_ExDFWModifyPathinfo(EX_DFW_REQ* req, int kind, char *pathinfo)**

機能 フォワーダーにリクエストされた URL のエイリアス以降部分を変更します。

引数 **req** : UserExit ルーチンリクエスト構造体を使用します。インターフェイス関数より渡された req を設定してください。

kind : UserExit ルーチン位置情報フラグを使用します。インターフェイス関数の引数として渡された kind を設定してください。

pathinfo : 新たに設定する URL のエイリアス以降を設定します。NULL を指定した場合はエイリアス以降を削除することができます。

戻り値 以下の値が返されます。
 EX_DFW_SUCCESS : 正常終了
 EX_DFW_ERROR : 異常終了
 EX_DFW_BADKIND : API が使用できないエントリー、または不正な kind 値

制限事項 インターフェイス関数の引数 kind が EX_DFW_KIND_REQ の場合のみ使用可能です。

IW_ExDFWGetUid

書式 **char *IW_ExDFWGetUid(EX_DFW_REQ* req, int kind, int flg)**

機能 現在アクセスしているユーザーのユーザー ID を取得します。

本 API は 8.0 以降のバージョンにて使用できます。

引数 **req** : UserExit ルーチンリクエスト構造体を使用します。インターフェイス関数より渡された req を設定してください。

kind : UserExit ルーチン位置情報フラグを使用します。インターフェイス関数の引数として渡された kind を設定してください。

flg : 取得ユーザー ID 情報フラグを使用します。以下の値を設定します。

EX_DFW_UID_LOGIN : ログインしているユーザーのユーザー ID を取得

EX_DFW_UID_INPUT : ログイン時に入力したユーザー ID を取得

戻り値 以下の値が返されます。

NULL 以外 : ユーザー ID が格納されているバッファのポインター (このバッファはリードオンリーとなります)。編集が必要な場合はデータを別の領域へコピーしてから使用してください

NULL : ユーザー ID 未保持、異常終了または API が使用できないエントリー

制限事項 ログインユーザー ID (既にログイン成功している現ユーザーのユーザー ID) はインターフェイス関数の引数 kind が EX_DFW_KIND_END の場合に取得可能であり、アクセスコントロール時のみ EX_DFW_KIND_SEND、EX_DFW_KIND_RECV でも使用可能です。

認証モジュール通信時の kind では取得はできません。

クライアント証明書を使用している場合はクライアント証明書のユーザー ID が取得されます。

入力ユーザー ID (ログイン画面などよりクライアントから入力されたユーザー ID、未ログイン状態である) はログイン処理時で、かつインターフェイス関数の引数 kind が EX_DFW_KIND_END の場合のみ、取得可能です。

ログアウト処理時は、本 API でユーザー ID は取得できません。

ログインユーザー ID は認証 DB に格納されたユーザー ID そのままで取得されますが、認証モジュールにてログイン時に UserExit ルーチンで変更した場合は、変更後のユーザー ID が取得されます。

入力ユーザー ID の最大サイズは 64 文字です。64 文字以上のユーザー ID が入力された場合は、ログインエラーとなるためユーザー ID の取得はできません。

入力ユーザー ID は、URL エンコードされて送信された場合は、デコードされて取得されます。また、文字コードは入力された文字コードのままで取得されます。

IW_ExDFWGetCertData

書式 `char *IW_ExDFWGetCertData( EX_DFW_REQ* req, int kind, char *name, int target )`

機能 ICP 2.0 で認証モジュールへ送信するリクエストデータ、認証モジュールより受信したレスポンスデータに含まれる **name: value** 形式データの **value** を取得します。
取得可能なデータは引数 **kind** により以下のバッファが対象となります。
 EX_DFW_KIND_*_SEND : 認証モジュールへ送信するリクエストデータ
 EX_DFW_KIND_*_RECV : 認証モジュールより受信したレスポンスデータ

引数 **req** : UserExit ルーチンリクエスト構造体を使用します。インターフェイス関数より渡された **req** を設定してください。

kind : UserExit ルーチン位置情報フラグを使用します。インターフェイス関数の引数として渡された **kind** を設定してください。

name : 取得データ名を指定します。大文字小文字は意識しません。

target : データの取得先を指定します。 以下の値を設定します。

EX_DFW_ICP_HEADER : ICP2.0 ヘッダー部

EX_DFW_ICP_BODY : ICP2.0 ボディ部

戻り値 以下の値が返されます。
 NULL 以外 : データが格納されているバッファのポインター（このバッファはリードオンリーとなります）。編集が必要な場合はデータを別の領域へコピーしてから使用してください。
 NULL : ICP 1.0 使用時、指定したデータ名が存在しない、異常終了または API が使用できないエントリー

制限事項 インターフェイス関数の引数 **kind** が **EX_DFW_KIND_LOGIN_SEND**、**EX_DFW_KIND_LOGIN_RECV**、**EX_DFW_KIND_LOGOUT_SEND**、**EX_DFW_KIND_LOGOUT_RECV**、**EX_DFW_KIND_ACC_SEND**、**EX_DFW_KIND_ACC_RECV**、**EX_DFW_KIND_PWDCHG_SEND**、**EX_DFW_KIND_PWDCHG_RECV**、**EX_DFW_KIND_ALOGIN_SEND**、**EX_DFW_KIND_ALOGIN_RECV** の場合のみ使用可能です。

ICP 2.0 を使用している場合のみ使用可能です。

ICP 2.0 ヘッダー部に含まれるステータス行は取得できません。

IW_ExDFWCtrlCertData

書式 **int IW_ExDFWCtrlCertData(EX_DFW_REQ* req, int kind, char *name, char *value, int target, int mode)**

機能 認証モジュールへ送信するリクエストデータ、認証モジュールより受信したレスポンスデータを制御します。

変更可能なデータは引数 **kind** により以下のものが変更されます。

EX_DFW_KIND_*_SEND : 認証モジュールへ送信するリクエストデータ

EX_DFW_KIND_*_RECV : 認証モジュールより受信したレスポンスデータ

引数 **req** : UserExit ルーチンリクエスト構造体を使用します。インターフェイス関数より渡された **req** を設定してください。

kind : UserExit ルーチン位置情報フラグを使用します。インターフェイス関数の引数として渡された **kind** を設定してください。

name : 対象データ名 (name: value の name) を指定します。大文字小文字は意識しません。

value : 対象データ値 (name: value の value) を指定します。削除処理の場合は「NULL」を指定してください。

target : データの制御先を指定します。以下の値を設定します。

EX_DFW_ICP_HEADER : ICP2.0 ヘッダー部

EX_DFW_ICP_BODY : ICP2.0 ボディ部

mode : データの制御方法を指定します。以下の値を設定します。

EX_DFW_ICP_ADD : name: value を最後に追加する処理

EX_DFW_ICP_MOD : name: value の value を変更する処理

EX_DFW_ICP_DEL : name: value 全体の削除する処理

戻り値 以下の値が返されます。

EX_DFW_SUCCESS : 正常終了

EX_DFW_ERROR : 異常終了

EX_DFW_BADKIND : API が使用できないエントリー、または不正な **kind** 値

EX_DFW_NOTFOUND : 対象が見つからない (変更処理、削除処理)

制限事項 インターフェイス関数の引数 kind が EX_DFW_KIND_LOGIN_SEND、EX_DFW_KIND_LOGIN_RECV、EX_DFW_KIND_LOGOUT_SEND、EX_DFW_KIND_LOGOUT_RECV、EX_DFW_KIND_ACC_SEND、EX_DFW_KIND_ACC_RECV、EX_DFW_KIND_PWDCHG_SEND、EX_DFW_KIND_PWDCHG_RECV、EX_DFW_KIND_ALOGIN_SEND、EX_DFW_KIND_ALOGIN_RECV の場合のみ使用可能です。

ICP2.0 を使用している場合のみ使用可能です。

ICP2.0 ヘッダー部に含まれるステータス行を制御することはできません。

認証モジュールへ送信するデータにバイナリーが含まれている場合は正しく通信を行うことができなくなります。

追加および変更処理で value に””（値なし）を指定した場合、データが「name:」のように「値なし」の状態になります。

4.2 認証モジュール用 UserExit ルーチン仕様

認証モジュール用 UserExit ルーチンでは以下のものが定義されています。

- ・ 認証モジュールよりコールされるインターフェイス関数
- ・ インターフェイス関数の戻り値としてクライアントに表示する画面を制御するコード
- ・ インターフェイス関数内で使用可能な API

次ページよりこれらの解説を行います。

IW_ExInterFace ⑩.⑩

認証モジュール用 UserExit ルーチンのインターフェイス関数の仕様は以下のとおりです。

書式 **int IW_ExInterFace(int kind, char* userid, char* password, char* sessionid, char* data, EX_CS_REQ *req)**

引数 **kind** : UserExit ルーチンがコールされた位置情報が格納されます。以下の値が設定されます。

EX_LOGIN_B	: ユーザー認証前
EX_LOGIN_A	: ユーザー認証後
EX_LOGIN_A_ERR	: ユーザー認証エラー後
EX_ACCESS_B	: アクセス権限チェック前
EX_ACCESS_A	: アクセス権限チェック後
EX_ACCESS_A_ERR	: アクセス権限チェックエラー後
EX_LOGOUT_B	: ログアウト前
EX_LOGOUT_A	: ログアウト後
EX_LOGOUT_A_ERR	: ログアウトエラー後
EX_PASSWD_B	: パスワード変更前
EX_PASSWD_A	: パスワード変更後
EX_PASSWD_A_ERR	: パスワード変更エラー後

userid : ブラウザーよりログイン時に入力されたユーザー ID が格納されます。この値はユーザー認証前 (EX_LOGIN_B) のみ編集可能です。

UserExit ルーチン内で編集を行った場合は、編集されたユーザー ID にて認証が行われます。またバックエンド Web サーバーへも、編集されたユーザー ID が通知されます。ユーザー ID を編集する場合は、64 バイト以内としてください。

password : ブラウザーよりログイン時に入力されたパスワードが格納されます。この値は、ユーザー認証前 (EX_LOGIN_B) のみ編集可能です。

UserExit ルーチン内で編集を行った場合は、編集されたパスワードにて認証が行われます。またバックエンド Web サーバーへも、編集されたパスワードが通知します。パスワードを編集する場合は、128 バイト以内としてください。

sessionid : IceWall SSO 用のセッション ID が格納されます。この引数は編集ができません。

data : UserExit ルーチンがコールされた位置情報に対する付加情報が格納されます。UserExit ルーチンがコールされた位置により、以下の内容が設定されます。各処理にてエラーとなった後でコールされた場合にはエラーコードが追加されます。

- ・ ユーザー認証前、ユーザー認証後 (ユーザー認証成功)

N_UID : 通常ログイン

F_UID : 強制ログイン (二重ログイン不可かつ排他ログインの場合)

N_CERT: クライアント証明書による認証 (通常ログイン)

F_CERT: クライアント証明書による認証 (強制ログイン)

- ・ ユーザー認証エラー後

N_UID,ErrCode : 通常ログインでエラーとなった理由がエラーコードとして付加されます。

F_UID,ErrCode : 強制ログイン (二重ログイン不可かつ排他ログインの場合) でエラーとなった理由がエラーコードとして付加されます。

N_CERT,ErrCode: クライアント証明書による認証 (通常ログイン) でエラーとなった理由がエラーコードとして付加されます。

F_CERT,ErrCode: クライアント証明書による認証 (強制ログイン) でエラーとなった理由がエラーコードとして付加されます。

- ・ アクセス権限チェック前、アクセス権限チェック後

アクセスチェックを行う URL

- ・ アクセス権限チェックエラー後

アクセスチェックを行った URL,ErrCode

- ・ ログアウト前、ログアウト後

MANUAL: ユーザー操作のログアウト

AUTO : セッションタイムアウトのログアウト

FORCED : 同一ユーザー ID のログインによる強制ログアウト

FORCEALL: 全ユーザー強制ログアウト **10.0**

- ・ ログアウトエラー後

MANUAL,ErrCode : ユーザー操作のログアウトでエラーとなった理由がエラーコードとして付加されます。

AUTO,ErrCode : セッションタイムアウトのログアウトでエラーとなった理由がエラーコードとして付加されます。

FORCED,ErrCode : 同一ユーザー ID のログインによる強制ログアウトでエラーとなった理由がエラーコードとして付加されます。

FORCEALL,ErrCode: 全ユーザー強制ログアウトでエラーとなった理由がエラーコードとして付加されます。 **10.0**

- ・ パスワード変更前

パスワード変更でブラウザーより入力された新パスワード

例: 新パスワードが “passwd” の場合

“passwd”

- ・ パスワード変更後

パスワード変更でブラウザーより入力された新パスワード

例: 新パスワードが “passwd” の場合

“passwd”

- ・パスワード変更エラー後
パスワード変更でブラウザーより入力された新パスワードとエラーとなった理由がエラーコードとして付加されます。
例：新パスワードが “passwd” の場合
“passwd,14”

req :UserExit ルーチンリクエスト構造体です。この構造体に認証モジュールの情報が格納されています。この構造体の情報を操作するには API を使用します。

戻り値 クライアントに表示するページを制御するコードを返します。
使用可能なコードについては次節を参照してください。

制限事項 本インターフェイス関数で渡される引数は `userid`、`password` 以外編集できません。これら以外の引数を編集した場合の動作は保証されません。

バージョン 10.0 以前に作成された UserExit ルーチンでログアウト時の付加情報を判断して開発している場合は、引数が追加されておりますので注意する必要があります。 **10.0**

認証モジュール UserExit 用クライアント画面制御コード

認証モジュール用 UserExit ルーチンにて使用可能なクライアント画面制御コードは以下のとおりです。表示される画面については「IceWall SSO 標準 HTML カスタマイズガイド」および「IceWall SSO リファレンスマニュアル」を参照してください。

制御コード	内容	
EXR_OK	内容	UserExit ルーチン内の処理終了後に認証モジュールの処理を続行する場合に使用します。
	表示画面	この制御コードを使用した場合に表示される画面は、その後の認証モジュールおよびフォワーダーの動作に依存します。
	使用可能位置	すべての位置で使用可能
EXR_UIDERR	内容	クライアントからのログイン要求をユーザー ID エラーとする場合に使用します。
	表示画面	ユーザー ID エラーページ (login_userid_error.html)
	使用可能位置	EX_LOGIN_B、EX_LOGIN_A、EX_LOGIN_A_ERR
EXR_PWDERR	内容	クライアントからのログイン要求をパスワードエラーとする場合に使用します。
	表示画面	パスワードエラーページ (login_pwd_error.html)
	使用可能位置	EX_LOGIN_B、EX_LOGIN_A、EX_LOGIN_A_ERR
EXR_PWDCHG	内容	クライアントからのログイン要求またはアクセス要求に対してパスワード変更要求を行う場合に使用します。
	表示画面	パスワード変更ページ (pwdchg.html)
	使用可能位置	EX_LOGIN_A、EX_ACCESS_B、EX_ACCESS_A、EX_ACCESS_A_ERR
EXR_USRLOCK	内容	クライアントからのログイン要求をアカウントロックエラーとする場合に使用します。
	表示画面	アカウントロックエラーページ (login_lock_error.html)
	使用可能位置	EX_LOGIN_B、EX_LOGIN_A、EX_LOGIN_A_ERR
EXR_LOGINNO	内容	クライアントからのログイン要求をログイン不可エラーとする場合に使用します。
	表示画面	ログインストップエラーページ (login_stop.html)
	使用可能位置	EX_LOGIN_B、EX_LOGIN_A、EX_LOGIN_A_ERR

制御コード	内容	
EXR_DLOGIN	内容	クライアントからのログイン要求に対して強制ログイン要求を行わせる場合に使用します。
	表示画面	強制ログインページ (login_force.html)
	使用可能位置	EX_LOGIN_B、EX_LOGIN_A、EX_LOGIN_A_ERR
EXR_GRPERR	内容	クライアントからのログイン要求を所属グループなしエラーとする場合に使用します。
	表示画面	所属グループなしエラーページ (login_group_error.html)
	使用可能位置	EX_LOGIN_B、EX_LOGIN_A、EX_LOGIN_A_ERR
EXR_PLOGIN	内容	クライアントからのユーザー ID によるログイン要求に対して初回認証済みエラーとする場合に使用します。この制御コードを使用する場合には、別途「IceWall SSO Client Certificates Option」が必要となります。
	表示画面	初回認証済みエラーページ (login_cert_error.html)
	使用可能位置	EX_LOGIN_B*1、EX_LOGIN_A*1、EX_LOGIN_A_ERR*1
EXR_PWDWARN	内容	パスワード有効期限切れの警告を通知する場合に使用します。
	表示画面	パスワード有効期限切れ警告ページ (pwdchg_warning.html)
	使用可能位置	EX_LOGIN_A
EXR_SNOERR	内容	クライアントからのクライアント証明書によるログイン要求に対して証明書シリアル No エラーとする場合に使用します。この制御コードを使用する場合には、別途「IceWall SSO Client Certificates Option」が必要となります。
	表示画面	証明書シリアル No エラーページ (login_error.html)
	使用可能位置	EX_LOGIN_B*1、EX_LOGIN_A*1、EX_LOGIN_A_ERR*1
EXR_RLOGIN	内容	クライアントからのアクセス要求に対して再ログイン要求を行う場合に使用します。
	表示画面	ログインページ (login.html、クライアント証明書使用時 : login_cert.html)
	使用可能位置	EX_ACCESS_B、EX_ACCESS_A、EX_ACCESS_A_ERR
EXR_ACCERR	内容	クライアントからのアクセス要求に対してアクセス権限エラーとする場合に使用します。
	表示画面	アクセス権限エラーページ (access_error.html)
	使用可能位置	EX_ACCESS_B、EX_ACCESS_A、EX_ACCESS_A_ERR

制御コード	内容	
EXR_LOGOUTNG	内容	ログアウト後エラーとなった場合に付加情報として追加されるエラーコードです。このコードは制御コードとして使用できません。
	表示画面	—
	使用可能位置	—
EXR_PWDPOERR	内容	クライアントからのパスワード変更要求をパスワードポリシーエラーとする場合に使用します。
	表示画面	パスワードポリシーエラーページ (pwdchg_policy_error.html)
	使用可能位置	EX_PASSWD_B、EX_PASSWD_A、EX_PASSWD_A_ERR
EXR_PWDVIOERR	内容	クライアントからのパスワード変更要求をパスワード変更権限なしエラーとする場合に使用します。
	表示画面	パスワード変更権限なしエラー画面 (pwdchg_vio_error.html)
	使用可能位置	EX_PASSWD_B、EX_PASSWD_A、EX_PASSWD_A_ERR
EXR_PWDVIOERR	内容	クライアントからのログイン要求またはパスワード変更要求に対して認証 DB ダウンエラーとする場合に使用します。
	表示画面	認証 DB ダウンエラーページ (system_ldap_error.html)
	使用可能位置	EX_LOGIN_B、EX_LOGIN_A、EX_LOGIN_A_ERREX_PASSWORD_B、EX_PASSWORD_A、EX_PASSWORD_A_ERR
EXR_SYSERR	内容	クライアントからのログイン要求、アクセス要求およびパスワード変更要求に対してシステムエラーとする場合に使用します。
	表示画面	システムエラー画面 (system_error.html)
	使用可能位置	EX_LOGIN_B、EX_LOGIN_A、EX_LOGIN_A_ERR、EX_ACCESS_B、EX_ACCESS_A、EX_ACCESS_A_ERR、EX_PASSWORD_B、EX_PASSWORD_A、EX_PASSWORD_A_ERR
EXR_LOGINLMT 10.0	内容	クライアントからのログイン要求に対してログイン上限数エラーとする場合に使用します。
	表示画面	ログイン上限数エラー画面 (login_limit_error.html)
	使用可能位置	EX_LOGIN_B、EX_LOGIN_A、EX_LOGIN_A_ERR

制御コード	内容	
EXR_ACLREQ (10.0)	内容	リクエスト権限なしエラーとする場合に使用します。
	表示画面	リクエスト権限なしエラー画面 (request_acl_error.html)
	使用可能位置	EX_LOGIN_B、EX_LOGIN_A、EX_LOGIN_A_ERR、EX_ACCESS_B、EX_ACCESS_A、EX_ACCESS_A_ERR、EX_PASSWORD_B、EX_PASSWORD_A、EX_PASSWORD_A_ERR
EXR_SIDERR (10.0)	内容	セッション ID 生成エラーとする場合に使用します。
	表示画面	システムエラー画面 (system_error.html)
	使用可能位置	EX_LOGIN_B、EX_LOGIN_A、EX_LOGIN_A_ERR
EXR_DBBUSY (10.0)	内容	DB ビジーエラーとする場合に使用します。
	表示画面	DB ビジーエラー画面 (system_busy_database.html)
	使用可能位置	EX_LOGIN_B、EX_LOGIN_A、EX_LOGIN_A_ERR、EX_PASSWORD_B、EX_PASSWORD_A、EX_PASSWORD_A_ERR
EXR_PWDMINLEN (10.0)	内容	最小パスワード長エラーとする場合に使用します。
	表示画面	パスワードポリシーエラーページ (pwdchg_policy_error.html)
	使用可能位置	EX_PASSWORD_B、EX_PASSWORD_A、EX_PASSWORD_A_ERR
EXR_PWDMAXLEN (10.0)	内容	最大パスワード長エラーとする場合に使用します。
	表示画面	パスワードポリシーエラーページ (pwdchg_policy_error.html)
	使用可能位置	EX_PASSWORD_B、EX_PASSWORD_A、EX_PASSWORD_A_ERR
EXR_PWDALPHANUM (10.0)	内容	パスワード文字種エラーとする場合に使用します。
	表示画面	パスワードポリシーエラーページ (pwdchg_policy_error.html)
	使用可能位置	EX_PASSWORD_B、EX_PASSWORD_A、EX_PASSWORD_A_ERR

制御コード	内容	
EXR_PWDSAMEPASS 10.0	内容	ユーザー ID 同一パスワードエラーとする場合に使用します。
	表示画面	パスワードポリシーエラーページ (pwdchg_policy_error.html)
	使用可能位置	EX_PASSWD_B、EX_PASSWD_A、EX_PASSWD_A_ERR
EXR_USREX1	内容	クライアントからのログイン要求、アクセス要求およびパスワード変更要求に対してユーザー定義エラー 1 とする場合に使用します。
	表示画面	ユーザー定義エラー 1 画面 (usr_ext1.html)
	使用可能位置	EX_LOGIN_B、EX_LOGIN_A、EX_LOGIN_A_ERR、EX_ACCESS_B、EX_ACCESS_A、EX_ACCESS_A_ERR、EX_PASSWORD_B、EX_PASSWORD_A、EX_PASSWORD_A_ERR
EXR_USREX2	内容	クライアントからのログイン要求、アクセス要求およびパスワード変更要求に対してユーザー定義エラー 2 とする場合に使用します。
	表示画面	ユーザー定義エラー 2 画面 (usr_ext2.html)
	使用可能位置	EX_LOGIN_B、EX_LOGIN_A、EX_LOGIN_A_ERR、EX_ACCESS_B、EX_ACCESS_A、EX_ACCESS_A_ERR、EX_PASSWORD_B、EX_PASSWORD_A、EX_PASSWORD_A_ERR
EXR_USREX3	内容	クライアントからのログイン要求、アクセス要求およびパスワード変更要求に対してユーザー定義エラー 3 とする場合に使用します。
	表示画面	ユーザー定義エラー 3 画面 (usr_ext3.html)
	使用可能位置	EX_LOGIN_B、EX_LOGIN_A、EX_LOGIN_A_ERR、EX_ACCESS_B、EX_ACCESS_A、EX_ACCESS_A_ERR、EX_PASSWORD_B、EX_PASSWORD_A、EX_PASSWORD_A_ERR

※ 1 IceWall SSO Client Certification Option が導入されていない場合は EXR_SYSERR となります。

なお、各エントリーにて使用可能な制御コード以外を使用した場合は次のように値が変更されますので注意してください。

エントリー	変更される値	表示画面
ログイン前 ログイン後 アクセス制御前 アクセス制御後	EXR_SYSERR	システムエラーページ (system_error.html)
パスワード変更前 パスワード変更後	EXR_SYSERR	パスワード変更エラーページ (pwdchg_error.html)
ログインエラー後 アクセス制御エラー後 ログアウト前 ログアウト後 ログアウトエラー後 パスワード変更エラー後	EXR_OK	表示されるページは、認証モジュール およびフォワーダーの動作に依存し ます。

4.2.1 インターフェイス関数内 API

インターフェイス関数内で使用可能な API は以下のとおりです。

API 名	内容
IW_ExCSNoSelectDBCIm	認証DBのカラムにあらかじめ値を設定します。
IW_ExCSGetCacheUser	キャッシュに存在するユーザー情報を取得します。
IW_ExCSModifyCacheUser (10.0)	キャッシュに存在するユーザー情報を変更します。
IW_ExCSRleaseRecord	IW_ExCSGetCacheUser、IW_ExCSModifyCacheUser で使用したユーザー情報を解放します。
IW_ExCSGetICP	ICP 2.0 で送信されたリクエストおよび返信するレスポンスの要素を取得します。
IW_ExCSAddICP	ICP 2.0 で送信されたリクエストおよび返信するレスポンスへ要素を追加します。
IW_ExCSModifyICP	ICP 2.0 で送信されたリクエストおよび返信するレスポンスの要素を変更します。
IW_ExCSDeleteICP	ICP 2.0 で送信されたリクエストおよび返信するレスポンスから要素を削除します。
IW_ExCSAddCacheUserGroup	UserExit エントリー実行中ユーザーのグループ情報に対して任意のグループ名を追加します。
IW_ExCSDeleteCacheUserGroup	UserExit エントリー実行中ユーザーのグループ情報に対して任意のグループ名を削除します。
IW_ExCSGetLoginUserCount (10.0)	ログイン中のユーザー数を取得します。

以降ではこれらの API 仕様の解説を行います。

IW_ExCSNoSelectDBCIm

書式 `int IW_ExCSNoSelectDBCIm ( EX_CS_REQ* req, int kind, char* column, char* value )`

機能 ユーザー認証時に認証 DB より取得するユーザー情報で、認証 DB より値を読み込まずにあらかじめ値を設定する API です。

引数 **req** : UserExit ルーチンリクエスト構造体を使用します。インターフェイス関数より渡された req を設定してください。

kind : UserExit ルーチン位置情報フラグを使用します。インターフェイス関数の引数として渡された kind を設定してください。

column : 認証 DB より読み込みを行わないカラム名を指定します。ただし、ユーザー ID のカラムは指定できません。

value : 認証 DB より読み込みを行わないカラムに対して設定する値を指定します。数値データの場合でも文字列で指定します。

戻り値 以下の値が返されます。

EXCS_SUCCESS	: 正常終了
EXCS_ERROR	: 異常終了
EXCS_BADKIND	: API が使用できないエントリー、または不正な kind 値
EXCS_BADCOLUMN	: 存在しないカラムが指定された
EXCS_BADUID	: ユーザー ID カラムが指定された

制限事項 インターフェイス関数の引数 kind が以下の場合、本 API は使用可能となります。
EX_LOGIN_B : ユーザー認証前

IW_ExCSGetCacheUser

書式 **EX_CS_RECORD *IW_ExCSGetCacheUser (EX_CS_REQ*req, int kind, char* sessionid, char* column)**

機能 ログインを行ったユーザーのユーザー情報を内部キャッシュより取得します。

引数 **req** : UserExit ルーチンリクエスト構造体を使用します。インターフェイス関数より渡された req を設定してください。

kind : UserExit ルーチン位置情報フラグを使用します。インターフェイス関数の引数として渡された kind を設定してください。

sessionid : ユーザー情報を取得するログインユーザーのセッションIDを指定します。インターフェイス関数の引数として渡された sessionid を設定してください。

column : 値を取得する認証 DB のカラム名を指定します。認証 DB カラム名以外にキャッシュ内部で保持されている情報の取得も可能です。以下の予約語をカラム名として指定します。

EXCS_API_LOGINAUTH	: 認証方法 (ユーザー ID による認証 : 1、クライアント証明書による認証 : 2)
EXCS_API_USERID	: ユーザー ID (EX_LOGIN_B でユーザー ID を変更している場合、それが取得されます)
EXCS_API_PASSWORD	: パスワード (EX_LOGIN_B でパスワードを変更している場合、それが取得されます)
EXCS_API_PASSWORDFLG	: パスワード有効期限切れフラグ (有効期限内 : 0、有効期限切れ : 1)
EXCS_API_SESSIONTIME	: ログイン有効期限 (YYYYMMDDhhmmss 形式)
EXCS_API_LOGINTIME	: 最終ログイン時刻 (YYYYMMDDhhmmss 形式)
EXCS_API_GROUP	: グループ情報 (複数のグループに属している場合、グループ名がカンマ区切りで連結されます)
EXCS_API_SOURCEADDR	: リクエスト元 IP アドレス
EXCS_API_ICPVER	: ICP バージョン番号
[任意の名称]	: ICP 2.0 にてログイン時に送信されたリクエスト元の環境情報

戻り値 以下の値が返されます。
 NULL 以外 : レコード構造体のポインター
 NULL : 異常終了

レコード構造体は以下のいくつかのメンバー変数を持つ構造体です。内容は以下のようになっています。

```
EX_CS_RECORD {  
    int result;  
    char *name;  
    char *value;  
}
```

result には API の処理結果が格納されます。

name には API の引数 column で指定したカラム名が格納されます。

value にはカラムに対する値が格納されます。

レコード構造体のポインターが返された場合、処理結果がメンバー変数 result に格納されます。格納される処理結果は以下のとおりです。

EXCS_SUCCESS	: 値の取得に成功した
EXCS_ERROR	: システム的なエラーが発生した
EXCS_BADKIND	: API が使用できないエントリー、または不正な kind 値
EXCS_BADCOLUMN	: 不正なカラム名が指定された
EXCS_BADSID	: 未ログインのユーザーが指定された

制限事項 インターフェイス関数の引数 kind の値が以下の場合、本 API は使用可能となります。

EX_LOGIN_A	: ユーザー認証後
EX_ACCESS_B	: アクセス権限チェック前
EX_ACCESS_A	: アクセス権限チェック後
EX_ACCESS_A_ERR	: アクセス権限チェックエラー後
EX_LOGOUT_B	: ログアウト前
EX_LOGOUT_A	: ログアウト後
EX_LOGOUT_A_ERR	: ログアウトエラー後
EX_PASSWD_B	: パスワード変更前
EX_PASSWD_A	: パスワード変更後
EX_PASSWD_A_ERR	: パスワード変更エラー後

本 API 使用後は、戻り値として取得した構造体を後述の API である IW_ExCSReleaseRecord を使用してユーザー情報の破棄を行ってください。

IW_ExCSModifyCacheUser ⑩⑩

書式 **EX_CS_RECORD* IW_ExCSModifyCacheUser (EX_CS_REQ* req, int kind, char* sessionid, char* column, char* value)**

機能 ログインを行ったユーザーのユーザー情報の内部キャッシュに格納されている値を変更します。値の変更後はユーザーが属しているグループ情報が再構築されます。

引数 **req** : UserExit ルーチンリクエスト構造体を使用します。インターフェイス関数より渡された req を設定してください。

kind : UserExit ルーチン位置情報フラグを使用します。インターフェイス関数の引数として渡された kind を設定してください。

sessionid : ユーザー情報を変更するログインユーザーのセッションIDを指定します。インターフェイス関数の引数として渡された sessionid を設定してください。

column : 値を変更する認証 DB のカラム名を指定します。認証 DB カラム名以外にキャッシュ内部で保持されている情報の変更も可能です。以下の予約語をカラム名として指定します。参照可能な内部情報すべてが変更可能ではないことに注意してください。

EXCS_API_PASSWORDFLG : パスワード有効期限切れフラグ

EXCS_API_SESSIONTIME : ログイン有効期限

EXCS_API_LOGINTIME : 最終ログイン時刻

[任意の名称] : ICP 2.0 にてログイン時に送信されたリクエスト元の環境情報

value : 変更後の値を指定します。数値データの場合でも文字列で渡します。

戻り値 以下の値が返されます。
 NULL 以外 : レコード構造体のポインター
 NULL : 異常終了

レコード構造体のポインターが返された場合、処理結果がメンバー変数 **result** に格納されます。格納される処理結果は以下のとおりです。

EXCS_SUCCESS : 値の取得に成功した

EXCS_ERROR : システム的なエラーが発生した

EXCS_BADKIND : API が使用できないエントリー、または不正な kind 値

EXCS_BADCOLUMN : 不正なカラム名が指定された

EXCS_BADSID : 未ログインのユーザーが指定された
EXCS_BADGROUP : 対象グループ情報の再構築でエラーが発生した **10.0**

なお、レコード構造体のメンバー **name** には API の引数で指定したカラム名が格納され、**value** には API で変更する前の値が格納されます。

制限事項 インターフェイス関数の引数 **kind** の値が以下の場合、本 API は使用可能となります。

EX_LOGIN_A : ユーザー認証後
EX_ACCESS_B : アクセス権限チェック前
EX_ACCESS_A : アクセス権限チェック後
EX_ACCESS_A_ERR : アクセス権限チェックエラー後
EX_LOGOUT_B : ログアウト前
EX_LOGOUT_A : ログアウト後
EX_LOGOUT_A_ERR : ログアウトエラー後
EX_PASSWD_B : パスワード変更前
EX_PASSWD_A : パスワード変更後
EX_PASSWD_A_ERR : パスワード変更エラー後

本 API 使用後は戻り値として取得された構造体を後述の API である IW_ExCSReleaseRecord() を使用してユーザー情報の破棄を行ってください。

IW_ExCSReleaseRecord

書式 **void IW_ExCSReleaseRecord (EX_CS_RECORD* rec)**

機能 レコード構造体を解放します。

引数 **rec** : レコード構造体を指定します。IW_ExCSGetCacheUser()、IW_ExCSModifyCacheUser() の戻り値として取得したものを設定してください。

戻り値 戻り値はありません。

制限事項 レコード構造体以外のデータを引数として設定した場合の動作は保証されません。

IW_ExCSGetICP

書式 **EX_CS_RECORD* IW_ExCSGetICP (EX_CS_REQ* req, int type, char* name)**

機能 ICP 2.0 でリクエスト元より送信されたリクエストメッセージおよび、それに対するレスポンスメッセージより構成要素を取得します。

引数 **req** : UserExit ルーチンリクエスト構造体を使用します。インターフェイス関数より渡された **req** を設定してください。

type : 編集対象となるメッセージ内の格納エリアを指定するフラグです。
以下の値を設定します。

EX_TYPE_EXTRA : 付加情報から取得

EX_TYPE_HEADER : ヘッダー情報から取得

EX_TYPE_SUPLE : 補足情報から取得

name : メッセージより取得する要素名を設定します。

戻り値 以下の値が返されます。
 NULL 以外 : レコード構造体のポインター
 NULL : 異常終了

レコード構造体のポインターが返された場合、処理結果がメンバー変数 **result** に格納されます。格納される処理結果は以下のとおりです。

EXCS_SUCCESS : 値の取得に成功した

EXCS_ERROR : システム的なエラーが発生した

EXCS_BADICP : 異なるバージョンのリクエストが指定された

EXCS_BADNAME : メッセージ内に存在しない要素名が指定された

なお、レコード構造体のメンバー **name** には API の引数で指定したカラム名が格納され、**value** には API で変更する前の値が格納されます。

制限事項 格納エリアについては「IceWall Cert Protocol 2.0 開発者マニュアル」を参照してください。

本 API は ICP 2.0 専用です。ICP 1.0 ではエラーとなります。

本 API 使用後は戻り値として取得された構造体を前述の API である **IW_ExCSReleaseRecord()** を使用してユーザー情報の破棄を行ってください。

IW_ExCSAddICP

書式 **int IW_ExCSAddICP (EX_CS_REQ* req, int type, char* name, char* value)**

機能 ICP 2.0 でリクエスト元より送信されたリクエストメッセージおよび、それに対するレスポンスメッセージへ要素情報を追加します。

引数 **req** : UserExit ルーチンリクエスト構造体を使用します。インターフェイス関数より渡された req を設定してください。

type : 編集対象となるメッセージ内の格納エリアを指定するフラグです。
以下の値を設定します。

EX_TYPE_EXTRA : 付加情報へ追加

EX_TYPE_HEADER : ヘッダー情報へ追加

EX_TYPE_SUPLE : 補足情報へ追加

name : メッセージへ格納する要素名を指定します。

value : メッセージへ格納する要素の値を指定します。数値データの場合でも文字列で渡します。

戻り値 以下の値が返されます。
EXCS_SUCCESS : 値の取得に成功した
EXCS_ERROR : システム的なエラーが発生した

制限事項 本 API は ICP 2.0 専用です。ICP 1.0 ではエラーとなります。

IW_ExCSModifyICP

書式 **EX_CS_RECORD* IW_ExCSModifyICP (EX_CS_REQ* req, int type, char* name, char* value)**

機能 ICP 2.0 でリクエスト元より送信されたリクエストメッセージおよび、それに対するレスポンスメッセージに含まれる構成要素を変更します。

引数 **req** : UserExit ルーチンリクエスト構造体を使用します。インターフェイス関数より渡された req を設定してください。

type : 編集対象となるメッセージ内の格納エリアを指定するフラグです。
以下の値を設定します。

EX_TYPE_EXTRA : 付加情報の要素を変更

EX_TYPE_HEADER : ヘッダー情報の要素を変更

EX_TYPE_SUPLE : 補足情報の要素を変更

name : メッセージ内の値を変更する要素名を指定します。

value : メッセージへ変更する要素の値を指定します。数値データの場合でも文字列で渡します。

戻り値 以下の値が返されます。
NULL 以外 : レコード構造体のポインター
NULL : 異常終了

レコード構造体のポインターが返された場合、処理結果がメンバー変数 **result** に格納されます。格納される処理結果は以下のとおりです。

EXCS_SUCCESS : 値の取得に成功した

EXCS_ERROR : システム的なエラーが発生した

EXCS_BADICP : 異なるバージョンのリクエストが指定された

EXCS_BADNAME : メッセージ内に存在しない要素名が指定された

なお、レコード構造体のメンバー **name** には API の引数で指定したカラム名が格納され、**value** には API で変更する前の値が格納されます。

制限事項 格納エリアについては「IceWall Cert Protocol 2.0 開発者マニュアル」を参照してください。

本 API 使用後は戻り値として取得された構造体を前述の API である IW_ExCSReleaseRecord() を使用してユーザー情報の破棄を行ってください。

本 API は ICP 2.0 専用です。ICP 1.0 ではエラーとなります。

IW_ExCSDeleteICP

書式 **EX_CS_RECORD*** IW_ExCSDeleteICP (**EX_CS_REQ*** req, int type, char* name)

機能 ICP 2.0 でリクエスト元より送信されたリクエストメッセージおよび、それに対するレスポンスメッセージより構成要素を削除します。

引数 **req** : UserExit ルーチンリクエスト構造体を使用します。インターフェイス関数より渡された req を設定してください。

type : 編集対象となるメッセージ内の格納エリアを指定するフラグです。
以下の値を設定します。

EX_TYPE_EXTRA : 付加情報の要素を削除

EX_TYPE_HEADER : ヘッダー情報の要素を削除

EX_TYPE_SUPLE : 補足情報の要素を削除

name : メッセージ内の値を変更する要素名を指定します。

戻り値 以下の値が返されます。
NULL 以外 : レコード構造体のポインター
NULL : 異常終了

レコード構造体のポインターが返された場合、処理結果がメンバー変数 **result** に格納されます。格納される処理結果は以下のとおりです。

EXCS_SUCCESS : 値の取得に成功した

EXCS_ERROR : システム的なエラーが発生した

EXCS_BADICP : 異なるバージョンのリクエストが指定された

EXCS_BADNAME : メッセージ内に存在しない要素名が指定された

なお、レコード構造体のメンバー **name** には API の引数で指定したカラム名が格納され、**value** には API で変更する前の値が格納されます。

制限事項 格納エリアについては「IceWall Cert Protocol 2.0 開発者マニュアル」を参照してください。

本 API 使用後は戻り値として取得された構造体を前述の API である IW_ExCSReleaseRecord() を使用してユーザー情報の破棄を行ってください。

本 API は ICP 2.0 専用です。ICP 1.0 ではエラーとなります。

IW_ExCSAddCacheUserGroup ⑩.0

書式 **int IW_ExCSAddCacheUserGroup (EX_CS_REQ *req, int kind, char *sessionid, char *groupname)**

機能 UserExit エントリ実行中ユーザーのグループ情報に対して任意のグループ名を追加します。

本 API は 8.0 R2 以降のバージョンにて使用できます。

引数 **req** : UserExit ルーチンリクエスト構造体を使用します。インターフェイス関数より渡された req を設定してください。

kind : UserExit ルーチン位置情報フラグを使用します。インターフェイス関数の引数として渡された kind を設定してください。

sessionid : ユーザー情報を取得するログインユーザーのセッションIDを指定します。インターフェイス関数の引数として渡された sessionid を設定してください。

groupname : 追加するグループ名を指定します。グループ名は大小文字を区別しません。対象ユーザーが所属していないグループ名を指定してください。追加するグループは、グループ設定ファイル (cert.grp) に定義されていないグループ、またはグループを追加するユーザーが所属条件に一致しないグループでも追加可能です。

戻り値 以下の値が返されます。

EXCS_SUCCESS : グループ情報の追加に成功した

EXCS_ERROR : システム的なエラーが発生した

EXCS_BADKIND : API が使用できないエントリ、または不正な kind 値

EXCS_BADSID : 未ログインユーザーのセッション ID が指定された

EXCS_BADNAME : 不正な追加グループ名が指定された

制限事項 インターフェイス関数の引数 kind の値が以下の場合、本 API は使用可能となります。

EX_LOGIN_A : ユーザー認証後

EX_ACCESS_B : アクセス権限チェック前

EX_ACCESS_A : アクセス権限チェック後

EX_PASSWD_B : パスワード変更前

EX_PASSWD_A : パスワード変更後

本 API を実行した後に IW_ExCSModifyCacheUser() を実行すると、本 API で設定したグループ情報は無効となります。API の実行順序に注意してください。

IW_ExCSDeleteCacheUserGroup

書式 **int IW_ExCSDeleteCacheUserGroup (EX_CS_REQ *req, int kind, char *sessionid, char *groupname)**

機能 UserExit エントリ実行中ユーザーのグループ情報に対して任意のグループ名を削除します。

本 API は 8.0 R2 以降のバージョンにて使用できます。

引数 **req** : UserExit ルーチンリクエスト構造体を使用します。インターフェイス関数より渡された req を設定してください。

kind : UserExit ルーチン位置情報フラグを使用します。インターフェイス関数の引数として渡された kind を設定してください。

sessionid : ユーザー情報を取得するログインユーザーのセッション ID を指定します。インターフェイス関数の引数として渡された sessionid を設定してください。

groupname : 削除するグループ名を指定します。グループ名は大小文字を区別しません。対象ユーザーが所属していないグループ名を指定してください。

戻り値 以下の値が返されます。

EXCS_SUCCESS : グループ情報の削除に成功した
EXCS_BADKIND : API が使用できないエントリ、または不正な kind 値
EXCS_BADSID : 未ログインユーザーのセッション ID が指定された
EXCS_BADNAME : 不正な削除グループ名が指定された
EXCS_BADGROUP : 削除対象グループ以外に所属していない

制限事項 インターフェイス関数の引数 kind の値が以下の場合、本 API は使用可能となります。

EX_LOGIN_A : ユーザー認証後
EX_ACCESS_B : アクセス権限チェック前
EX_ACCESS_A : アクセス権限チェック後
EX_PASSWD_B : パスワード変更前
EX_PASSWD_A : パスワード変更後

本 API を実行した後に IW_ExCSModifyCacheUser() を実行すると、本 API で設定したグループ情報は無効となります。API の実行順序に注意してください。

IW_ExCSGetLoginUserCount 10.0

書式 **int IW_ExCSGetLoginUserCount(EX_CS_REQ *req)**

機能 ログイン中のユーザー数を取得します。

本 API は 10.0 以降のバージョンにて使用できます。

引数 **req** : UserExit ルーチンリクエスト構造体を使用します。インターフェイス関数より渡された **req** を設定してください。**req** が NULL または異常な場合は異常終了します。

戻り値 以下の値が返されます。
 ログインユーザー数 (0 以上) : 正常終了
 EXCS_ERROR : 異常終了

制限事項 本 API はすべてのエントリーで使用可能です。

5 注意事項

UserExit ルーチンを作成および導入する上で注意しなければならない点をご紹介します。

5.1 UserExit ルーチンの影響範囲

UserExit ルーチン内で実行される処理は、フォワーダーおよび認証モジュールの内部処理と密接な関係にあるため、UserExit ルーチン内でシステム的なエラーが発生した場合、そのモジュールがダウンする可能性があります。作成されたライブラリのテストを十分に行った上で、各モジュールへ組み込んでください。

また、UserExit ルーチン内で `exit()` をコールすると、そのモジュールの実行プロセスは終了してしまうため、記述しないでください。

5.2 パフォーマンスの低下

UserExit ルーチン内で処理を追加する場合、追加した処理を実行するため、パフォーマンスが低下します。極力複雑な処理は行わないようにしてください。

5.3 メモリリーク

UserExit ルーチン内で独自に取得したメモリ領域については、必ず解放するようにしてください。解放しない場合はメモリリークの原因となります。

5.4 従来バージョンからのバージョンアップについての注意事項

64 バイト長でセッション ID を生成する設定の場合は、UserExit で取得されるセッション ID の長さも 64 バイト長とする対応が必要になります。 **10.0**

5.5 サポートについて

製品のサポートは、標準で添付されるライブラリを使用した場合にのみ受けることができます。お客さまが作成された UserExit ルーチンが導入されている状態ではサポートを受けることができません。標準添付のライブラリはサポート時に必要となりますので、必ずバックアップを行ってください。

6 制限事項

UserExit ルーチンを作成する場合には、以下の制限事項があることに注意してください。

以下の制限内で作成されない場合には、各プロセス自体が実行できなくなる、動作が不安定になる原因になります。

6.1 モジュール共通の制限

フォワーダー、認証モジュール共通の制限事項です。

6.1.1 リンクする標準ライブラリのバージョンについて

フォワーダーおよび認証モジュールは、ダイナミックにライブラリを使用して動作するように、作成されています。

ただし、UserExit ルーチンにアーカイブライブラリをリンクしている場合には、UserExit ルーチンにリンクされているライブラリが利用されることになるため、バージョンの違いによる不整合を招く可能性があります。(OS への Patch を適用しても解決しない、などのトラブルにもなる可能性があります)

注) 特に Oracle など、インストール環境によって生成されるライブラリ構成が変わる (libclntsh.sl の構成が変わる) ものを UserExit ルーチンで利用する場合にはリンクされているライブラリの状態について十分注意する必要があります。

6.1.2 互換性について

UserExit ルーチンは、IceWall の上位および下位バージョンとの互換性がないため、UserExit ルーチンを作成したバージョン以外のバージョンで使用する場合にはリビルドする必要があります。このため、IceWall 本体のバージョンアップを行った場合には、UserExit ルーチンをリビルドするようにしてください。

6.2 フォワーダーの制限

フォワーダーに関する制限事項です。

6.2.1 OS 共通の制限

64bit バイナリーとしてビルドしてください。

6.2.2 HP-UX 版 (Itanium)

コンパイルオプションには必ず「+DD64」を付加してください。

UserExit ルーチンを組み込むフォワーダーが動作するプロセッサと同一アーキテクチャのコンパイラでビルドしてください。

6.2.3 Linux 版

コンパイルオプションには必ず「-m64」を付加してください。

6.3 認証モジュールの制限

認証モジュールに関する制限事項です。

6.3.1 OS 共通の制限

64bit バイナリーとしてビルドしてください。

認証モジュールは内部処理でスレッドを使用しており、UserExit ルーチンはスレッドよりコールされるため、標準ライブラリ関数は必ずスレッドセーフ版を使用するようにしてください。また、ユーザー関数を作成する場合は、スレッドセーフとして動作する必要があります。

6.3.2 HP-UX 11i v3 版 (Itanium)

コンパイルオプションには必ず「+DD64」を付加してください。

UserExit ルーチンを組み込む認証モジュールが動作するプロセッサと同一アーキテクチャのものでビルドしてください。また、リンクするライブラリは 64bit ライブラリのみとしてください。

6.3.3 Linux 版

コンパイルオプションには必ず「-m64」を付加してください。

認証モジュールのスレッドは「Linux Threads」と「Native POSIX Library Thread」のどちらもサポートされます。

7 UserExit ルーチン開発のヒント

UserExit ルーチンの開発を行うにあたり、そのヒントとなる方法をご紹介します。

7.1 フォワーダー UserExit ルーチン

7.1.1 IceWall に対するリクエストの判別方法

クライアントから送信されたリクエストが IceWall に対するものか、バックエンド Web サーバーに対するものかどうかはメソッドと POST データの内容より判断します。

以下はその判断方法です。

- (1) IW_ExDFWInterface() の引数 kind の値を確認します。
- (2) kind の値が EX_DFW_KIND_REQ の場合、IW_ExDFWGetMethod() でメソッドを取得します。
- (3) メソッドが POST の場合、IW_ExDFWGetPostdata() で POST データを取得します。
- (4) POST データ内にフォワーダー設定ファイル (dfw.conf) の POSTKEY_LOGIN 項目で設定した値が含まれている場合はログイン処理、POSTKEY_LOGOUT 項目で設定した値が含まれている場合はログアウト処理、POSTKEY_PWDCHG 項目で設定した値が含まれている場合はパスワード変更処理となります。

7.1.2 バックエンド Web サーバーへのリクエストを制御する方法

バックエンド Web サーバーに対して送信するリクエストに含まれる HTTP ヘッダーの値の変更や、新規に HTTP ヘッダーを追加することが可能です。

以下はその制御方法です。

- (1) IW_ExDFWInterface() の引数 kind の値を確認します。
- (2) kind の値が EX_DFW_KIND_SEND の場合、IW_ExDFWGetBuffer() でリクエストデータバッファを取得します。
- (3) リクエストデータバッファの内容を作業バッファにコピーします。作業バッファはあらかじめ確保しておく必要があります。
- (4) 作業バッファ内の HTTP ヘッダー部分に対して目的の処理を行います。
- (5) IW_ExDFWModifyBuffer() で作業バッファの内容をリクエストデータとしてセットします。なお、バックエンド Web サーバーに送信する POST データがバイナリーデータの場合は IW_ExDFWModifyBufferLength() を使用します。

7.1.3 バックエンド Web サーバーからのレスポンスを制御する方法

バックエンド Web サーバーより受信したレスポンスに含まれる HTTP ヘッダーの値の変更や、新規に HTTP ヘッダーを追加することが可能です。

以下はその制御方法です。

- (1) IW_ExDFWInterface() の引数 kind の値を確認します。
- (2) kind の値が EX_DFW_KIND_RECV の場合、IW_ExDFWGetBuffer() でレスポンスデータバッファを取得します。
- (3) レスポンスデータバッファの内容を作業バッファにコピーします。作業バッファはあらかじめ確保しておく必要があります。
- (4) 作業バッファ内の HTTP ヘッダー部分に対して目的の処理を行います。

- (5) IW_ExDFWModifyBuffer() で作業バッファの内容をレスポンスデータとしてセットします。なお、バックエンド Web サーバーより受信したコンテンツがバイナリーデータの場合は IW_ExDFWModifyBufferLength() を使用します。

7.2 認証モジュール UserExit ルーチン

7.2.1 各処理後のエラー判別方法

各処理にてエラーが発生した後にコールされた UserExit ルーチンで、どのようなエラーが発生したかを引数より判断します。

以下はその判断方法です。(ログインエラーの場合)

- (1) IW_ExInterFace() の引数 kind の値を調べます。
- (2) kind の値が EX_LOGIN_A_ERR の場合、IW_ExInterFace() の引数 data に含まれるカンマを調べます。
- (3) カンマ以降の文字列を取得し数値に変換します。
- (4) 変換した数値を制御コードと比較してエラー内容を判断します。

8 サンプルソース

以下に各 UserExit ルーチンのサンプルソースをご紹介します。

8.1 フォワーダー UserExit サンプル

8.1.1 GET メソッドを POST メソッドへ変更

```
int IW_ExDFWInterFace( req, kind )
EX_DFW_REQ *req;
int    kind;
{
    char postdata[128];

    if ( kind == EX_DFW_KIND_REQ ) {
        if ( strcmp( IW_ExDFWGetMethod( req, kind ), "GET" ) ) {
            IW_ExDFWModifyMethod( req, kind, EX_DFW_METHOD_POST );
            strcpy( postdata, "Name1=Value1&Name2=Value2" );
            IW_ExDFWModifyPostdata( req, kind, postdata );
        }
    }

    return( EX_DFW_OK );
}
```

8.1.2 リクエストの HTTP ヘッダーを変更

```
int IW_ExDFWInterFace( req, kind )
EX_DFW_REQ *req;
int    kind;
{
    char *req1, *req2;

    if ( kind == EX_DFW_KIND_SEND ) {
        req1 = IW_ExDFWGetBuffer( req, kind );
        req2 = (char*)malloc( strlen( req1 ) + 1 );
        strcpy( req2, req1 );
        HttpHeaderModify( req2, "User-Agent", "MyBrowser2.1" ); ※ 1
        IW_ExDFWModifyBuffer( req, kind, req2 );
        free( req2 );
    }

    return( EX_DFW_OK );
}
```

※ 1 HttpHeaderModify() 関数は API として存在するものではありません。
HTTP ヘッダー内の値を書き換える処理を想定した仮想の関数です。

8.2 認証モジュール UserExit のサンプル

8.2.1 特定のユーザー ID を強制的に guest へ変更

```
int IW_ExInterFace( kind, userid, password, sessionid, data, req )
int kind;
char *userid;
char *password;
char *sessionid;
char *data;
EX_CS_REQ *req;
{

    if ( kind == EX_LOGIN_B ) {
        if ( strcmp( userid, "user01" ) ) {
            strcpy( userid, "guest" );
            strcpy( password, "guest" );
        }
    }

    return( EXR_OK );
}
```

8.2.2 特定の URL に対するリクエストをログイン要求に変更

```
int IW_ExInterFace( kind, userid, password, sessionid, data, req )
int kind;
char *userid;
char *password;
char *sessionid;
char *data;
EX_CS_REQ *req;
{

    if ( kind == EX_ACCESS_B ) {
        if ( strcmp( data, "http://www.hp.com/members" ) ) {
            return( EXR_RLOGIN );
        }
    }

    return( EXR_OK );
}
```

9 参考

標準でインストールされる UserExit ルーチン開発キットの内容です。
開発キットの構成は以下のディレクトリおよびファイルとなっています。

ディレクトリおよびファイル					内容
/opt/icewall-ss0	/developkit	/dfw	/DfwExit	/dfwinterface.c	フォワーダー用 UserExit ルーチン 開発キット
				/dfwinterface.h	
				/Makefile	
		/certd	/CertExit	/iwintface.c	認証モジュール用 UserExit ルーチン 開発キット
				/iwintface.h	
				/Makefile	

なお、この開発キットで作成されるライブラリ名は標準でインストールされるライブラリ名と同じものになりますので、作成および導入する際には必ずバックアップを行ってください。

9.1 フォワーダー用 UserExit ルーチン開発キット

9.1.1 スケルトンソース (dfwinterface.c)

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>

#include "dfwinterface.h"

int IW_ExDFWInterFace( req, kind )
EX_DFW_REQ *req;
int kind;
{
    switch( kind ) {
        case EX_DFW_KIND_REQ: break;
        case EX_DFW_KIND_SEND: break;
        case EX_DFW_KIND_RECV: break;
        case EX_DFW_KIND_END: break;
        case EX_DFW_KIND_LOGIN_SEND: break;
        case EX_DFW_KIND_LOGIN_RECV: break;
        case EX_DFW_KIND_LOGOUT_SEND: break;
        case EX_DFW_KIND_LOGOUT_RECV: break;
        case EX_DFW_KIND_ACC_SEND: break;
        case EX_DFW_KIND_ACC_RECV: break;
        case EX_DFW_KIND_PWDCHG_SEND: break;
        case EX_DFW_KIND_PWDCHG_RECV: break;
        case EX_DFW_KIND_ALOGIN_SEND: break;
        case EX_DFW_KIND_ALOGIN_RECV: break;
    }

    return( EX_DFW_OK );
}
```

}

9.1.2 ヘッダーファイル (dfwinterface.h)

```

#ifndef DFWINTFACE_H
#define DFWINTFACE_H

/* Kind */
#define EX_DFW_KIND_REQ          0      /* リクエスト情報取得後 */
#define EX_DFW_KIND_SEND        1      /* データ送信前 */
#define EX_DFW_KIND_RECV        2      /* データ送信後 */
#define EX_DFW_KIND_END          3      /* コンテンツ送信前 (6.0SP1) */
#define EX_DFW_KIND_LOGIN_SEND  4      /* ログイン要求送信前 (8.0R3) */
#define EX_DFW_KIND_LOGIN_RECV  5      /* ログイン要求送信後 (8.0R3) */
#define EX_DFW_KIND_LOGOUT_SEND 6      /* ログアウト要求送信前 (8.0R3) */
#define EX_DFW_KIND_LOGOUT_RECV 7      /* ログアウト要求送信後 (8.0R3) */
#define EX_DFW_KIND_ACC_SEND     8      /* アクセス要求送信前 (8.0R3) */
#define EX_DFW_KIND_ACC_RECV     9      /* アクセス要求送信後 (8.0R3) */
#define EX_DFW_KIND_PWDCHG_SEND 10     /* パスワード変更要求送信前 (8.0R3) */
#define EX_DFW_KIND_PWDCHG_RECV 11     /* パスワード変更要求送信後 (8.0R3) */
#define EX_DFW_KIND_ALOGIN_SEND 12     /* Agent ログイン要求送信前 (8.0R3) */
#define EX_DFW_KIND_ALOGIN_RECV 13     /* Agent ログイン要求送信後 (8.0R3) */
/* API Return */
#define EX_DFW_ERROR             -1     /* 異常終了 */
#define EX_DFW_SUCCESS           0      /* 正常終了 */
#define EX_DFW_BADKIND           1      /* 不正 Kind */
#define EX_DFW_NOTFOUND          2      /* 制御対象無し (8.0R3) */
/* API Switch */
#define EX_DFW_HTTP              0      /* HTTP 動作 */
#define EX_DFW_HTTPS             1      /* HTTPS 動作 */
#define EX_DFW_METHOD_GET        0      /* メソッド GET */
#define EX_DFW_METHOD_POST       1      /* メソッド POST */
#define EX_DFW_UID_LOGIN         0      /* ログインユーザ ID(7.0SP2) */
#define EX_DFW_UID_INPUT         1      /* 入力ユーザ ID(7.0SP2) */
#define EX_DFW_ICP_HEADER        0      /* ヘッダ部 (8.0R3) */
#define EX_DFW_ICP_BODY          1      /* ボディ部 (8.0R3) */
#define EX_DFW_ICP_DEL           0      /* 削除処理 (8.0R3) */
#define EX_DFW_ICP_ADD           1      /* 追加処理 (8.0R3) */
#define EX_DFW_ICP_MOD           2      /* 変更処理 (8.0R3) */
/* Return */
#define EX_DFW_OK                 0      /* ページなし (正常終了) */
#define EX_DFW_LOGIN              10     /* ログインページ */
#define EX_DFW_LOGINCERT          17     /* 証明書ログインページ */
#define EX_DFW_LOGINFORCE         6      /* 強制ログインページ */
#define EX_DFW_LOGINUIDERR        1      /* ユーザ ID エラーページ */
#define EX_DFW_LOGINPWDERR        18     /* ログインパスワードエラーページ */
#define EX_DFW_LOGINLOCKERR       4      /* アカウントロックページ */
#define EX_DFW_LOGINCERTERR       7      /* 初回認証済みエラーページ */
#define EX_DFW_LOGINSERIALERR     8      /* シリアル No エラーページ */
#define EX_DFW_LOGINGRPERR        9      /* グループエラーページ */
#define EX_DFW_LOGINSTOP          5      /* ログインストップページ */
#define EX_DFW_LOGINLIMITERR      19     /* ログイン数オーバーページ */
#define EX_DFW_LOGINTIMEERR       46     /* ログイン送信時間制限ページ */
#define EX_DFW_LOGOUT             20     /* ログアウトページ */
#define EX_DFW_LOGOUTOK           21     /* ログアウト成功ページ */
#define EX_DFW_LOGOUTNO           22     /* ログアウト不可ページ */
#define EX_DFW_LOGOUTERR          23     /* ログアウトエラーページ */

```



```

#define EX_DFW_ACCESS 11 /* アクセス権限エラーページ */
#define EX_DFW_SENDErr 24 /* POST 送信エラーページ */
#define EX_DFW_REQACLERR 50 /* リクエスト ACL エラーページ (10.0) */
#define EX_DFW_POSTERR 45 /* POSTDATA 最大値エラーページ (8.0R3) */
#define EX_DFW_PWDCHG 3 /* パスワード変更ページ */
#define EX_DFW_PWDCHGOK 25 /* パスワード変更成功ページ */
#define EX_DFW_PWDOLDERR 26 /* 旧パスワードエラーページ */
#define EX_DFW_PWDREERR 27 /* 再入力パスワード変更エラーページ */
#define EX_DFW_PWDPCYERR 28 /* パスワード変更ポリシーエラーページ */
#define EX_DFW_PWDVIOERR 29 /* パスワード変更権限なしページ */
#define EX_DFW_PWDNOLOGIN 30 /* パスワード変更未ログインページ */
#define EX_DFW_PWDERR 2 /* パスワード変更エラーページ */
#define EX_DFW_PWDWARNING 44 /* パスワード有効期限切れ警告ページ */
#define EX_DFW_PWDTIMEERR 47 /* パスワード変更送信時間制限ページ */
#define EX_DFW_SYSERR 13 /* システムエラーページ */
#define EX_DFW_ALIASNO 31 /* エイリアスなしエラーページ */
#define EX_DFW_ALIASBAD 32 /* 未定義エイリアスエラーページ */
#define EX_DFW_DOWNCERTD 33 /* Certd ダウンエラーページ */
#define EX_DFW_DOWNDB 12 /* DB ダウンエラーページ */
#define EX_DFW_DOWNBKEND 34 /* バックエンドダウンエラーページ */
#define EX_DFW_TOUTCERTD 35 /* Certd タイムアウトページ */
#define EX_DFW_TOUTBKEND 36 /* バックエンドタイムアウトページ */
#define EX_DFW_DBBUSYERR 49 /* DB ビジーエラーページ (10.0) */
#define EX_DFW_FILTERGET 37 /* GET フィルタエラーページ */
#define EX_DFW_FILTERPOST 38 /* POST フィルタエラーページ */
#define EX_DFW_FILTERHTML 39 /* HTML フィルタエラーページ */
#define EX_DFW_FILTERHOST 40 /* ホストフィルタエラーページ */
#define EX_DFW_FILTERREQ 48 /* リクエストフィルタエラーページ (10.0) */
#define EX_DFW_USREX1 14 /* ユーザ定義エラー 1 ページ */
#define EX_DFW_USREX2 15 /* ユーザ定義エラー 2 ページ */
#define EX_DFW_USREX3 16 /* ユーザ定義エラー 3 ページ */
#define EX_DFW_USREX4 41 /* ユーザ定義エラー 4 ページ */
#define EX_DFW_USREX5 42 /* ユーザ定義エラー 5 ページ */
#define EX_DFW_USREX6 43 /* ユーザ定義エラー 6 ページ */

typedef struct ex_dfw_req EX_DFW_REQ;

extern int IW_ExDFWInterFace( EX_DFW_REQ *req, int kind );
extern int IW_ExDFWChangeProtocol( EX_DFW_REQ *ex_rec, int kind, int flg );
extern char *IW_ExDFWGetMethod( EX_DFW_REQ *ex_rec, int kind );
extern int IW_ExDFWModifyMethod( EX_DFW_REQ *ex_rec, int kind, int method );
extern char *IW_ExDFWGetPostdata( EX_DFW_REQ *ex_rec, int kind );
extern int IW_ExDFWModifyPostdata( EX_DFW_REQ *ex_rec, int kind, char *buf );
extern char *IW_ExDFWGetBuffer( EX_DFW_REQ *ex_rec, int kind );
extern int IW_ExDFWModifyBuffer( EX_DFW_REQ *ex_rec, int kind, char *buf );
extern int IW_ExDFWModifyBufferLength( EX_DFW_REQ *ex_rec, int kind, char *buf, int len );
extern int IW_ExDFWModifyRequestPath( EX_DFW_REQ *ex_rec, int kind, char *path );
extern int IW_ExDFWModifyPathinfo( EX_DFW_REQ *ex_rec, int kind, char *path );
extern char *IW_ExDFWGetUid( EX_DFW_REQ *ex_rec, int kind, int flg );
extern char *IW_ExDFWGetCertData( EX_DFW_REQ *ex_rec, int kind, char *name, int target );
extern int IW_ExDFWCtrlCertData( EX_DFW_REQ *ex_rec, int kind
    ,char *name, char *value, int target, int mode );

#endif /* #ifndef DFWINTFACE_H */

```

9.1.3 メイクファイル (HP-UX Itanium64Bit 版 : Makefile)

```
LD=cc

LDFLAGS=+z +DD64 +noobjdebug -Ae

LIBS=

INCLUDE=-I./

MAKEFILE=Makefile

OBJS=dfwinterface.o

PROGRAM=DfwExit

INST_LIB=DfwExit

SRCS=dfwinterface.c

all:$(PROGRAM)

$(PROGRAM):$(OBJS)
ld -b -o lib$(PROGRAM).sl $(OBJS) $(LIBS)

.c.o:
$(LD) $(CC_CMD) $(LDFLAGS) $(INCLUDE) -c $(SRCS)

debug:
make CC_CMD="-DDEBUG -g"

clean:
rm -f $(OBJS) lib$(PROGRAM).sl core

install:
cp -p lib$(PROGRAM).sl ../lib/lib$(INST_LIB).sl

###
dfwintface.o: ./dfwinterface.h
```

9.1.4 メイクファイル (Linux64Bit 版 : Makefile)

```
LD                =    gcc
LDFLAGS           =    -DLinux -m64 -fPIC
LIBS              =
INCLUDE           =    -I./
MAKEFILE          =    Makefile
OBJS              =    dfwinterface.o
PROGRAM           =    DfwExit
INST_LIB          =    DfwExit
SRCS              =    dfwinterface.c

all:              $(PROGRAM)

$(PROGRAM):       $(OBJS)
gcc -shared -o lib$(PROGRAM).sl $(OBJS) $(LIBS)

.c.o:
$(LD) $(CC_CMD) $(LDFLAGS) $(INCLUDE) -c $(SRCS)

debug:
make CC_CMD="-DDEBUG -g"

clean:
rm -f $(OBJS) lib$(PROGRAM).sl core

install:
cp -p lib$(PROGRAM).sl ../lib/lib$(INST_LIB).sl
rm -f ../lib/lib$(INST_LIB).so
ln -s ../lib/lib$(INST_LIB).sl ../lib/lib$(INST_LIB).so

###
dfwintface.o: ./dfwinterface.h
```

9.2 認証モジュール用 UserExit ルーチン開発キット

9.2.1 スケルトンソース (iwintface.c)

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>

#include "iwintface.h"

int IW_ExInterFace( kind, userid, password, sessionid, data, req )
int    kind;
char   *userid;
char   *password;
char   *sessionid;
char   *data;
EX_CS_REQ *req;
{
    switch( kind ) {
        case EX_LOGIN_B : break;
        case EX_LOGIN_A : break;
        case EX_LOGIN_A_ERR : break;
        case EX_ACCESS_B : break;
        case EX_ACCESS_A : break;
        case EX_ACCESS_A_ERR : break;
        case EX_LOGOUT_B : break;
        case EX_LOGOUT_A : break;
        case EX_LOGOUT_A_ERR : break;
        case EX_PASSWD_B : break;
        case EX_PASSWD_A : break;
        case EX_PASSWD_A_ERR : break;
    }

    return( EXR_OK );
}
```

9.2.2 ヘッダーファイル (iwinface.h)

```

/*-----*/
/* UserExit Interface */
/*-----*/

#ifndef IWINTFACE_H
#define IWINTFACE_H

/*-----*/
/* Define */
/*-----*/

#define EX_LOGIN_B          1          /* ユーザー認証前 */
#define EX_LOGIN_A          2          /* ユーザー認証後 */
#define EX_ACCESS_B        3          /* アクセス権限チェック前 */
#define EX_ACCESS_A        4          /* アクセス権限チェック後 */
#define EX_LOGOUT_B        5          /* ログアウト前 */
#define EX_LOGOUT_A        6          /* ログアウト後 */
#define EX_LOGIN_A_ERR     7          /* ユーザー認証後エラー */
#define EX_ACCESS_A_ERR    8          /* アクセス権限チェック後エラー */
#define EX_LOGOUT_A_ERR    9          /* ログアウト後エラー */
#define EX_PASSWD_B        10         /* パスワード前 */
#define EX_PASSWD_A        11         /* パスワード後 */
#define EX_PASSWD_A_ERR    12         /* パスワード後エラー */

#define EX_N_UID            "N_UID"    /* ユーザー ID による認証 */
#define EX_F_UID            "F_UID"    /* ユーザー ID による強制認証 */
#define EX_N_CERT          "N_CERT"    /* 証明書による認証 */
#define EX_F_CERT          "F_CERT"    /* 証明書による強制認証 */
#define EX_N_SAML          "N_SAML"    /* SAML による認証 */
#define EX_F_SAML          "F_SAML"    /* SAML による強制認証 */
#define EX_N_FEDE          "N_FEDE"    /* フェデレーションによる認証 */
#define EX_F_FEDE          "F_FEDE"    /* フェデレーションによる強制認証 */
#define EX_MANUAL          "MANUAL"    /* ユーザーによるログアウト */
#define EX_AUTO            "AUTO"      /* 自動ログアウト */
#define EX_FORCED          "FORCED"    /* 強制ログアウト */
#define EXR_OK             0          /* 処理成功 */
#define EXR_UIDERR         1          /* ユーザー ID エラー */
#define EXR_PWDERR         2          /* パスワードエラー */
#define EXR_PWDCHG         3          /* パスワード変更要求 */
#define EXR_USRLOCK        4          /* アカунツロックエラー */
#define EXR_LOGINNO        5          /* ログインロックエラー */
#define EXR_DLOGIN         6          /* 二重ログインエラー */
#define EXR_PLOGIN         7          /* 並行利用エラー */
#define EXR_SNOERR         8          /* シリアル No. エラー */
#define EXR_GRPERR         9          /* 所属グループなしエラー */
#define EXR_RLOGIN         10         /* 再ログイン要求 */
#define EXR_ACCERR         11         /* アクセス権限エラー */
#define EXR_DBERR          12         /* 認証 DB エラー */
#define EXR_SYSERR         13         /* システムエラー */
#define EXR_USREX1         14         /* ユーザー定義エラー 1 */
#define EXR_USREX2         15         /* ユーザー定義エラー 2 */
#define EXR_USREX3         16         /* ユーザー定義エラー 3 */
#define EXR_LOGOUTNG       17         /* ログアウトエラー */
#define EXR_PWDPOERR       18         /* パスワードポリシーエラー */
#define EXR_PWDVIOERR      19         /* パスワード権限エラー */
#define EXR_PWDWARN        20         /* パスワード有効期限切れ警告 */
#define EXR_LOGINLMT       21         /* ユーザー数オーバーエラー */

```

```

#define EXR_ACLREQ          22          /* リクエスト実行権限エラー          */
#define EXR_SIDERR          23          /* セッション ID 生成エラー          */
#define EXR_DBBUSY          24          /* DB ビジーエラー                  */
#define EXR_PWDMINLEN       25          /* 最小パスワード長エラー            */
#define EXR_PWDMAXLEN       26          /* 最大パスワード長エラー            */
#define EXR_PWDALPHANUM     27          /* パスワード文字種エラー            */
#define EXR_PWDSAMEPASS     28          /* ユーザ ID をパスワードに使用      */

#define EX_TYPE_EXTRA       1          /* ICP - 付加情報                    */
#define EX_TYPE_HEADER      2          /* ICP - ヘッダー情報                */
#define EX_TYPE_SUPLE       3          /* ICP - 補足情報                    */

/*-----*/
/* API Define                                          */
/*-----*/

#define EXCS_ERROR          -1          /* 異常終了                          */
#define EXCS_SUCCESS        0          /* 正常終了                          */
#define EXCS_BADKIND        1          /* 不正 Kind エラー                  */
#define EXCS_BADSID         2          /* 該当ユーザーなしエラー            */
#define EXCS_BADCOLUMN      3          /* 該当カラムなしエラー              */
#define EXCS_BADGROUP       4          /* グループ無所属エラー              */
#define EXCS_BADREPLI       5          /* レプリケーションエラー            */
#define EXCS_BADUID         6          /* ユーザー ID 不参照設定エラー      */
#define EXCS_BADICP         7          /* ICP プロトコルエラー              */
#define EXCS_BADNAME        8          /* 該当 NAME なしエラー              */

/*-----*/
/* API Structure                                      */
/*-----*/

typedef struct ex_cs_req EX_CS_REQ;    /* 総合情報構造体                    */

typedef struct _EX_CS_RECORD{          /* API 用レコード構造体              */
int result;                          /* 戻り値 (EXCS_SUCCESS ~)            */
char *name;                          /* カラム名                          */
char *value;                         /* カラム値 (参照値 or 旧カラム値)    */
}EX_CS_RECORD;

/*-----*/
/* API Prtotype                                       */
/*-----*/

extern void IW_ExCSReleaseRecord( EX_CS_RECORD *record );
extern int IW_ExCSNoSelectDBClm( EX_CS_REQ *req, int kind, char *column,
                                char *value );
extern EX_CS_RECORD *IW_ExCSGetCacheUser( EX_CS_REQ *req, int kind,
                                           char *sessionid, char *column );
extern EX_CS_RECORD *IW_ExCSModifyCacheUser( EX_CS_REQ *req, int kind,
                                              char *sessionid, char *column, char *value );
extern EX_CS_RECORD *IW_ExCSGetICP( EX_CS_REQ *req, int type, char *name );
extern EX_CS_RECORD *IW_ExCSModifyICP( EX_CS_REQ *req, int type, char *name,
                                        char *value );
extern EX_CS_RECORD *IW_ExCSDeleteICP( EX_CS_REQ *req, int type, char *name );
extern int IW_ExCSAddICP( EX_CS_REQ *req, int type, char *name, char *value );
extern int IW_ExCSAddCacheUserGroup( EX_CS_REQ *req, int kind,
                                     char *sessionid, char *groupname );
extern int IW_ExCSDeleteCacheUserGroup( EX_CS_REQ *req, int kind,
                                        char *sessionid, char *groupname );
extern int IW_ExCSGetLoginUserCount( EX_CS_REQ *req );

```

```
extern EX_CS_RECORD *IW_ExCSModifyCacheUser_V2( EX_CS_REQ *req, int kind,
                                                char *sessionid, char *column,
                                                char *value);
extern int IW_ExCSAddCacheUserGroup_V2( EX_CS_REQ *req, int kind,
                                         char *sessionid, char *groupname);
extern int IW_ExCSDeleteCacheUserGroup_V2( EX_CS_REQ *req, int kind,
                                           char *sessionid, char *groupname);

#endif /* #ifndef IWINTFACE_H */
```

9.2.3 メイクファイル (HP-UX Itanium 64Bit 版 : Makefile)

```
CC      = cc

CCFLAGS      = -D_UNIX -D_HPUX_SOURCE -D_POSIX_C_SOURCE=199506L -Aa +e -
D_FILE_OFFSET_BITS=64 -z +DD64 +z

LIBS      =

INCLUDE      = -I./

MAKEFILE     = Makefile

OBJS      = iwintface.o

PROGRAM     = CertExit

SRCS      = iwintface.c

all:      $(PROGRAM)

$(PROGRAM):  $(OBJS)
            ld -b -o lib$(PROGRAM).sl $(OBJS) $(LIBS)

.c.o:
            $(CC) $(CCFLAGS) $(INCLUDE) -c $(SRCS)

clean::
            rm -f $(OBJS) lib$(PROGRAM).sl core
```

9.2.4 メイクファイル (Linux64Bit 版 : Makefile)

```
CC      = gcc

CCFLAGS = -m64 -fPIC -DLinux -D_FILE_OFFSET_BITS=64 -D_LARGEFILE_SOURCE
-D_GNU_SOURCE

LIBS     =

INCLUDE  = -I./

MAKEFILE = Makefile

OBJS     = iwintface.o

PROGRAM  = CertExit

SRCS     = iwintface.c

all:      $(PROGRAM)

$(PROGRAM): $(OBJS)
            gcc -shared -o lib$(PROGRAM).sl $(OBJS) $(LIBS)

.c.o:
            $(CC) $(CCFLAGS) $(INCLUDE) -c $(SRCS)

clean::    rm -f $(OBJS) lib$(PROGRAM).sl core
```