



ライセンス管理ガイド

FLEXnet Publisher ライセンス・ツールキット 11.6

FNP-116-LA02

法的情報と Acresto への連絡方法

部品番号： FNP-116-LA02

製品リリース日： 2008 年 5 月

Acresto Software への連絡方法

Acresto の Web サイト <http://www.acresto.com/company.htm> にアクセスすると、世界中のどこからでも弊社に連絡できます。

著作権情報

Copyright © 2007-2010 Acresto Software Inc. and/or InstallShield Co. Inc. All Rights Reserved.

この製品には、Acresto Software Inc.、InstallShield Co. Inc.、およびそれぞれのライセンスが所有する独占的かつ機密の技術、情報、創作物などが含まれています。このような技術の全体または一部を、Acresto Software Inc. または InstallShield Co. Inc.、あるいは両社の書面による事前の許可なく、使用、複製、出版、配布、表示、改変、転送することは、その形態や手段にかかわらず固く禁じられています。Acresto Software Inc. または InstallShield Co. Inc.、あるいはこの両社が書面で明確に提供した場合を除いて、この技術を有することは、禁反語、黙示、その他を問わず、Acresto Software Inc. または InstallShield Co. Inc. の知的所有権の定めるライセンスや権利の授与とは解釈されません。

技術や関連情報の複製が Acresto Software Inc. または InstallShield Co. Inc. によって許可された場合は、この著作権所有権情報の全文を表示する必要があります。

商標

Acresto、FLEXenabled、FLEX/m、FLEXnet、FLEXcertified、FLEXnet Connect、FLEXnet Connector、FLEXnet Manager、FLEXnet Publisher、Globetrotter、InstallShield、InstallShield Developer、InstallShield DevStudio、および InstallShield Professional は、Acresto Software Inc. または InstallShield Co. Inc. あるいは両社の米国またはその他の国における登録商標または商標です。このマニュアルに記載されているその他のブランド名および製品名はすべて、各所有者の商標または登録商標です。

制限付き権利についての説明

本ソフトウェアおよびドキュメントは、48 C.F.R. Section 2.101 で定義されている「商用品目 (commercial items)」に該当し、48 C.F.R. Section 12.212 および 48 C.F.R. Section 227.2702 で定義されている「商用コンピュータ・ソフトウェア (commercial computer software)」および「商用コンピュータ・ソフトウェア説明書 (commercial computer software documentation)」から構成されています。48 C.F.R. Section 12.212 または 48 C.F.R. Section 227.2702-1 ~ 227.2702-4 に従い、米国政府のエンド・ユーザに対するこの商用コンピュータ・ソフトウェアおよび商用コンピュータ・ソフトウェア説明書の使用許可は、(A) 商用品目としてのみ付与され、(B) このソフトウェアに関する Acresto Software の標準的な商用契約の条項に従う他のすべてのエンド・ユーザに付与される権利のみが付与されるものとします。明示されていない権利は、米国著作権法によって保護されています。

目次

	法的情報と Acrezzo への連絡方法	2
	このマニュアルの最新情報	9
1	ライセンス・サーバを実行するマシンの選択	11
	ライセンス・サーバによって使用されるリソース	11
	ソケット	11
	CPU 時間	12
	ディスク領域	12
	メモリ	12
	ネットワーク帯域幅	12
	リモートでマウントされたディスク	13
2	lmadmin – GUI ベースのライセンス・サーバ・マネージャ	15
	lmadmin のライセンス管理機能	16
	ライセンス・サーバ・マネージャの手動開始	18
	ライセンス・サーバ・マネージャの手動停止	18
	オペレーティング・システムのサービスとしてのライセンス・サーバのインストール	19
3	lmgd – ライセンス・サーバ・マネージャ	21
	lmgd のコマンド・ライン構文	22
	UNIX プラットフォームにおけるライセンス・サーバ・マネージャの起動	23
	手動での開始	23
	自動での開始	24

Windows におけるライセンス・サーバ・マネージャの起動	25
コマンド・ラインからの手動開始.....	25
ライセンス・サーバ・マネージャを Windows サービスとして設定する.....	25
ライセンス・サーバ・マネージャ・サービスの開始遅延の設定.....	26
lmtools ユーティリティを使用したライセンス・サーバの手動開始.....	27
システムの起動時にライセンス・サーバが自動的に開始します.....	29
4 ライセンス管理ツールの使用法	31
コマンド・ライン・ユーティリティ	31
lmutil の一般的な引数.....	32
lmborrow.....	33
借用の開始.....	33
借用ライセンスの設定のクリア.....	34
借用ライセンスのステータスの確認.....	34
借用ライセンスの早期返却.....	34
lmdiag.....	35
lmdown.....	36
lmhostid.....	37
lminstall.....	39
lmnewlog.....	40
lmpath.....	40
lmremove.....	42
lmreread.....	43
lmstat.....	44
lmswitch.....	46
lmswitchr.....	47
lmver.....	48
lmtools (Windows のみ)	48
5 IPv6 のサポート	51
IPv6 をサポートする機能.....	51
混合プロトコル環境でのライセンス・サーバの配備.....	52
IPv6 アドレスでのワイルドカードの使用.....	54
6 3 サーバ冗長化の使用	55
3 サーバ冗長化の概要	55
この構成のライセンス・サーバの管理.....	58
ライセンス・サーバに対する 3 サーバ冗長化の設定.....	59
3 サーバ冗長化による他の機能の使用.....	60
トラブルシューティングのヒントと制限事項.....	62

7	ライセンス・ファイルの読み込み	63
	ライセンス・ファイルの場所の指定	63
	環境変数を使用したライセンス検索パスの設定	65
	ライセンス・ファイルのフォーマットの概要	65
	ライセンス・ファイルの構文	66
	SERVER 行.....	66
	VENDOR 行	69
	USE_SERVER 行.....	70
	FEATURE 行および INCREMENT 行	70
	ソートの規則	74
	FEATURE および INCREMENT 行のフォーマットの変更	75
	PACKAGE 行	75
	UPGRADE 行	77
	10 進形式の FEATURE 行	78
	ライセンス・ファイルの行の順序.....	78
8	ライセンス・モデル	79
	フローティング (同時) ・ライセンス.....	79
	ノードロック ・ライセンス.....	80
	ノードロック ・ライセンスとフローティング ・ライセンスの混在	80
	カウント ・ライセンスと非カウント ・ライセンスの比較.....	81
	モバイル ・ライセンス	81
	ラップトップ ・コンピュータへのノードロック	82
	FLEXnet ID ドングルへのノードロック.....	82
	FLOAT_OK キーワード付き FLEXnet ID ドングルへのノードロック	82
	FLOAT_OK キーワード付きの FLEXnet ID ドングルの開始	82
	FLOAT_OK を使用する FLEXnet ID ドングルライセンスの返却	83
	FLOAT_OK 付きの FLEXnet ID ドングルの使用例	83
	BORROW によるライセンスの借用	84
	ライセンス借用の開始	84
	アプリケーション ・インタフェース.....	85
	Imborrow ユーティリティの実行.....	85
	LM_BORROW 環境変数の直接設定	85
	ライセンスの借用.....	86
	借用期間のクリア	86
	借用ステータスのチェック	86
	借用ライセンスの早期返却.....	87
	ライセンス借用のサポート	87
	ユーザ名へのノードロック	87
	プリペイド ・ライセンス ・プールからの充当	88

9	複数のソフトウェア・パブリッシャからのライセンスの管理	89
	複数ライセンス管理方法の概要	89
	複数のマシン	90
	ライセンス・サーバの起動	91
	単一マシンと複数のライセンス・サーバ・インスタンス	91
	ライセンス・サーバの起動	92
	単一マシンと単一ライセンス・サーバおよび複数ライセンス・ファイル	93
	ライセンス・サーバの起動	94
	複数のライセンス・ファイルの管理	94
	その他の注意事項	95
	ライセンス・ファイルの結合	95
	ライセンス・サーバの起動	96
	ライセンス・ファイル結合の条件	97
	ライセンス・ファイルの結合方法	98
	バージョン・コンポーネントの互換性	98
10	サポートされるプラットフォームの hostid	99
	hostid の形式	99
	マシンの hostid の取得	99
	特殊な hostid	101
11	トラブルシューティング	103
	一般的なトラブルシューティングのヒント	103
	FLEXLM_DIAGNOSTICS	104
	レベル 1 の内容	104
	レベル 2 の内容	105
	レベル 3 の内容 (バージョン 6.0 以降のみ)	106
12	オプション・ファイルの管理	107
	オプション・ファイルの作成	108
	オプション・ファイルの構文	108
	BORROW_LOWWATER	112
	DEBUGLOG	112
	EXCLUDE	113
	EXCLUDE_BORROW	113
	EXCLUDE_ENTITLEMENT	114
	EXCLUDEALL	115
	FQDN_MATCHING	115
	GROUP	117
	GROUPCASEINSENSITIVE	117
	HOST_GROUP	118
	INCLUDE	118
	INCLUDE_BORROW	119

INCLUDE_ENTITLEMENT	120
INCLUDEALL	121
LINGER	121
MAX.	122
MAX_BORROW_HOURS	123
MAX_OVERDRAFT	123
NOLOG	124
REPORTLOG	124
LM_PROJECT によるプロジェクトについてのレポート	125
RESERVE	125
TIMEOUT	126
TIMEOUTALL	127
ベンダ・デーモンによるオプション・ファイルの使用方法	127
オプション・ファイル内での優先度の規則	127
オプション・ファイルの例	128
簡単なオプション・ファイルの例	128
複数ユーザのアクセスの制限	128
EXCLUDE の例	129
EXCLUDE_ENTITLEMENT の例	129
INCLUDE の例	130
INCLUDE_ENTITLEMENT の例	130
13 環境変数	131
環境変数の設定方法	131
Windows レジストリ	131
優先度	131
環境変数	132
14 エラー・コード	135
エラー・メッセージのフォーマット	135
フォーマット 1 (ショート)	136
フォーマット 2 (ロング - バージョン 6.0 以降)	136
エラー・コードの説明	136
15 レポート・ログ・ファイル	145
レポート・ログ出力の管理	145
ベンダ・デーモンのレポート・ログ出力の有効化	146
ベンダ・デーモンのレポート・ログ出力のリダイレクト	146

16	デバッグ・ログ・ファイル.....	147
	デバッグ・ログ出力の管理.....	147
	ライセンス・サーバのデバッグ・ログ出力の取得.....	148
	特定のベンダ・デーモンのデバッグ・ログ出力の取得.....	148
	実行中のベンダ・デーモンのデバッグ・ログ出力のリダイレクト.....	148
	ベンダ・デーモンのデバッグ・ログ出力の出力規制.....	148
	デバッグ・ログ・メッセージ.....	149
	情報メッセージ.....	149
	設定問題のメッセージ.....	151
	デーモンのソフトウェア・エラーに関するメッセージ.....	152
17	FLEXnet のライセンス・バージョンの識別.....	153
	コンポーネント間のバージョンの互換性.....	153
	ライセンス・ファイルのバージョンの確認.....	154
	バージョンの概要.....	154
	索引.....	161

このマニュアルの最新情報

この項では、今回のリリースで更新されたマニュアルの内容について説明します。

新しいライセンス・サーバ、`lmadmin`

このリリースには GUI ベースのサーバ・マネージャが含まれています。使用できるプラットフォームは限られており、Windows 32 ビット、Mac OS X、および Red Hat Enterprise Linux 32 ビットです。このマニュアルの [lmadmin - GUI ベースのライセンス・サーバ・マネージャ](#) に `lmadmin` の概要が記載されています。`lmadmin` のインストール方法と使い方の詳細については、ライセンス・サーバ・マネージャ (`lmadmin`) インストール・ガイドと、`lmadmin` のオンライン・ヘルプを参照してください。

ライセンス・サーバ・マネージャの起動遅延

[ライセンス・サーバ・マネージャ・サービスの開始遅延の設定](#) では、その他の必要コンポーネントがすべてロードされてからライセンス・サーバ・マネージャが開始されるように `lmgrd` ライセンス・サーバを設定する方法について説明します。遅延が必要になる典型的なシナリオは、FLEXnet ID のドングルを使用してライセンス・サーバをマシンにロックする必要がある場合です。

1

ライセンス・サーバを実行するマシンの選択

この章では、ライセンス・サーバ・マシンとして使用するマシンを決定するためのガイドラインを示します。

ライセンス・サーバによって使用されるリソース

この項では、ライセンス・サーバによって使用されるリソースについて説明します。サーバ・システムを選択するときには、これらのリソースに対するシステムの制限についても考慮する必要があります。ライセンス数が少ない場合は（およそ 100 以下）、どのワークステーション上でもシステム制限が問題となることはほとんどありません。

ソケット

TCP/IP ポートを使うとき、ライセンス・サーバに接続している FLEXenabled アプリケーションのそれぞれが 1 つ以上のソケットを使用します。ソフトウェア・パブリッシャが FLEXnet Publisher ライセンス・ツールキットを実装している方法によっては、FLEXenabled アプリケーションに 1 つまたは複数のソケットが必要になる場合があります。この情報については、ソフトウェア・パブリッシャにお問い合わせください。ライセンス・サーバ・システムが使用できるソケットの数は、ファイル記述子に対するプロセスあたりのシステム制限によって決定されます。ライセンス・サーバが使用するソケットの総数は、それに接続している FLEXenabled アプリケーションに必要な合計ソケット数よりも若干多くなります。

1 台のマシン上でライセンス・サーバに必要なソケット数が過剰になった場合は、ライセンス・ファイルを別のサーバの複数のファイルに分割して、ネットワーク・トラフィックを軽減することをおすすめします（この場合、ソフトウェア・パブリッシャが新しいライセンスの発行に同意する必要があります）。ライセンス・ファイルを分割すると、FLEXenabled アプリケーションは LM_LICENSE_FILE 環境変数によるライセンス検索パスを使用して複数のサーバからライセンスをチェックアウトできます。

CPU 時間

クライアント数が少ない場合、ライセンス・サーバが使用する CPU 時間はわずかです。数日経過しても、サーバが使用する CPU 時間がわずか数秒の場合もあります。

クライアント数が多い場合や（クライアントはそれぞれライセンス・サーバとハートビート・メッセージを交換します）、チェックアウト／チェックインの活動が 1 秒あたり数百回の頻度で行われる場合は、サーバが使用する CPU 時間が極端に増えることもありますが、このような場合でも、CPU 使用率はそれほど高くなりません。この場合は、選択したシステムに十分な CPU サイクルを確保してください。

ディスク領域

ライセンス・サーバによって作成される出力ファイルは、デバッグとレポートのログ・ファイルだけです。レポート・ログ・ファイルは、FLEXnet Manager による正確な使用状況レポートの作成に使用されます。ライセンス処理の活動が多い場合は、このログ・ファイルのサイズが非常に大きくなります。これらのファイルをどこに出力し、どのようなサイクルで切り替えおよび保管を行うかを検討する必要があります。ディスク領域の余裕がない場合は、ログ・ファイルの出力を抑制することもできます。

ネットワーキングへの依存を避けるために、ログ・ファイルはサーバ・システムのローカル・ファイルとして出力することをおすすめします。

参照

[環境変数を使用したライセンス検索パスの設定](#)

[UNIX プラットフォームにおけるライセンス・サーバ・マネージャの起動](#)

[レポート・ログ・ファイル](#)

[デバッグ・ログ・ファイル](#)

メモリ

ライセンス・サーバが使用するメモリは少量です。ベンダ・デーモンはそれぞれが約 2 MB を使用しますが、ベンダ・デーモンのメモリ使用量は、ライセンス・ファイルのサイズ、オプション・ファイルのサイズ、および同時ユーザ数に応じて増加します。GUI ベースのサーバ・マネージャ、ladmin は、標準的な使用時に 7～10 MB のメモリを使用します。Solaris の場合、コマンド・ラインのライセンス・サーバ・マネージャ、lmgd は約 2 MB 使用します。

ネットワーク帯域幅

FLEXnet Publisher ライセンス・ツールキットからネットワークを介して送信されるデータは、比較的少量です。チェックアウトやチェックインなどのトランザクション 1 回につき転送されるデータ量は通常 1 KB 未満です。つまり、FLEXnet Publisher ライセンス・ツールキットは、クライアント数が少なければ、ダイヤルアップ SLIP 回線などの低速ネットワークでも効率的に実行できます。

FLEXenabled アプリケーションの数が数百にも上ると、これらのアプリケーションのそれぞれがベンダ・デーモンとの間でハートビート・メッセージを交換するため、ネットワーク帯域幅の使用量が増えます。この場合は、FLEXenabled アプリケーションとサーバを同じローカル・エリア・ネットワークで実行し、ライセンスを 2 台のサーバの 2 つのファイルに振り分けなければならない可能性もあります。ユーザは LM_LICENSE_FILE 環境変数でライセンス検索パスを使用して、両方のサーバに効率的にアクセスできます。ネットワーク通信が低速のときや、FLEXenabled クライアントがダイヤルアップ・リンクを介してネットワークに接続している場合は、企業でパフォーマンスの問題が発生する可能性があります。

GUI ベースのライセンス・サーバ・マネージャ、lmadmin を使用しているときには、lmadmin ユーザ・インタフェースに接続しているクライアントを考慮する必要があります。クライアントの数とページの更新頻度によっては、ネットワーク・トラフィックに著しい負荷がかかることがあります。

参照

[ライセンス・ファイルの場所の指定](#)

リモートでマウントされたディスク

ライセンス・サーバを実行するときには、リモートでマウントされたディスクを使用しないようにおすすめます。つまり、lmadmin または lmgrd、ベンダ・デーモン、ライセンス・ファイル、およびデバッグとレポートのログ・ファイルのすべてを、ローカルでマウントされたディスクに配置することをおすすめます。これらのファイルのいずれかがリモートでマウントされたディスクにある場合は、障害の発生点が 2 倍に増えることとなります。これによって、すべてのライセンスを一時的に失う可能性があります。ファイルをすべてローカルでマウントすると、サーバが実行している限りライセンスを使用できます。ファイルが別のシステムにあると、ライセンス・サーバまたはファイル・サーバに障害が発生した場合に、ライセンスを使用できなくなる場合があります。

第 1 章：ライセンス・サーバを実行するマシンの選択
ライセンス・サーバによって使用されるリソース

ladmin – GUI ベースのライセンス・サーバ・マネージャ

ライセンス・サーバ・マネージャは、ライセンス・サーバを構成しているコンポーネントの 1 つです (もう一つはベンダ・デーモン)。ライセンス・サーバ・マネージャは FLEXenabled アプリケーションとの最初の通信を処理し、該当するベンダ・デーモンに接続情報を渡します。ライセンス・サーバ・マネージャの目的は以下のとおりです。

- さまざまなソフトウェア・パブリッシャからライセンス権利を提供するために必要なベンダ・デーモンの開始と維持。
- 適切なベンダ・デーモンに対するアプリケーション・チェックアウトなどの要求の参照。

ライセンス・サーバ・マネージャには次の 2 種類のバージョンがあります。

- **ladmin** – グラフィカル・ユーザ・インタフェースを備えた最新のライセンス・サーバ・マネージャ。
- **lmgd** – コマンド・ライン・インタフェースを備えたオリジナルのライセンス・サーバ・マネージャ。

この項では ladmin について説明します。lmgd については、「[lmgd – ライセンス・サーバ・マネージャ](#)」を参照してください。

ladmin はライセンス・サーバとベンダ・デーモンをよりよく管理する手段を提供しています。以下に、改良された機能の概要を述べます。ladmin の詳細については、ライセンス・サーバ・マネージャ (ladmin) インストール・ガイドと、ladmin から利用できるオンライン・ヘルプを参照してください。ladmin および付属のマニュアルは Acreso のダウンロード・サイトから無償で入手できます。

ladmin の機能

- ベンダ・デーモンとライセンス・サーバ・マネージャの直接設定 – ライセンス・サーバのポート番号、ベンダ・デーモンのパスとポート、および 3 サーバ冗長ポートは、ライセンス・ファイルの編集なしに設定できます。
- 設定可能なアラート – 問題について警告するアラートを発行するように ladmin を設定できます。ライセンスの失効期日、ライセンスの使用不可、ベンダ・デーモンのステータスなどがその例です。

- ライセンス権利のステータス表示 – 使用可能なすべてのライセンス権利と使用中のライセンス権利の表示を設定できます。この表示には、ライセンス・ファイルと信頼できる記憶領域の同時（フローティング）ライセンスをすべて含めることができます。また、ライセンス・サーバでアクティブ化可能ライセンス（信頼できる記憶領域に保管）が使用可能な場合は、それらも含めることができます。
- GUI のボタンはコマンド・ライン・ユーティリティの「Stop Server」や「Reread License Files」に代わるものです。Imadmin から直接使用できるライセンス管理機能については、「Imadmin のライセンス管理機能」のリストを参照してください。
- ライセンス・ファイルの最小限の編集 – オプション・ファイルの指定には編集が必要です。

このリリースの Imadmin では使用可能なプラットフォーム数が限られています。詳細については、ソフトウェア・パブリッシャに問い合わせるか、Acesso のダウンロード・サイトを参照してください。Imadmin はバージョン 9.2 以降のライセンス・コンポーネントと互換性があります。ライセンスしたアプリケーションで提供されているライセンス・コンポーネントのバージョンを確認する方法については、「コンポーネント間のバージョンの互換性」を参照してください。

Imadmin のライセンス管理機能

Imadmin は、以前コマンドライン・ベースの管理ユーティリティや、Windows プラットフォームでは Imtools で提供されていたライセンス管理機能もいくつか提供しています。次の表は、Imadmin 内で提供されている機能のうち、ライセンス管理ユーティリティが提供する機能に取って代わるものを示します。

表 2-1: Imadmin のライセンス管理機能

Imadmin の機能	説明	置き換えるユーティリティ
ダッシュボード – ライセンス	使用可能なライセンス権利と使用中のライセンス権利の詳細を表示します。	Imstat
ベンダ・デーモン設定 – 管理 – 停止	ベンダ・デーモンを停止します。	Imdown – 一部の使用ケース
ベンダ・デーモン設定 – 管理 – ライセンス・ファイルの再読み込み	Imadmin 設定に含まれているライセンス・ファイルからライセンス権利を再読み込みします。ライセンス・ファイルの内容が更新される場合にのみ必要です。	Imreread
ベンダ・デーモン設定 – 管理 – レポート・ログの切り替え	レポート・ログを新しいファイル名に切り替えます。	Imswitchr
サーバ設定 – サーバの停止	ライセンス・サーバを停止します。Imadmin のデフォルト設定ではこのボタンが無効になります。有効にするには、-allowStopServer 引数を使用して Imadmin を開始します。	Imdown – 一部の使用ケース

次の表は、ladmin の使用時に不要になったコマンドライン・ユーティリティと必要なユーティリティについて示します。

表 2-2: ladmin で使用するライセンス管理ユーティリティ

ユーティリティ	ladmin 使用時に必要
lmborrow	ライセンス・ファイルと借用機能でライセンス権利を使用する場合は必要。
lmdiag	ライセンスのチェックアウト問題を診断するために必要。
lmdown	通常は不要。ladmin のデフォルト設定は lmdown の動作を無効にします。有効にするには、-allowStopServer 引数を使用して ladmin を開始します。
lmhostid	通常はホスト ID として使用されるさまざまな識別を含め、実行しているシステムに関する情報を表示するので、通常 ladmin には不要。クライアント・システムのホスト ID を判別するために必要。
lminstall	必要。ライセンス・ファイルのフォーマットを変換します。
lmnewlog	必要。オプション・ファイルでレポート・ログのファイル名を編集したくないという理由で lmswitchr の代わりにこの機能を使用して新しいレポート・ログに変更する場合は必要。
lmpath	必要。ライセンス・ファイルのパス設定をユーザが直接制御できます。
lmremove	必要。ハングしたライセンスを解放して、フリー・ライセンスのプールに戻します。ladmin のデフォルト設定では lmremove の動作が無効になります。有効にするには、-allowLicenseReclaim 引数を使用して ladmin を開始します。
lmreread	通常は不要。
lmstat	追加情報（借用や予約など）を表示する場合にのみ必要。
lmswitch	必要。デバッグ・ログの場所とサイズを制御します。
lmswitchr	不要。
lmver	必要。ライブラリまたはバイナリ ファイルのバージョンをレポートします。ladmin のバージョンは、-version 引数を使用して開始すると確認できます。

ライセンス・サーバ・マネージャの手動開始

ライセンス・サーバは以下のいずれかの方法で開始できます。

- Windows プラットフォームでは、Windows エクスプローラでインストール先ディレクトリを開き、ladmin.exe ファイルをダブルクリックします。このメカニズムでは、デフォルト以外のコマンドライン・オプションを指定できません。
- ルート・インストール・ディレクトリから ladmin コマンドを実行します。使用可能なコマンドライン・オプションを確認するには、次のコマンドを実行します。

`ladmin -help`

ヘルプには、デフォルトのオプションと、どのオプションが永続的 (ladmin の後のインスタンスでも効力のあるオプション) であるかが示されます。

- シェル・スクリプト・ファイル (Unix) またはバッチ・ファイル (Windows) を作成します。これは、目的のコマンドライン・オプションを使用して ladmin コマンドを実行してから、そのファイルを実行します。



注意：ユーザ・インタフェースのデフォルトのライセンス・サーバ・ポートか HTTP ポートが使用中の場合、サーバは開始しません。競合する場合は、`-licPort` オプションまたは `-webPort` オプション、あるいはその両方を使用して、デフォルトを上書きします。

ライセンス・サーバ・マネージャの手動停止

`allowStopServer` コマンドライン・オプションは、ladmin ユーザ・インタフェースの **[Stop Server]** ボタンの表示を切り替えます。デフォルトでは **[Stop Server]** ボタンが表示されません。この場合、管理者以外がモニタの目的でインタフェースを使用できます。

`allowStopServer` コマンドライン・オプションは持続的です。`-allowStopServer yes` オプションを使用して ladmin が開始された場合や、以前に開始されたことがある場合、ライセンス・サーバの管理インタフェースの管理セクションにある **[Server Configuration]** タブに **[Stop Server]** ボタンが表示されます。**[Stop Server]** ボタンをクリックすると、ライセンス・サーバ・マネージャ (ladmin) とすべてのベンダ・デーモンが停止します。

コマンドライン・オプション `-allowStopServer no` を使用して ladmin が開始され場合、持続的なコマンドライン・オプション `-allowStopServer yes` が使用されたことがない場合、または `-allowStopServer no` が `-allowStopServer` オプションの最近の使用であった場合は、ライセンス・サーバ管理インタフェースを使用してライセンス・サーバを停止できません。この場合、ライセンス・サーバを停止するには、ladmin プロセスを停止する必要があります。

Unix システムでは、`ps` ユーティリティを使用してプロセスを識別し、それを `kill` コマンドで強制終了します (`kill -9` は使用せず、デフォルト・シグナルのみで `kill` を使用してください。そうでないと、ライセンス・サーバが正常に停止しません)。Windows システムでは、タスク・マネージャを使用して `ladmin.exe` プロセスを識別し、それを終了します。

管理インタフェースからライセンス・サーバを再起動することはできません。前のセクションで説明した方法でライセンス・サーバを再起動してください。

オペレーティング・システムのサービスとしてのライセンス・サーバのインストール

lmadmin ライセンス・サーバ・マネージャの開始と停止は手動でもできますが、オペレーティング・システムの再起動時に自動的に開始するように、オペレーティング・システムのサービスとしてインストールすることをおすすめします。

Windows システム

Windows システムでは、lmadmin ライセンス・サーバをサービスとしてインストールできます。Windows Administrators グループのユーザのみがこの操作を実行できます。[スタートアップの種類]を[自動]に設定し、システムの再起動時にサービスが自動的に開始するようにします。以下のライセンス・サーバ・マネージャ (lmadmin) コマンドライン・オプションを使用して、サービスのインストールとアンインストールを行います (次の表を参照)。



Important: ライセンス・サーバ・マネージャを Windows のサービスとしてインストールするコマンドを実行した後、ライセンス・サーバ・マネージャは自動的に開始します。初回は Windows Services Console を使用してサービスを開始する必要があります。

表 2-3: lmadmin を Windows のサービスとして設定するために使用するためのコマンドライン・オプション。

lmadmin コマンドライン・オプション	説明
-installService	ライセンス・サーバ・マネージャを実行する Windows サービスを、定義した名前で作成します。
-removeService	定義した名前の Windows サービスをアンインストールします。削除する前に、必ずサービスを停止してください。
-delay nn	これは -installService オプションと一緒に使用されます。サービスを開始してから実際に実行し始めるまでの遅延の秒数 (nn) を設定します。遅延が必要になる典型的なシナリオは、FLEXnet ID ドングルを使用してライセンス・サーバをマシンにロックする必要がある (FLEXid が SERVER 行で使用されている) 場合です。このシナリオでは、ドングル・デバイス・ドライバが正常にロードされる前にライセンス・サーバがロードされるため、システムの再起動時にライセンス・サーバが開始に失敗することがあります。

Red Hat Linux システムと Sun Solaris システム

Red Hat Linux システムと Sun Solaris システムでは、インストールした `/examples` ディレクトリにサブディレクトリの `/service` が入っています。`/service` ディレクトリにはシェルスクリプト・ファイルの `ladmin` が入っています。この `ladmin` ファイルをライセンス・サーバ・マネージャと間違えないでください。スクリプト・ファイルには、Red Hat Linux システムまたは Sun Solaris システムのどこにこのファイルをインストールするかの詳細を含め、インストール手順が含まれています。このスクリプトはテスト済みで、大半のインストールに適していますが、特定の要件に合わせて変更が必要な場合もあります。

Mac OS システム

Mac OS システムでは、管理者は `/Library/StartupItems/LMadmin` などのディレクトリに、独自の起動スクリプトを作成する必要があります。インストールされる `/examples/service/ladmin` スクリプトは、Linux システムと Solaris システムにインストールされるスクリプトで、参照のために提供されているだけで、Mac OS システムでは正常に動作しません。Mac OS システムにシステム・サービスとして実行ファイルをインストールする方法の詳細については、<http://www.oreilly.com/pub/a/mac/-2003/10/21/startup.html> などで公開されているリファレンスを参照してください (Acesso Software では、そのようなリファレンスから得た情報の正確さや、作成した起動スクリプトについての責任は負いません)。

lmgrd – ライセンス・サーバ・マネージャ

ライセンス・サーバ・マネージャは、ライセンス・サーバを構成しているコンポーネントの1つです（もう一つはベンダ・デーモン）。ライセンス・サーバ・マネージャは FLEXenabled アプリケーションとの最初の通信を処理し、該当するベンダ・デーモンに接続情報を渡します。ライセンス・サーバ・マネージャの目的は以下のとおりです。

- lmgrd の開始に使用されるライセンス・ファイルの VENDOR 行に列挙されているすべてのベンダ・デーモンの開始と維持。
- 適切なベンダ・デーモンに対するアプリケーション・チェックアウトなどの要求の参照。

lmgrd はライセンス・サーバ・マネージャのアプリケーションベースのバージョンです。ほとんどのプラットフォームでは、コマンドラインから制御します。Windows では、GUI ツールの lmtools があり、lmgrd の管理に使用できます。

新しい lmgrd は古いベンダ・デーモンや FLEXenabled アプリケーションでも使用できますが、新しいベンダ・デーモンまたは FLEXenabled アプリケーションは古い lmgrd で正しく機能しない場合があります。常に、できるだけ新しいバージョンの lmgrd ユーティリティを使用してください。最新バージョンはダウンロード・サイトから入手できます。詳細については、「[コンポーネント間のバージョンの互換性](#)」を参照してください。

Imgrd のコマンド・ライン構文

Imgrd を起動すると、ベンダや機能に関する情報が入っているライセンス・ファイルが検出され、それらのベンダ・デーモンが開始します。

使用法

```
Imgrd [-c license_file_list] [-l [+]debug_log_path]  
      [-2 -p] [-local] [-x lmdown] [-x lmremove] [-z] [-v] [-help]
```

各パラメータの意味は、次のとおりです。

表 3-1: Imgrd のコマンド・ライン構文の使用法

用語	説明
-c <i>license_file_list</i>	指定したライセンス・ファイルを使用します。
-l [+] <i>debug_log_path</i>	デバッグ情報を <i>debug_log_path</i> ファイルに書き込みます。このオプションは数字の 1 ではなく、英字の l を使用します。 <i>debug_log_path</i> の前に + 文字を付けると、ログにエントリが追加されます。このファイルの詳細については、「 デバッグ・ログ・ファイル 」を参照してください。
-2 -p	lmdown、lmreread、および lmremove の使用を管理者 (デフォルトでは root) だけに許可します。Imadmin という UNIX グループがある場合は、そのグループのメンバだけに使用を許可します。root がこのグループのメンバでない場合、上記のユーティリティに対するパーミッションは root にも与えられません。-2 -p を指定して Imgrd を開始すると、Windows ユーザは lmdown を使用してライセンス・サーバを停止できなくなります。
-local	Imgrd を実行しているマシンからのみ lmdown コマンドと lmreread コマンドの実行を許可します。
-x lmdown	lmdown コマンドを無効にします (どのユーザも lmdown を実行できなくなります)。lmdown を無効にした場合は、kill pid を使用して Imgrd を停止するか (UNIX の場合)、Windows タスク・マネージャまたは Windows サービスから Imgrd とベンダ・デーモン・プロセスを停止します。UNIX では、kill コマンドに -9 引数を使用しないでください。
-x lmremove	lmremove コマンドを無効にします (どのユーザも lmremove を実行できなくなります)。
-z	フォアグラウンドで実行します。デフォルトでは、バックグラウンドで実行します。-l <i>debug_log_path</i> を指定すると、ウィンドウは使用されません。-l 引数を指定しなければ、Imgrd とベンダ・デーモンごとに別のウィンドウが使用されます。
-v	Imgrd のバージョン番号と著作権情報を表示して終了します。
-help	使用方法を表示して終了します。

UNIX プラットフォームにおけるライセンス・サーバ・マネージャの起動

ライセンス・ファイル内のライセンスをカウントする場合は (ライセンス・カウント > 0)、**FLEXenabled** アプリケーションを使用する前にライセンス・サーバ・マネージャ (ライセンス・サーバ) を起動する必要があります。

ライセンス・サーバ・マネージャ **lmgrd** は、コマンド・ラインから手動で開始するか、システム起動時に自動的に開始します。以下の項で、これらの方法を説明します。



注意: **lmgrd** は、ライセンス・ファイル内の **SERVER** 行で指定されているシステムでのみ開始してください。

3 台の冗長サーバで構成されるライセンス・サーバを実行している場合は、ファイル・サーバではなく、各システムでローカルにライセンス・ファイルの同一コピー (さらに **lmgrd** およびベンダ・デーモンのバイナリ) を管理する必要があります。このようにしないと、これらのファイルを格納しているファイル・サーバがシングル・ポイント障害の発生点になるため、冗長サーバを使用するメリットが失われます。

手動での開始

UNIX コマンド・ラインから **lmgrd** を手動で開始するには、次の構文を使用します。

```
lmgrd -c license_file_list -L [+]debug_log_path
```

各パラメータの意味は、次のとおりです。

- **license_file_list** は次の項目の 1 つまたは複数です。
 - ライセンス・ファイルのフル・パス
 - ディレクトリ (そのディレクトリ内の *.lic という名前のファイルがすべて使用される)
 - **debug_log_path** は、デバッグ・ログ・ファイルのフル・パスです。

debug_log_path の前に + 文字を付けると、ログのエントリが追加されます。

root 以外のユーザとして **lmgrd** を開始します。**root** として開始したプロセスにはセキュリティ上のリスクが生じることがあります。**lmgrd** を **root** ユーザが開始する必要がある場合は、**su** コマンドを使用して **lmgrd** を非特権ユーザとして実行します。

```
su username -c "lmgrd -c license_file_list -l debug_log_path"
```

ここで、**username** は非特権ユーザです。ライセンス・ファイルに列挙されているベンダ・デーモンが **username** にパーミッションを実行したことを確認する必要があります。ライセンス・ファイル内のベンダ・デーモンのパスは、すべて **VENDOR** 行に列挙されています。

自動での開始

UNIX では、該当するブート・スクリプト (/etc/rc.boot、/etc/rc.local、/etc/rc2.d/Sxxx、/sbin/rc2.d/Sxxx など) を編集します。次のようなコマンドを含めます。詳細については、次の注意事項を参照してください。

```
/bin/su daniel -c 'echo starting lmgrd > ¥  
/home/flex1m/v11/hp700_u9/boot.log'  
  
/bin/nohup /bin/su daniel -c 'umask 022; ¥  
/home/flex1m/v11/hp700_u9/lmgrd -c ¥  
/home/flex1m/v11/hp700_u9/license.dat >> ¥  
/home/flex1m/v11/hp700_u9/boot.log'  
  
/bin/su daniel -c 'echo sleep 5 >> ¥  
/home/flex1m/v11/hp700_u9/boot.log'  
  
/bin/sleep 5  
  
/bin/su daniel -c 'echo lmdiag >>¥  
/home/flex1m/v11/hp700_u9/boot.log'  
  
/bin/su daniel -c '/home/flex1m/v11/hp700_u9/lmdiag -n -c¥  
/home/flex1m/v11/hp700_u9/license.dat >> ¥  
/home/flex1m/v11/hp700_u9/boot.log'  
  
/bin/su daniel -c 'echo exiting >>¥  
/home/flex1m/v11/hp700_u9/boot.log'
```

以下は、このスクリプト作成時の注意事項です。

- ブート時にはパスが仮定されないため、すべてのパスをフル・パスで指定します。
- パスが仮定されないため、ベンダ・デーモンは lmgrd と同じディレクトリにある必要があります。あるいは、ライセンス・ファイル内の VENDOR 行を編集してベンダ・デーモンのフル・パスを追加する必要があります。
- su コマンドを使用して、lmgrd を root 以外のユーザ **daniel** として実行します。root パーミッションを必要としないプログラムを root として実行するとセキュリティ上のリスクが生じるため、lmgrd を root として実行することはおすすめできません。lmgrd は root パーミッションを必要としません。
- **daniel** には csh のログインがあるため、**daniel** として実行されるすべてのコマンドは、csh 構文に従う必要があります。**daniel** として実行されないコマンドは、ブート・スクリプトで使用されるため、すべて /bin/sh 構文に従う必要があります。
- 一部のオペレーティング・システム、特に HP-UX では、nohup と sleep を使用する必要があります。Solaris など、一部のオペレーティング・システムでは必要ありませんが、すべてのシステムで使用する方が安全です。
- lmdiag は診断ツールとして使用され、サーバの実行とライセンスの使用状況を確認します。



注意：システムを再起動するまでベンダ・デーモンは開始しません。

Windows におけるライセンス・サーバ・マネージャの起動

この項では、コマンド・ラインから手動で起動する手順と、ライセンス・サーバ・マネージャ (lmgrd) をサービスとして設定する方法を説明します。

コマンド・ラインからの手動開始

コマンド・ラインから lmgrd を開始するには、次の手順に従います。

1. 次の構文を使用して、Windows コマンド・シェルから lmgrd をアプリケーションとして開始します。

```
C:¥fnp> lmgrd -c license_file_list -L [+]debug_log_path
```

各パラメータの意味は、次のとおりです。

- `license_file_list` は次の項目の 1 つまたは複数です。
 - ライセンス・ファイルのフル・パス
 - ディレクトリ (そのディレクトリ内の *.lic という名前のファイルがすべて使用されます)
- `debug_log_path` は、デバッグ・ログ・ファイルのフル・パスです。
`debug_log_path` の前に + 文字を付けると、ログのエントリが追加されます。

パス名の途中にスペースがあるときは、パスの全体を引用符で囲む必要があります。

Windows の場合、lmgrd をサービスとしてインストールすると、ユーザ・インタフェースから開始と停止を実行したり、バックグラウンドで実行したりできます。

ライセンス・サーバ・マネージャを Windows サービスとして設定する

ライセンス・サーバ・マネージャ (lmgrd) をサービスとして設定するには、管理者権限が必要です。サービスは `LocalSystem` アカウントで実行する必要があります。このアカウントは、このユーティリティをサービスとして実行するために必要です。

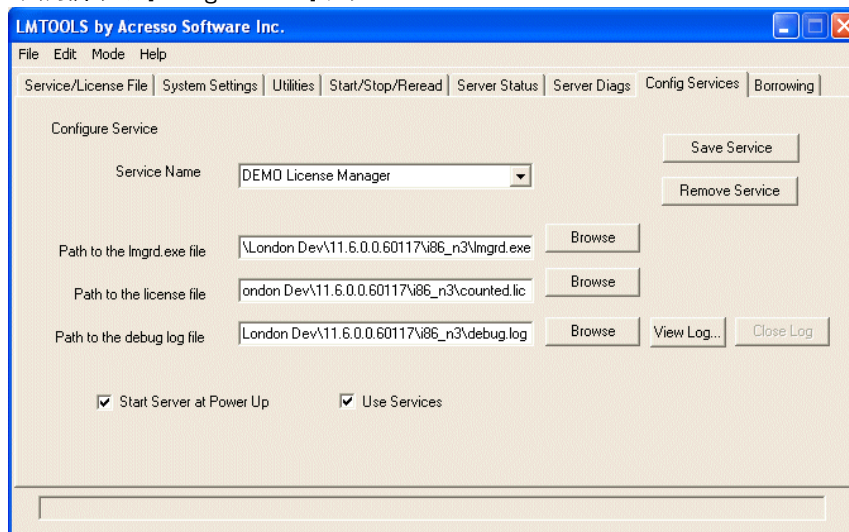


Task ライセンス・サーバをサービスとして設定するには、次の手順に従います。

1. `lmtools` ユーティリティを実行します。
2. **[Configuration using Services]** ボタンをクリックし、**[Config Services]** タブをクリックします。
3. **[Service Name]** 内に、定義するサービスの名前を入力します (たとえば、**DEMO License Manager**)。このフィールドを空白にした場合、サービスの名前は「**FLEXnet Licensing Service**」になります。
4. **[Path to the lmgrd.exe file]** フィールドに、このライセンス・サーバの `lmgrd.exe` を直接入力または参照により入力します。

5. **[Path to the license file]** フィールドに、このライセンス・サーバのライセンス・ファイルを直接入力または参照により入力します。
6. **[Path to the debug log file]** 内に、このライセンス・サーバが書き込むデバッグ・ログ・ファイルを直接入力するか参照により入力します。デバッグ・ログ・ファイル名の前にプラス記号(+)を付けると、ログのエントリが追加されます。デバッグ・ログ・ファイルのデフォルトのロケーションは c:\winnt\System32 フォルダです。別のロケーションを指定する場合は、完全修飾パスを指定する必要があります。
7. 新しい **DEMO License Manager** サービスを保存するには、**[Save Service]** をクリックします。

図 3-1: 入力済みの [Config Services] タブ



ライセンス・サーバ・マネージャ・サービスの開始遅延の設定

ライセンス・サーバを開始する前に他のドライバやサービスの開始を待つ必要がある状況では、ライセンス・サーバ・サービス開始前の遅延を設定できます。遅延が必要になる典型的なシナリオは、FLEXnet ID ドングルを使用してライセンス・サーバをマシンにロックする必要がある (FLEXid が SERVER 行で使用されている) 場合です。このシナリオでは、ドングル・デバイス・ドライバが正常にロードされる前にライセンス・サーバがロードされるため、システムの再起動時にライセンス・サーバが開始に失敗することがあります。



Task ライセンス・サーバ・マネージャ・サービスの開始遅延を設定するには、次の手順に従います。

1. ライセンス・サーバ・マネージャをサービスとして設定します (「[ライセンス・サーバ・マネージャを Windows サービスとして設定する](#)」)。
2. ライセンス・サーバ・マネージャ・サービスのレジストリ・エントリを次の場所で探します。

HKEY_LOCAL_MACHINE\SOFTWARE\FLEX\lm License Manager\service_name

service_name はライセンス・サーバ・マネージャ・サービスの名前です。

3. オプションで、20 秒を超える遅延を設定するには、レジストリ・エントリに文字列値を追加して、このエントリのフィールドを次のように設定します。

名前 – unlimitedServiceDelay

型 – REG_SZ (文字列値を作成すると自動設定される)

データ – 値を設定しない

4. レジストリ・エントリに文字列値を追加し、このエントリのフィールドを次のように設定します。

名前 – serviceDelay

型 – REG_SZ (文字列値を作成すると自動設定される)

データ – サービス遅延の秒数。前もって unlimitedServiceDelay を定義していなければ (手順 3 を参照)、この値は 1 ~ 20 秒の範囲に制限されます。

lmtools ユーティリティを使用したライセンス・サーバの手動開始

lmtools はライセンス・サーバ・マネージャのグラフィカル・ユーザ・インタフェースです。lmtools には次のような機能を実行します。

- ライセンス・サーバの起動、停止、設定
- hostid などのシステム情報の取得
- サーバ・ステータスの取得

lmtools ユーザ・インタフェースから lmgrd の動作を制御するには、それを最初にライセンス・サーバ・マネージャ・サービスとして設定する必要があります。「[ライセンス・サーバ・マネージャを Windows サービスとして設定する](#)」の手順を実行してから、次の手順に進みます。

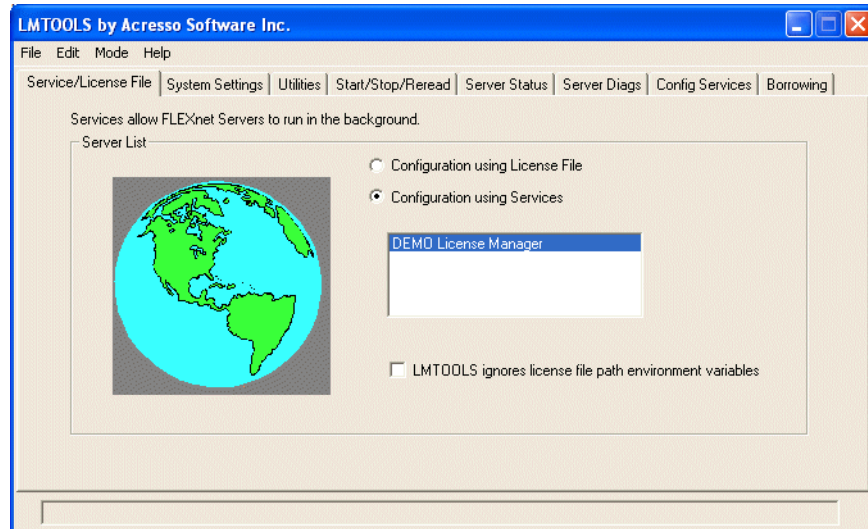
ライセンス・サーバ・マネージャ・サービスの設定後、lmtools インタフェースからサービスを開始して lmgrd を開始します。

lmtools インタフェースからサービスを開始するには、次の手順に従います。

1. lmtools を開始し、**[Service/License File]** タブを表示します。
2. **[Configuration using Services]** ボタンをクリックします。

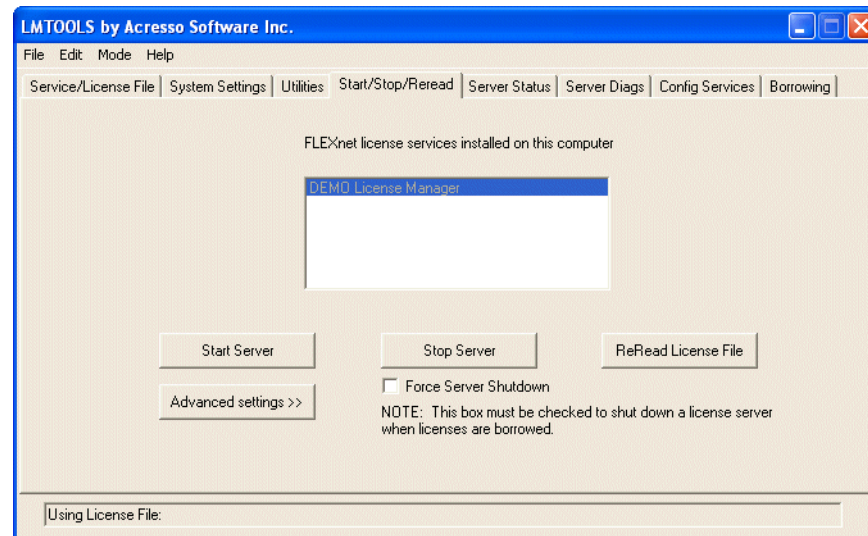
3. 選択ダイアログ・ボックス内のリストからサービス名を選択します。この例では、サービス名は「**DEMO License Manager**」です。

図 3-2: [Service/License File] タブ



4. [Start/Stop/Reread] タブをクリックします。
5. [Start Server] ボタンをクリックして DEMO License Manager を開始します。DEMO License Manager ライセンス・サーバが開始し、デバッグ・ログ出力を c:\%prods%i86_n3¥debuglog に書き込みます。

図 3-3: [Start/Stop/Reread] タブ



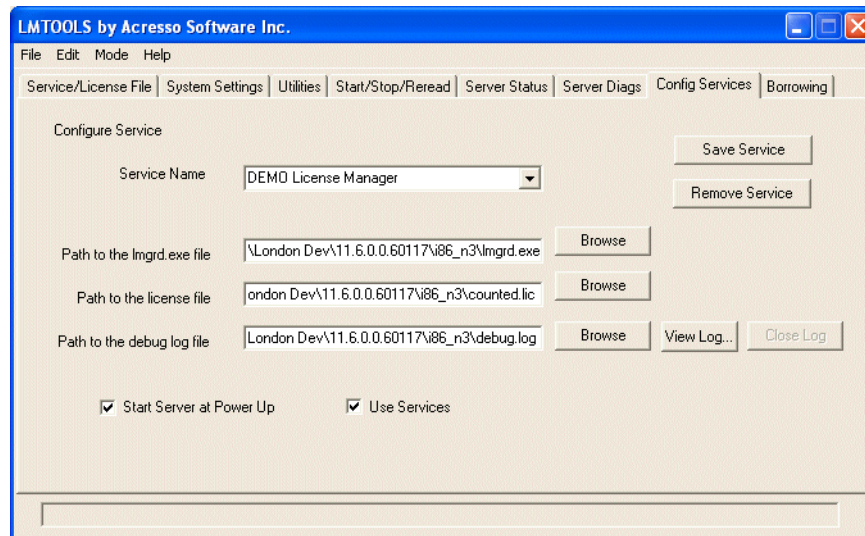
システムの起動時にライセンス・サーバが自動的に開始します

システム起動時に lmgrd が自動的に開始するためには、それを最初にライセンス・サーバ・マネージャ・サービスとして設定する必要があります。続ける前に、「[ライセンス・サーバ・マネージャを Windows サービスとして設定する](#)」の説明に従ってから、以下の手順に進んでください。

lmgrd をサービスとして設定するには、次の手順に従います。

1. lmttools を開始し、サービス名を選択し、**[Config Services]** タブをクリックします。

図 3-4: [Config Services] タブ



2. **[Use Services]** チェック・ボックスをオンにして、このライセンス・サーバ・マネージャを Windows のサービスにします。
3. **[Start Server at Power Up]** チェック・ボックスをオンにして、サーバがシステム起動時に開始するように設定します。

以後、マシンを再起動すると、このライセンス・サーバ・マネージャは Windows サービスとして自動的に開始します。

第 3 章：Imgrd – ライセンス・サーバ・マネージャ
Windows におけるライセンス・サーバ・マネージャの起動

4

ライセンス管理ツールの使用法

ライセンス管理者のライセンス管理とライセンス・サーバの管理に役立つライセンス管理ツールは、Acesso のダウンロード・サイトから入手できます。常に最新バージョンのユーティリティを使用するように努めてください。[ライセンス管理ユーティリティ](#) にユーティリティのリストを示します。

コマンド・ライン・ユーティリティ

ライセンス・サーバ・ユーティリティはすべて、`lmutil` という 1 つの実行ファイルにパッケージ化されています。`lmutil` は、個別のコマンドとしてインストールされるか (個々のコマンド名へのリンクを作成するか、`lmutil` を個々のコマンド名としてコピー)、個々のコマンドを `lmutil command` として実行するラッパーとしてインストールされます。たとえば、`lmutil lmstat` または `lmutil lmdown`。

Windows システムでは、`lmutil command` 形式のコマンドを使用できます。また、これらのコマンドに対応するグラフィカル・ユーザ・インタフェースもあります。詳細については、「[lmtools \(Windows のみ\)](#)」を参照してください。

表 4-1: ライセンス管理ユーティリティ

ユーティリティ	説明
<code>lmborrow</code>	ライセンスの借用をサポートする。
<code>lmdiag</code>	ライセンスのチェックアウトに関する問題を診断する。
<code>lmdown</code>	ライセンス・サーバ上 (3 サーバ冗長化として構成されたライセンス・サーバの場合は 3 台のマシンすべて) の、選択したベンダ・デーモン (<code>lmgrd</code> およびすべてのベンダ・デーモン) を正常に停止する。
<code>lmhostid</code>	システムの <code>hostid</code> をレポートする。
<code>lminstall</code>	ライセンス・ファイルのフォーマットを変換する。

表 4-1: ライセンス管理ユーティリティ (続き)

ユーティリティ	説明
lmnewlog	既存のレポート・ログ情報を別の名前のファイルに移動し、既存のファイル名で新しいレポート・ログ・ファイルを開始する。
lmpath	ライセンス・ファイルのパス設定をユーザが直接制御できるようにする。
lmremove	ハングしたライセンスを解放して、フリー・ライセンスのプールに戻す。
lmreread	ライセンス・デーモンにライセンス・ファイルを再読み込みさせ、新しいベンダ・デーモンをすべて起動する。
lmstat	ライセンス・サーバのステータスを表示する。
lmswitch	デバッグ・ログの場所とサイズを制御する。
lmswitchr	レポート・ログを新しいファイル名に切り替える。
lmver	ライブラリまたはバイナリ・ファイルのバージョンを報告する。

- `lmpath` ユーティリティは、バージョン 7.0 のユーティリティで導入されました。
- `lmborrow` ユーティリティは、バージョン 8.0 のユーティリティで導入されました。
- `lmswitch` ユーティリティは、バージョン 8.0 のベンダ・デーモンで導入されました。
- `lmswitchr` ユーティリティは、バージョン 5.0 のベンダ・デーモンで導入されました。

lmutil の一般的な引数

以下は、ほとんどの `lmutil` ユーティリティで有効な引数です。

表 4-2: `lmutil` の有効な引数

引数	説明
<code>-c license_file_path</code>	ほとんどの <code>lmutil</code> ユーティリティでは、ライセンス・ファイルのパスが必要です。これは、 <code>-c license_file_path</code> 引数で指定するか、または <code>LM_LICENSE_FILE</code> 環境変数を設定して指定します。指定しない場合、デフォルトの場所が使用されます。また、 <code>VENDOR_LICENSE_FILE</code> 環境変数の設定も各ユーティリティに適用されます。ユーティリティによっては、ライセンスの検索パスに複数のライセンス・ファイルのパスを指定できます。この場合、UNIX ではコロン、Windows ではセミコロンで区切ります。 パス名にスペースが含まれる場合は、二重引用符で囲む必要があります。
<code>-help</code>	使用方法を表示して終了します。
<code>-v</code>	ユーティリティのバージョンを表示して終了します。
<code>-verbose</code>	発見されたすべてのエラーについて、詳細な説明を表示します。

- ユーティリティで適用される `VENDOR_LICENSE_FILE` 環境変数は、バージョン 7.0 のユーティリティから導入されました。
- `-verbose` オプションは、バージョン 6.0 のユーティリティで導入されました。

lmborrow

`lmborrow` は、`BORROW` 属性が含まれたライセンスの借用をサポートします。このユーティリティは、ライセンスが借用されたシステムで実行する必要があります。このユーティリティを使用して、以下を実行できます。

- 借用期間の設定による借用の開始
- 借用期間のクリア
- 借用ステータスの確認
- 借用ライセンスの早期返却

借用の開始

借用を開始するには、コマンド・ラインまたは `lmtools` から `lmborrow` を実行して借用期間を設定します。

```
lmborrow {vendor | all} enddate [time]
```

各パラメータの意味は、次のとおりです。

表 4-3: 借用を開始するための `lmborrow` 引数

引数	説明
<code>vendor</code>	借用するライセンスを提供するベンダ・デーモン名。そのライセンス・サーバ内のすべてのベンダ・デーモンの場合は <code>all</code> を指定します。
<code>enddate [time]</code>	ライセンスを返却する <code>dd-mmm-yyyy</code> 形式の日付。time は省略可能で、FLEXenabled アプリケーションのローカル時間を 24 時間形式 (<code>hh:mm</code>) で指定します。time が指定されていない場合、チェックアウトは所定の終了日まで継続します。

次に例を示します。

```
lmborrow sampled 20-aug-2007 13:00
```

これを使用すると、レジストリ (Windows の場合 `h`) または `$HOME/.flexlmborrow` (UNIX の場合) に、`LM_BORROW` と借用期間が設定されます。

対象となるベンダについてライセンスを借用するには、ユーザが `lmborrow` を実行する当日に同じマシンでアプリケーションを実行して、ライセンスをチェックアウトします。その日にアプリケーションを複数回実行した場合、ライセンスが重複して借用されることはありません。借用の開始日とは別の日にアプリケーションが実行された場合、ライセンスは借用されません。

借用を開始する方法には、`lmborrow` ユーティリティ以外にも以下の方法があります。

- アプリケーションに借用インタフェースが用意されている場合は、そのインタフェースを使用する。

- LM_BORROW 環境変数を直接設定する。

これらの方法の詳細については、「[ライセンス借用の開始](#)」を参照してください。

借用ライセンスの設定のクリア



Task レジストリまたは `$HOME/.flexlmborrow` の LM_BORROW の設定をクリアするには、次の手順に従います。

- `lmborrow -clear` コマンドを発行します。

LM_BORROW の設定をクリアすると、ライセンスの借用が停止し、借用が再度開始されるまでは停止したままになります。ユーザは、オフラインで使用される機能のライセンスを借用した後で、ネットワークから切断する前に `vendor name` が提供する借用目的ではない追加機能をチェックアウトするアプリケーションを実行したい場合などに、`lmborrow -clear` を実行します。LM_BORROW をクリアしても、既に借用したライセンスのステータスは変更されません。

借用ライセンスのステータスの確認



Task 借用した機能に関する情報を出力するには、次の手順に従います。

- 借用が行われたマシンで次のコマンドを発行します。

```
lmborrow -status
```

ステータスを調べるときに、借用先システムがネットワークに接続している必要はありません。

借用ライセンスの早期返却



Task 借用したライセンスを早期返却するには、次の手順に従います。

1. 借用先システムをネットワークに再接続します。
2. 借用を開始したのと同じマシンから次のコマンドを発行します。

```
lmborrow -return [-fqdn] [-c license_file_list] [-c display] feature
```

各パラメータの意味は、次のとおりです。

表 4-4: 借用ライセンス早期返却のための lmborrow 引数

引数	説明
<code>-fqdn</code>	完全修飾ホスト名を使用して借用先システムにアクセスするように lmborrow に指示します。ライセンスの借用時に、相対識別名ではなく完全修飾ホスト名を使用する場合は、このオプションを使用します。ライセンスを借用時に使用したホスト名の形式を確認するには、lmmstat を使用します。
<code>-c license_file_list</code>	指定したライセンス・ファイルを使用します。設定によっては、ライセンス・ファイルを早期返却する場合に、ライセンス・ファイルの指定が必要になります。
<code>-d display</code>	借用が開始されたディスプレイを指定します。現在のディスプレイが、借用の開始時に使用されたディスプレイと異なる場合に必要です。 Windows では、システム名か、(ターミナル・サーバ環境の場合は) ターミナル・サーバ・クライアント名です。UNIX では、/dev/ttyxx 形式の名前か、または X-Display 名です。
<code>feature</code>	早期返却する、借用した機能の名前です。lmborrow -status を使用して、借用した機能名のリストを取得します。

ライセンスを早期返却すると、返却したライセンスを提供しているベンダ・デーモンの LM_BORROW 設定がクリアされます。

借用システムがネットワークに再接続されていない状態で早期返却を試みると、ライセンスは返却されず、LM_BORROW も変更されません。さらに、ユーザにエラー・メッセージが発行され、システムをネットワークに接続する必要があることが通知されます。

- 借用ライセンスの早期返却はバージョン 8.3 のユーティリティで導入されました。

lmdiag

lmdiag を使用すると、ライセンスをチェックアウトできない場合に問題を診断できます。

使用法

```
lmdiag [-c license_file_list] [-n] [feature[:keyword=value]]
```

各パラメータの意味は、次のとおりです。

表 4-5: lmdiag 引数の使用法

引数	説明
<code>-c license_file_list</code>	指定したファイルを診断します。
<code>-n</code>	非対話型モードで実行します。このモードでは、lmdiag の入力プロンプトは表示されません。このモードでは、詳細接続診断は使用できません。
<code>feature</code>	この機能のみを診断します。

表 4-5: lmdiag 引数の使用法

引数	説明
<code>keyword=value</code>	指定する機能に対応する行がライセンス・ファイルに複数ある場合に、lmdiag のレポート対象となる特定の行を指定します。次に例を示します。 <code>lmdiag f1:HOSTID=12345678</code> <code>hostid</code> が “12345678” の行にチェックアウトを試みます。keyword は次のいずれかです。VERSION、HOSTID、EXPDATE、KEY、VENDOR_STRING、ISSUER

feature を指定しない場合、lmdiag はリスト内のライセンス・ファイルのすべての機能を対象とします。lmdiag は最初にライセンスに関する情報を出力してから、各ライセンスのチェックアウトを実行します。チェックアウトが成功すると、lmdiag では成功したことが通知されます。チェックアウトが失敗すると、lmdiag によって失敗の理由が通知されます。lmdiag がライセンス・サーバに接続できないという理由でチェックアウトに失敗した場合は、詳細接続診断を実行するオプションがあります。

これらの詳細診断では、ライセンス・サーバの各 TCP/IP ポートに接続し、ライセンス・ファイルのポート番号の誤りを検出します。lmdiag は、受信を待機している各 TCP/IP ポート番号を示し、lmgd または lmadmin プロセスの場合、lmdiag はそれについても示します。lmdiag は、テストする機能のベンダ・デーモンを検出した場合は、問題を修正するためにライセンス・ファイルの正しいポート番号を通知します。

参照

[FLEXLM_DIAGNOSTICS](#)

lmdown

lmdown ユーティリティを使用すると、すべてのマシンから選択したライセンス・デーモン (lmgd および選択したベンダ・デーモン) を正常に停止できます。

使用法

```
lmdown -c license_file_list [-vendor vendor_daemon] [-q] [-all] [-force]
```

各パラメータの意味は、次のとおりです。

表 4-6: lmdown 引数の使用法

引数	説明
<code>-c license_file_list</code>	指定したライセンス・ファイルを使用します。lmdown では、常に <code>-c license_file_list</code> を指定することをおすすめします。
<code>-vendor vendor_daemon</code>	このベンダ・デーモンのみを停止します。lmgd は実行を継続します。バージョン 6.0 の lmdown と lmgd が必要です。
<code>-q</code>	プロンプトやヘッダを出力しません。これを指定しないと、lmdown から “Are you sure? [y/n]:” という質問が表示されます。

表 4-6: lmdown 引数の使用法

引数	説明
-all	複数のサーバを指定した場合に、それらすべてを自動的に停止します。 -all を指定した場合、 -q も暗黙的に指定したものとみなされます。
-force	ライセンスを借用している場合、 lmdown は、ライセンス・サーバが稼動しているシステムからのみ実行し、さらにユーザが -force を追加した場合にのみ実行します。

lmdown が複数のサーバを検出し (たとえば **-c** で指定したディレクトリに多数の *.lic ファイルがある場合)、**-all** を指定していない場合は、停止するライセンス・サーバの選択肢が表示されます。



注意 : UNIX では、ライセンス・サーバ・システムを停止するために **kill -9** を使用しないでください。Windows では、タスク・マネージャを使用して **FLEXnet** ライセンス・サービスを強制終了する場合には、必ず最初に **lmgrd** プロセスを終了してから、ベンダ・デーモン・プロセスをすべて終了してください。

lmdown ユーティリティを使用して、3 サーバ冗長化として構成されたライセンス・サーバを停止する場合、サーバが停止するまでに 1 分の遅延があります。**lmdown** ユーティリティは 3 台のライセンス・サーバをすべて停止します。冗長ライセンス・サーバのうち 1 台だけを停止すると、2 つの障害ポイントが残るのでおすすめてできませんが、やむを得ない場合は、そのライセンス・サーバの **lmgrd** とベンダ・デーモンの両方のプロセスを停止する必要があります。

ライセンス・サーバ・マネージャ **lmadmin** または **lmgrd** の起動時に、**lmdown** が不正に実行されるのを防止できます。サーバを停止すると、ユーザがライセンスを失います。

参照

[lmadmin のライセンス管理機能](#)

lmdown の使用の保護の詳細については、「[lmgrd のコマンド・ライン構文](#)」を参照してください。

[lmrread](#)

lmhostid

lmhostid ユーティリティは、現在のプラットフォームの **hostid** を返します。引数を指定せずに **lmhostid** を呼び出すと、現在のプラットフォームのデフォルトの種類 **hostid** が表示されます。それ以外の場合、指定した **type** に対応する **hostid** が表示されます (現在のプラットフォームでサポートされている場合)。

使用法

lmhostid [-n] [-type] [-utf8]

各パラメータの意味は、次のとおりです。

表 4-7: lmhostid 引数の使用法

引数	説明
-n	hostid のみを単独の文字列として返します。ライセンス・ファイルの HOSTID= での使用に適しています。ヘッダ・テキストは出力されません。
-type	次のいずれかの hostid の種類です。指定しなかった場合、現在のプラットフォームのデフォルトの hostid が表示されます。デフォルトの種類のリストについては、「 サポートされるプラットフォームの hostid 」を参照してください。 PLATFORM-DEPENDENT HOSTIDS <ul style="list-style-type: none">• -ether - イーサネット・アドレス• -string - 文字列 id• -vsn - ボリュームのシリアル番号 (Windows プラットフォームのみ)。• -flexid - パラレルまたは USB FLEXnet ID ドングルの ID。これは、FLEXnet ID ドングルをサポートしているプラットフォームにのみ該当します。完全なリストについては、「マシンの hostid の取得」を参照してください。• -long - 32 ビット hostid 各プラットフォームに共通の hostid <ul style="list-style-type: none">• -user - 現在のユーザ名• -display - 現在のディスプレイ名。Windows では、システム名か、(ターミナル・サーバ環境の場合は)ターミナル・サーバ・クライアント名です。UNIX では、/dev/ttyxx 形式の名前か、または X-Display 名です。• -hostname - 現在のホスト名• -internet - 現在のプラットフォームの IP アドレス (###.###.###.### 形式)。
-utf8	hostid は、ASCII 文字列ではなく UTF-8 でコード化された文字列として出力されます。hostid に ASCII の A ~ Z, a ~ z, または 0 ~ 9 以外の文字が含まれる場合は、このオプションを使用して lmhostid を指定します。hostid を正しく表示するには、UTF-8 エンコード文字列を表示できるメモ帳などのユーティリティを使用します。

次に、このコマンドの出力例を示します。

```
lmhostid - Copyright (c) 1989-2008 Acrecco Software Inc. All Rights Reserved.
```

```
The FLEXnet host ID of this machine is ""00ff5018c189 0019d244e9fc 0016cfdaf65d 001558809422  
005056c00001 005056c00008""
```

```
Only use ONE from the list of hostids.
```

参照

[サポートされるプラットフォームの hostid](#)

lminstall

lminstall ユーティリティは主に、10 進形式のライセンスを入力して、判読可能な形式のライセンス・ファイルを生成するためのものです。

使用法

```
lminstall [-i in_lic_file ] [-maxlen n] [-e err_file] [-o out_lic_file] ¥
          [-overfmt {2 | 3 | 4 | 5 | 5.1 | 6 | 7 | 7.1 | 8}] [-odecimal]
```

通常、10 進形式から判読可能な形式に変換するには、引数なしで lminstall を使用します。その場合、出力ライセンス・ファイルの名の入力プロンプトが表示されます。デフォルトのファイル名は、今日の日付を `yyyymmdd.lic` の形式で表したものです。アプリケーションのデフォルトのライセンス・ファイル・ディレクトリをソフトウェア・パブリッシャが指定している場合は、このファイルをそのディレクトリに移動します。それ以外の場合は、`LM_LICENSE_FILE` または `VENDOR_LICENSE_FILE` 環境変数を使用して、*.lic ファイルのディレクトリを指定します。

入力を完了するには、行に **q** とだけ入力するか、空白を 2 行入力します。

入力ファイルのみを指定して出力ファイルを指定しなかった場合は、`stdout` に出力されます。入力ファイルと出力ファイルのいずれも指定しなかった場合、lminstall は、入力は `stdin` であるとみなし、出力ファイルの名前の指定を求めるプロンプトをユーザに表示します。

lminstall は、判読可能な形式から 10 進形式へのライセンスの変換や、バージョンの異なるライセンス間での形式の変換にも使用されます。



Task 判読可能な形式から 10 進形式に変換するには、次のように指定します。

```
lminstall -i in_lic_file -o out_lic_file -odecimal
```



Task v5.1 形式に変換するには、次のように指定します。

```
lminstall -i in_lic_file -o out_lic_file -overfmt 5.1
```



Task 行の最大長をたとえば 50 文字に制限するには、次のように指定します。

```
lminstall -maxlen 50
```

変換エラーは必要に応じて報告され、`-e err_file` と指定してファイルに書き込むことができます。lminstall には最大 1,000 行という入力制限があります。

lmnewlog

lmnewlog ユーティリティは、既存のレポート・ログ情報を新しいファイルに移動してから、最初のレポート・ログ・ファイル名で新しいレポート・ログ・ファイルを開始することによって、レポート・ログ・ファイルを切り替えます。lmswitchr ではなく lmnewlog を使用してレポート・ログを切り替える場合は、ベンダ・デーモンのオプション・ファイルの REPORTLOG 行でファイル名を変更する必要があります。バージョン 7.1 以降のベンダ・デーモンが必要です。

使用法

```
lmnewlog [-c license_file_list] feature renamed_report_log
```

または

```
lmnewlog [-c license_file_list] vendor renamed_report_log
```

各パラメータの意味は、次のとおりです。

表 4-8: lmnewlog 引数の使用法

引数	説明
-c license_file_list	指定したライセンス・ファイルを使用します。
feature	このライセンス・ファイルの任意の機能です。
vendor	このライセンス・ファイルのベンダ・デーモンの名前です。
renamed_report_log	既存のレポート・ログ情報の移動先となる新しいファイル・パスです。

Impath

Impath ユーティリティを使用すると、ライセンス・パスの設定を直接制御できます。現在のライセンス・パスの設定の追加、上書き、取得に使用します。

使用法

```
Impath {-add | -override} {vendor | all} license_file_list
```

各パラメータの意味は、次のとおりです。

表 4-9: Impath 引数の使用法

引数	説明
-add	license_file_list を現在のライセンスの検索パスに付加するか、ない場合はライセンスの検索パスを作成して、license_file_list に初期化します。重複は破棄されます。

表 4-9: `lmpath` 引数の使用法

引数	説明
<code>-override</code>	<p>既存のライセンス検索パスを <code>license_file_list</code> で上書きします。 <code>license_file_list</code> が NULL 文字列 "" の場合には、指定したリストが削除されます。</p> <ul style="list-style-type: none"> • <code>lmpath -override all ""</code> - <code>LM_LICENSE_FILE</code> の値を削除します。 • <code>lmpath -override vendor ""</code> - <code>VENDOR_LICENSE_FILE</code> の値を削除します。
<code>vendor</code>	ベンダ・デーモンの名前です。 <code>VENDOR_LICENSE_FILE</code> の値に反映されます。
<code>all</code>	すべてのベンダ・デーモンを参照します。 <code>LM_LICENSE_FILE</code> の値に反映されます。
<code>license_file_list</code>	UNIX では、値をコロンで区切ります。Windows では、値をセミコロンで区切ります。 <code>license_file_list</code> が NULL 文字列 "" の場合には、指定したエントリが削除されます。



注意: `lmpath` は、Windows ではレジストリのエントリ、UNIX では `$HOME/.flexlmcrc` を設定すると機能します。



Task 現在のライセンス・パス設定を表示するには

```
lmpath -status
```

次のように表示されます。

```
lmpath - Copyright (C) 1989-2008 Acreesso Software Inc.  

Known Vendors:
```

```
demo:  ./counted.lic:./uncounted.lic
```

```
Other Vendors:
```

```
/usr/local/flexlm/licenses/license.lic
```

パスをディレクトリに設定した場合は、すべての *.lic ファイルが別々に表示されます。

lmremove

lmremove ユーティリティを使用すると、指定した機能の単一ユーザのライセンスを削除できます。アプリケーションがアクティブな場合は、lmremove による解放直後にライセンスを再度チェックアウトします。

使用法

```
lmremove [-c license_file_list] feature user user_host display
```

または

```
lmremove [-c license_file_list] -h feature server_host port handle
```

各パラメータの意味は、次のとおりです。

表 4-10: lmremove 引数の使用法

引数	説明
<code>-c license_file_list</code>	ライセンス・ファイルを指定します。
<code>feature</code>	ユーザがチェックアウトした機能の名前です。
<code>user</code>	ライセンスの削除対象となるユーザの名前です。lmstat -a で報告される名前です。
<code>user_host</code>	ユーザがログインしているホストの名前です。lmstat -a によってレポートされるものと同じです。
<code>display</code>	ユーザが作業しているディスプレイの名前です。lmstat -a で報告される名前です。
<code>server_host</code>	ライセンス・サーバが実行しているホストの名前です。
<code>port</code>	ライセンス・サーバが実行している TCP/IP ポート番号です。lmstat -a で報告されるポート番号です。
<code>handle</code>	lmstat -a で報告されるライセンス・ハンドルです。

`user`、`user_host`、`display`、`server_host`、`port`、および `handle` の情報は、lmstat -a の出力から取得する必要があります。

lmremove は `user_host` と `display` にある `user` のすべてのインスタンスを機能の使用状況から削除します。オプションの `-c license_file_list` を指定した場合は、そのファイルがライセンス・ファイルとして使用されます。

`-h` バリエーションは、lmstat -a で報告された `server_host`、`port`、およびライセンス・`handle` を使用します。次に lmstat -a の出力例を示します。

```
joe nirvana /dev/tty5 (v1.000) (cloud9/7654 102), start Fri 10/29 18:40
```

この例の場合、ユーザは **joe**、ユーザ・ホストは **nirvana**、ディスプレイは **/dev/tty5**、サーバ・ホストは **cloud9**、TCP/IP ポートは **7654**、ライセンス・ハンドルは **102** です。



Task このライセンスを削除するには、次のいずれかのコマンドを発行します。

```
1mremove f1 joe nirvana /dev/tty5
```

または

```
1mremove -h f1 cloud9 7654 102
```

ハンドルを指定して削除する場合、ライセンスが重複してグループ化されているときには、重複するライセンスも削除されます。ライセンスの残留を設定している場合に、`1mremove` を使用してライセンスを再利用すると、`1mremove` が開始しますが、ライセンスの残留期間は上書きされません。

ユーザのライセンスが削除されると影響が大きいため、ライセンス・サーバ・マネージャ `1madmin` または `1mgrd` の起動時に、`1mremove` が不正に実行されるのを防止できます。

参照

[1madmin のライセンス管理機能](#)

`1mremove` へのアクセス保護の詳細については、「[1mgrd のコマンド・ライン構文](#)」を参照してください。

1mreread

`1mreread` ユーティリティは、ライセンス・サーバ・マネージャにライセンス・ファイルを再読み込みさせ、新しく追加されたベンダ・デーモンを起動します。また、現在実行中のすべてのベンダ・デーモンは、ライセンス・ファイルとエンドユーザのオプション・ファイルを再読み込みして変更をチェックするように指示されます。レポート・ログが有効になっている場合は、ベンダ・デーモンの内部データ・バッファに残っているレポート・ログ・データがすべてフラッシュされます。`1mreread` はシステムのホスト名の変更を認識しますが、サーバの TCP/IP ポート番号の変更には使用できません。

オプションのベンダ・デーモン名を指定した場合は、指定されたデーモンのみがライセンス・ファイルとそのオプション・ファイルを再読み込みします（この場合、`1madmin` または `1mgrd` はライセンス・ファイルを再読み込みしません）。

使用法

```
1mreread [-c license_file_list] [-vendor vendor] [-all]
```

各パラメータの意味は、次のとおりです。

表 4-11: 1mreread 引数の使用法

引数	説明
<code>-c <i>license_file_list</i></code>	指定したライセンス・ファイルを使用します。
<code>-vendor <i>vendor</i></code>	<i>vendor</i> オプションで指定したベンダ・デーモンのみがライセンス・ファイルとオプション・ファイルを再読み込みします。さらに、 <code>1mgrd</code> は必要に応じて <i>vendor</i> を再起動します。
<code>-all</code>	複数の <code>1mgrd</code> を指定した場合に、すべての <code>1mgrds</code> に再読み込みを指示します。



注意：-c license_file_list オプションを指定した場合、指定したライセンス・ファイルは、lmgrd ではなく lmreread によって読み込まれます。lmgrd は、最初に読み込んだファイルを再読み込みします。

ライセンス・サーバ・マネージャ lmgrd の起動時に、lmreread が不正に実行されるのを防止できます。

参照

lmadmin で使用可能な GUI ベースの機能については、「[lmadmin のライセンス管理機能](#)」を参照してください。

lmreread へのアクセス保護の詳細については、「[lmgrd のコマンド・ライン構文](#)」を参照してください。

オプション・ファイルの再読み込みに関連するベンダ・デーモンの機能は、バージョン 8.0 のベンダ・デーモンで導入されました。

lmstat

The lmstat ユーティリティでは、以下のようなネットワーク・ライセンス活動のステータスをモニタできます。

- 動作しているデーモン
- ライセンス・ファイル
- 個々の機能のユーザ
- 個々のベンダ・デーモンが処理する機能のユーザ
- 借用している BORROW ライセンス

lmstat は、ライセンス・サーバから受信した情報を出力するので、非カウント・ライセンスなどのアンサーブド・ライセンスについてはレポートしません。非カウント・ライセンスについてレポートするには、サーブド・ライセンス・ファイルにライセンスを追加し、(*@host, port@host*、または *USE_SERVER* と入力して)アプリケーションがそのライセンス・ファイルのライセンス・サーバを使用するように指定する必要があります。lmstat では、キューイングされたユーザと重複グループ化によって共有されたライセンスも返されません。

使用法

```
lmstat [-a] [-c license_file_list] [-f [feature]] [-i [feature]] [-s[server]]  
        [-S [vendor]] [-t timeout_value]
```

各パラメータの意味は、次のとおりです。

表 4-12: lmstat 引数の使用法

引数	説明
-a	すべての情報を表示します。このオプションは、大きい負荷がかかる可能性のあるコマンドです。アクティブなユーザが多数いる場合、このコマンド・オプションによって大量のネットワーク活動が発生します。
-c license_file_list	指定したライセンス・ファイルを使用します。
-f [feature]	feature のユーザを表示します。feature を指定しなかった場合は、すべての機能の使用状況が表示されます。

表 4-12: lmstat 引数の使用法

引数	説明
-i [<i>feature</i>]	指定した <i>feature</i> について、機能定義行の情報を表示します。 <i>feature</i> を指定しなかった場合は、すべての機能について表示します。
-s [<i>server</i>]	<i>server</i> について、 <code>\$VENDOR_LICENSE_FILE</code> または <code>\$LM_LICENSE_FILE</code> に示されているすべてのライセンス・ファイルのステータスを表示します。 <i>server</i> を指定しなかった場合は、すべてのサーバについて表示します。
-S [<i>vendor</i>]	<i>vendor</i> の機能のすべてのユーザを表示します。
-t <i>timeout_value</i>	接続タイムアウトを <i>timeout_value</i> に設定します。この値は lmstat が <i>server</i> に接続しようとする時間を制限します。

次は、`lmstat -a` の出力例です。

```
lmstat - Copyright (c) 1989-2008 Acrecco Software Inc. All Rights Reserved.
Flexible License Manager status on Wed 11/28/2007 14:49
[Detecting lmgrd processes...]
License server status: 27000@prod
  License file(s) on prod: C:\%prod%i86_n3%counted.lic:

prod: license server UP v11.5
Feature usage info:
Users of f1: (Total of 4 licenses issued; Total of 1 license in use)
  "f1" v1.0, vendor: demo
  floating license
    daniel myhost2 19.36.18.26 (v1.0) (myhost1/27000 102), start Fri
      5/3 7:29
```

各パラメータの意味は、次のとおりです。

表 4-13: lmstat の出力

出力	引数	説明
<i>daniel</i>	<i>user</i>	ユーザ名。
<i>myhost2</i>	<i>user_host</i>	ユーザが実行しているホスト。
<i>19.36.18.26</i>	<i>display</i>	ユーザが実行しているディスプレイ。
<i>v1.0</i>	<i>version</i>	機能のバージョン。
<i>myhost1</i>	<i>server_host</i>	ライセンス・サーバが実行しているホスト。
<i>27000</i>	<i>port</i>	ライセンス・サーバが実行している <i>server_host</i> の TCP/IP ポート。
<i>102</i>	<i>handle</i>	ライセンス・ハンドル。
<i>start Fri 5/3 7:29</i>	<i>checkout_time</i>	このライセンスがチェックアウトされた時刻。

`lmremove` を使用してライセンスを削除するときに、*user*、*user_host*、*display*、*server_host*、*port*、および *handle* の情報が使用されます。

lmswitch

lmswitch ユーティリティは、ベンダ・デーモンの既存のデバッグ・ログを終了して、別のファイル名でそのベンダ・デーモンの新しいデバッグ・ログを開始することによって、特定のベンダ・デーモンが書き込むデバッグ・ログ・ファイルを切り替えます。また、そのベンダ・デーモンが書き込むデバッグ・ログ・ファイルがまだ存在しない場合にも、新しいデバッグ・ログ・ファイルを開始します。

使用法

```
lmswitch [-c license_file_list] vendor new_debug_log
```

各パラメータの意味は、次のとおりです。

表 4-14: lmswitch 引数の使用法

引数	説明
-c license_file_list	指定したライセンス・ファイルを使用します。
vendor	このライセンス・ファイルのベンダ・デーモンです。
new_debug_log	新しいデバッグ・ログ・ファイルのパスです。

デフォルトでは、lmgrd のデバッグ・ログ出力、およびその lmgrd によって開始されたすべてのベンダ・デーモンのデバッグ・ログ出力は、同じデバッグ・ファイルに書き込まれます。lmswitch を使用すると、ログ・ファイルをベンダ・デーモンごとに分離でき、デバッグ・ログ・ファイルのサイズを制御できます。

このベンダ・デーモンのデバッグ・ログ出力が別個のファイルが既に指定されていない場合は、lmswitch はこのベンダ・デーモンに対し、new_debug_log ファイルにデバッグ・ログ出力の書き込みを開始するよう指示します。このベンダ・デーモンが既に独自のデバッグ・ログに書き込んでいる場合は、lmswitch はこのベンダ・デーモンに対し、現在のデバッグ・ログ・ファイルを終了して new_debug_log ファイルにデバッグ・ログ出力の書き込みを開始するよう指示します。



注意：lmswitch の効果が続くのは、そのベンダ・デーモンが停止するか、またはそのオプション・ファイルが lmreread で再読み込みされるまでです。ベンダ・デーモンが再起動されるか、またはそのオプション・ファイルが再読み込みされると、ベンダ・デーモンはオプション・ファイルの DEBUGLOG 行を検索し、独自のファイルにデバッグ・ログ出力を書き込むかどうか、およびその場合はどのファイルに書き込むかを判別します。

参照：

ladmin 表示については、「[ladmin のライセンス管理機能](#)」を参照してください。

[DEBUGLOG](#)

[lmreread](#)

[デバッグ・ログ・ファイル](#)

lmswitchr

lmswitchr ユーティリティは、レポート・ログ・ファイルを切り替えます。既存のレポート・ログを閉じ、新しいファイル名で新しいレポート・ログを開始します。また、レポート・ログ・ファイルが存在しない場合には、新しいレポート・ログ・ファイルを開始します。

使用法

```
lmswitchr [-c license_file_list] feature new_report_log
```

またはバージョン 5.0 以降のベンダ・デーモン

```
lmswitchr [-c license_file_list] vendor new_report_log
```

各パラメータの意味は、次のとおりです。

表 4-15: lmswitchr 引数の使用法

引数	説明
<code>-c license_file_list</code>	指定したライセンス・ファイルを使用します。
<code>feature</code>	このライセンス・ファイルの任意の機能です。
<code>vendor</code>	このライセンス・ファイルのベンダ・デーモンです。
<code>new_report_log</code>	新しいレポート・ログ・ファイルのパスです。

ベンダ・デーモンのレポート・ログが有効になっていない場合、lmswitchr がベンダ・デーモンに対し、`new_report_log` にレポート・ログ出力の書き込みを開始するよう指示します。ベンダ・デーモンのレポート・ログが既に有効になっている場合、lmswitchr はベンダ・デーモンに対し、そのレポート・ログ・ファイルを終了して `new_report_log` に新しいレポート・ログ出力の書き込みを開始するよう指示します。



注意： lmswitchr の効果が続くのは、そのベンダ・デーモンが停止するか、またはそのオプション・ファイルが `lmreread` で再読み込みされるまでです。ベンダ・デーモンが再起動されるか、またはそのオプション・ファイルが再読み込みされると、ベンダ・デーモンは、オプション・ファイルの `REPORTLOG` 行を検索し、独自のファイルにレポート・ログ出力を書き込むかどうか、およびその場合はどのファイルに書き込むかを判別します。

参照：

[REPORTLOG](#)

[lmnewlog](#)

[lmreread](#)

[レポート・ログ・ファイル](#)

lmver

lmver ユーティリティは、FLEXnet Publisher ライセンス・ツールキット・ライブラリまたはバイナリ・ファイルのバージョンを報告します。

使用法

lmver *filename*

ここで *filename* は次のいずれかです。

- FLEXnet Publisher ライセンス・ツールキットで構築された実行可能ファイルの名前
- lmgrd
- ライセンス管理ツール
- ベンダ・デーモン

たとえば、**spell** というアプリケーションがある場合は、「**lmver spell**」と入力します。

lmtools (Windows のみ)

lmtools ユーティリティは、ライセンス・サーバを管理できるグラフィカル・ユーザ・インタフェースです。この実行ファイルは 32 ビットおよび 64 ビットの Windows パッケージで利用可能です。常に最新バージョンを使用するように努めてください。最新バージョンはソフトウェアのダウンロード・サイトから入手できます。

このユーティリティには次のような機能があります。

- ライセンス・サーバの起動、停止、設定
- *hostid* などのシステム情報の取得
- サーバ・ステータスの取得

lmtools ユーティリティには、ライセンス・サーバの設定モードが次の 2 とおりあります。

- ライセンス・ファイルを使用した設定
- サービスを使用した設定

Windows Vista では、lmtools ユーティリティを管理者として実行する必要があります。この実行ファイルを管理者として実行しない場合は、(システムで UAC プロンプトが無効になっていなければ) 起動と同時にユーザ・アカウント・コントロール (UAC) ダイアログが表示されます。

ライセンス・ファイルを使用した設定

特定のライセンス・ファイルに操作を実行します。ファイルは、ローカルとリモートのいずれかです。このモードでは、lmgrd プロセスを起動できませんが、それ以外はすべて実行できます。

このモードを設定するには、次の手順に従います。

1. lmtools ユーティリティを実行します。
2. **[Configuration using License File]** ボタンをクリックします。
3. 1 つまたは複数のライセンス・ファイル名を入力するか、`port@host` と指定します。

サービスを使用した設定

サービスに処理が実行され、lmtools を実行しているシステムでローカルの lmgrd プロセスを起動できます。サービスの設定の詳細については、「[ライセンス・サーバ・マネージャを Windows サービスとして設定する](#)」を参照してください。

IPv6 のサポート

インターネット・プロトコル・バージョン 6 (IPv6) は次世代の IP プロトコルです。この項には、IPv6 アドレスをサポートするネットワークを使用しているネットワーク管理者向けの情報が含まれています。この項の情報は、IPv6 ネットワーク・プロトコルの知識がある読者を対象としています。この章の以下の項では、FLEXnet Publisher ライセンス・ツールキットの IPv6 サポートについて説明します。

- IPv6 をサポートする機能
- 混合プロトコル環境でのライセンス・サーバの配備

IPv6 をサポートする機能

この項では、IPv6 をサポートしているライセンス管理者が設定できる FLEXnet Publisher ライセンス・ツールキットの機能について説明します。この項では、ライセンス・ファイルベースのライセンスと、信頼されたストレージベースのライセンスの両方で使用されるコンポーネントについて説明します。

IPv6 をサポートしているソフトウェア・パッケージをソフトウェア・パブリッシャから購入するとき、システム (FLEXenabled クライアントとライセンス・サーバ) の IP アドレスを集めて提供する必要があります。これらはライセンス・ファイルで使用されます。

ライセンス・ファイル

ライセンス・ファイルの **SERVER** 行で、ホストの値として IPv6 アドレスを定義できます。

オプション・ファイル

以下を使用するとき、オプション・ファイルに IPv6 アドレスを含めてホスト制限を指定できます。

- これらのキーワードの INTERNET 型 - EXCLUDE、EXCLUDEALL、EXCLUDE_BORROW、INCLUDE、INCLUDEALL、INCLUDE_BORROW、MAX、RESERVE
- これらのキーワードの HOST 型 - EXCLUDE、EXCLUDE_ENTITLEMENT、EXCLUDEALL、EXCLUDE_BORROW、INCLUDE、INCLUDE_ENTITLEMENT、INCLUDEALL、INCLUDE_BORROW、MAX、RESERVE
- HOST_GROUP キーワード (IP アドレスを使用)。

ライセンス検索パス

ライセンス・サーバの識別に 'port@host' 表記を使用するライセンス検索パスのエントリは、'host' の値として IPv6 のアドレスを指定できます。

混合プロトコル環境でのライセンス・サーバの配備

FLEXnet Publisher ライセンス・ツールキットのコンポーネントが IPv6 アドレスを使用して正常に動作するには、企業内のすべてのシステム (ネットワークのハードウェアとソフトウェアを含む) が IPv6 アドレスを使用した通信をサポートするように正しく設定する必要があります。IPv6 または IPv4/IPv6 デュアル通信をサポートする FLEXenabled アプリケーションをテストまたは配備する前に、ネットワーク上のすべてのシステムが問題なく通信できることを確認してください。

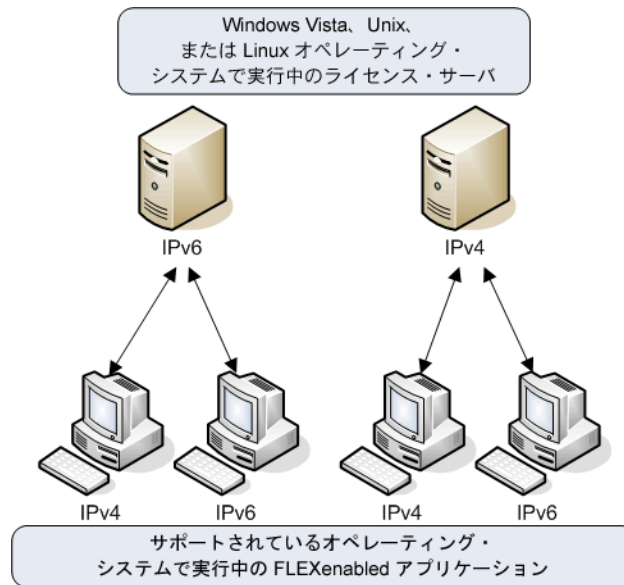
ライセンス・サーバを以下のいずれかのオペレーティング・システムで実行する場合は、(ネットワークが正しく設定されていれば) IPv4 または IPv6 を使用して FLEXenabled クライアントと通信できます。

- Windows Vista のサポートされているエディション
- サポートされている Linux プラットフォーム
- サポートされている Unix プラットフォーム

これらのオペレーティング・システムはデュアルレイヤ通信をサポートしているので、IPv4 と IPv6 の両方の FLEXenabled クライアントが IPv6 のライセンス・サーバと通信できます。また、IPv6 のクライアントは IPv4 アドレスを使用して IPv4 のライセンス・サーバと通信できます。図 5-1 は、この動作を表しています。

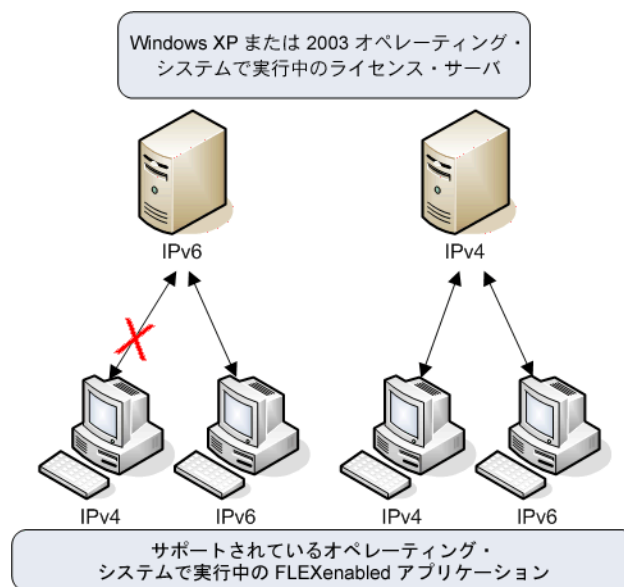
ライセンス・サーバとして lmadm を使用している場合は、IPv4 と IPv6 の両方のクライアントをサポートします。IPv4 と IPv6 に別々のベンダ・デーモンが必要になるため、ベンダ・デーモン実行ファイルの 1 つで名前を変更する必要があります。

図 5-1: Windows Vista、Unix、または Linux で実行しているライセンス・サーバ



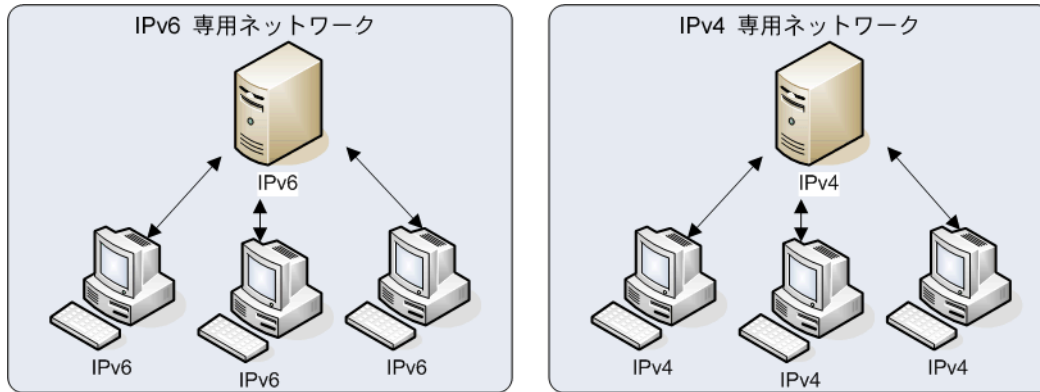
ライセンス・サーバが Windows XP または Windows Server 2003 オペレーティング・システムで実行している場合は、デュアルレイヤのサポートが限られているため、いくつかの制限があります。IPv4 FLEXenabled クライアントは、これらのオペレーティング・システムで実行している IPv6 ライセンス・サーバと通信できません。ただし、IPv6 FLEXenabled クライアントは、これらのオペレーティング・システムで実行している IPv4 サーバと通信できます。図 5-2 は、この動作を表しています。

図 5-2: Windows 2003 または XP で実行しているライセンス・サーバ



Windows 2003 または Windows XP でライセンス・サーバを実行している企業の場合、ライセンス管理者は、IPv6 ライセンス・サーバを使用する IPv6 FLEXenabled クライアント用と、IPv4 ライセンス・サーバを使用する IPv4 FLEXenabled クライアント用に 2 つのネットワークを作成して別々に管理する必要があります。次の図は、この構成を示しています。

図 5-3: IPv4 と IPv6 の別個の環境



IPv6 アドレスでのワイルドカードの使用

アドレスをすべて指定する代わりに、ワイルドカード文字“*”をフィールド全体またはバイトごとに使用してアドレス範囲を指定することもできます。たとえば、この例では機能定義の行が 4 つの特定のアドレスにロックされています。

```
FEATURE f1 myvendor 1.0 1-jan-2008 uncounted ¥
  HOSTID="INTERNET=127.17.0.1,¥
    INTERNET=2001:0db8:0000:0000:ff8f:effa:13da:0001,¥
    INTERNET=127.17.0.4,¥
    INTERNET=2001:0db8:0000:0000:ff8f:effa:13da:0004" ¥
  SIGN="<...>"
```

次の例では、機能定義の行が、その上の行の 4 つのアドレスを含むアドレス範囲全体を指定しています。

```
FEATURE f1 myvendor 1.0 1-jan-2008 uncounted ¥
  HOSTID="INTERNET=127.17.0.*,¥
    INTERNET=2001:0db8:0000:0000:*:*:*:000*"¥
  SIGN="<...>"
```

3 サーバ冗長化の使用

この項では、ライセンス・サーバを3サーバ冗長構成で設定する方法を説明します。

ライセンス管理者は、次のどちらかの方法でライセンス・サーバのフェールオーバー保護を実装できます。

- 3サーバ冗長化：3サーバ冗長化の目的でライセンス・サーバ・システム3台をセットで構成して管理します。これはフェールオーバーの保護だけを提供します。ライセンス管理者は、3台のサーバで1つのバージョンのライセンス・ファイルとベンダ・デーモンを管理します。
- ライセンス検索パスを使用した冗長化：企業で使用可能な合計ライセンスのサブセットをそれぞれに使用して、複数の独立したライセンス・サーバを設定および管理します。ライセンス・サーバがライセンス検索パスにある FLEXenabled クライアントを設定します。これは負荷分散機能と、限られたフェールオーバー保護を提供します。ライセンス管理者は、各ライセンス・サーバでさまざまなバージョンのライセンス権利を管理しなければなりません。

ソフトウェア・パブリッシャと協力して3サーバ冗長化を有効にし、フェールオーバー保護を設定してください。3サーバ冗長化は FLEXnet Publisher ライセンス・ツールキットが提供している機能の1つで、フェールオーバー保護と同時に、ライセンス権利が不正に複製されるのを防ぎます。この構成では、1台のライセンス・サーバだけが FLEXenabled アプリケーションにライセンスを供与します。

ライセンス・サーバ3台のうち少なくとも2台が実行および通信していると、システムが FLEXenabled アプリケーションにライセンスを提供します。3サーバ冗長化は、ライセンス・ファイル・ベースのライセンスでのみサポートされています。信頼できる記憶領域ベースのライセンスではサポートされていません。

3 サーバ冗長化の概要

FLEXnet Publisher ライセンス・ツールキットで3サーバ冗長化機能を使用すると、3台のライセンス・サーバが組になって運用します。少なくとも2台のサーバが実行および通信していることを確認するため、ライセンス・サーバ間で定期的にメッセージがやりとりされます。3台のうち少なくとも2台のライセンス・サーバが実行し、相互通信していると、クォーラムが形成されます。

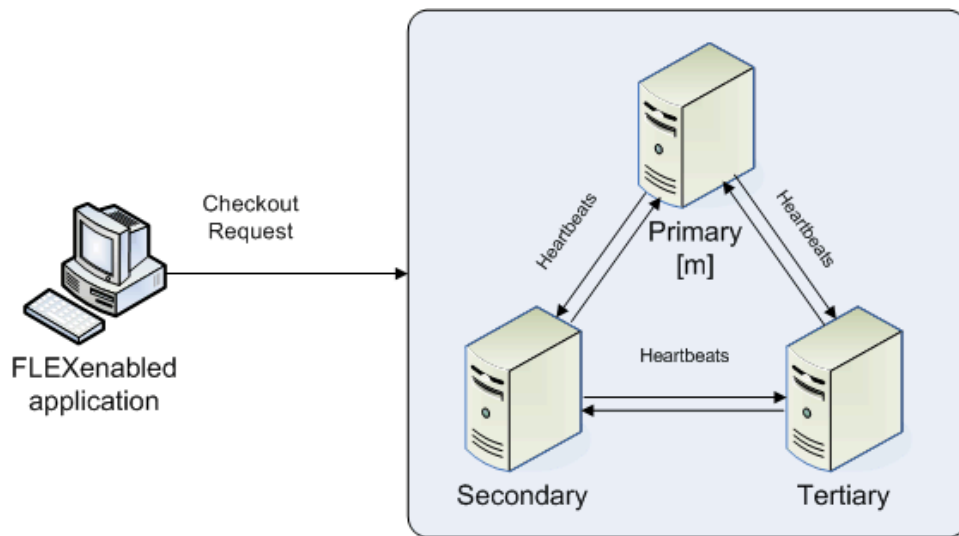
ライセンス・サーバはプライマリ、セカンダリ、ターシャリのいずれかで識別されます。1 台はマスタ [m] としても指定され、以下の処理を担当します。

- FLEXenabled アプリケーションへのライセンス供与
- デバッグ・ログへの情報の記録
- レポート・ログへの情報の記録

マスタが故障した場合は、別のライセンス・サーバがマスタになります。

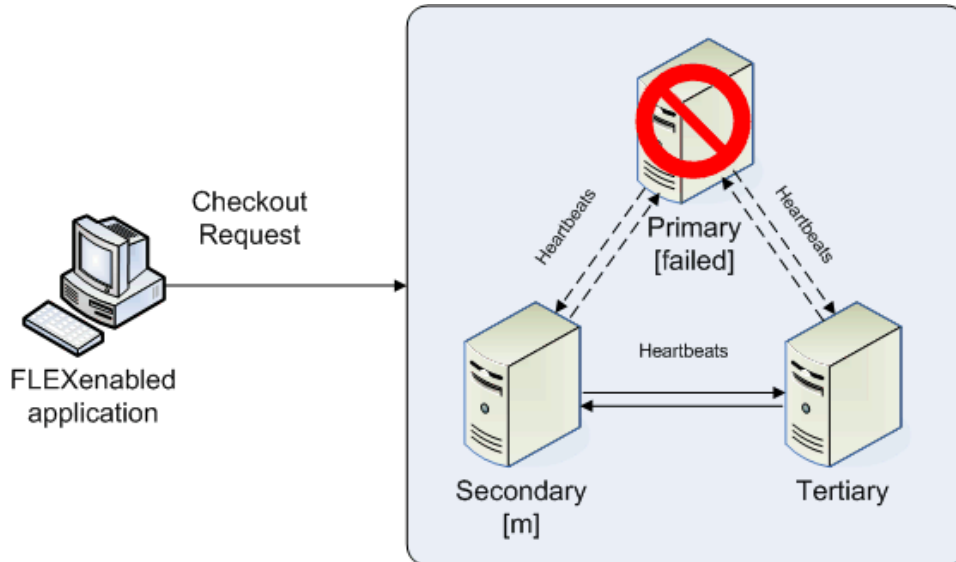
次の図では、プライマリ・ライセンス・サーバがマスタ [m] です。FLEXenabled アプリケーションがライセンスのチェックアウト要求を送信すると、マスタが応答し、FLEXenabled アプリケーションにライセンスを供与します。

図 6-1: 3 サーバ冗長化の概要



マスタが故障した場合は、セカンダリ・ライセンス・サーバがマスタになり（下図参照）、FLEXenabled アプリケーションにライセンスを供与します。ターシャリ・ライセンス・サーバはマスタにはなれません。プライマリ・サーバとセカンダリ・サーバの両方がダウンした場合、ライセンスは FLEXenabled アプリケーションに供与されなくなります。マスタは、3 台のうち少なくとも 2 台が実行および通信していなければ、ライセンスを供与しません。

図 6-2: 3 サーバ冗長化のバックアップ・フェールオーバー



ライセンス・サーバの通信の仕組み

各ライセンス・サーバは起動時にライセンス・ファイルを読み取り、他のライセンス・サーバと通信できることを確認します。各ライセンス・サーバは、他のライセンス・サーバと最初の接続を確立するまで定期的にメッセージを送信します。

最初の通信が確立すると、各ライセンス・サーバは他のライセンス・サーバに定期的にハートビートを送信します。ハートビートとは、TCP/IP を介して送信されるメッセージです。各ライセンス・サーバはハートビートを送信して、他のライセンス・サーバからの応答を待ちます。応答を受信しないと、ライセンス・サーバはベンダ・デーモンを停止して、ライセンスを供与できないようにします。パブリッシャまたはライセンス管理者は HEARTBEAT_INTERVAL プロパティを使用して、ライセンス・サーバがハートビートの受信を待つ時間を設定できます。

ネットワーク通信の質が悪いと、システム・パフォーマンスが低下します。また、低速なネットワーク通信はライセンス・サーバ間のハートビートの伝送に遅延を引き起こします。

この構成のライセンス・サーバの管理

lmstat ユーティリティの使用

lmstat ユーティリティが生成する出力メッセージによって、どのライセンス・サーバがマスタかがわかります。次の lmstat 出力例では、セカンダリ・ライセンス・サーバがマスタです。

```
[Detecting lmgrd processes...]  
License server status: 30000@RMD-PRIMARY,30000@RMD-SECONDARY,  
30000@RMD-TERTIARY  
License file(s) on RMD-PRIMARY: C:\server\3.lic:  
RMD-PRIMARY: license server UP v11.4  
RMD-SECONDARY: license server UP (MASTER) v11.4  
RMD-TERTIARY: license server UP v11.4
```

ライセンス・サーバの起動と停止

システム全体を起動するには、各ライセンス・サーバ・マネージャ (lmadmin または lmgrd) を開始する必要があります。通常は、セカンダリまたはターシャリ・ライセンス・サーバを起動する前にプライマリ・サーバを起動してください。このようにすると、他のライセンス・サーバが起動する前に、プライマリ・ライセンス・サーバがマスタになることができます。セカンダリとターシャリをプライマリの前に起動した場合は、セカンダリが自動的にマスタになります。

プライマリ・ライセンス・サーバに PRIMARY_IS_MASTER キーワードを設定しない場合は、ライセンス・サーバを起動する順序が重要になります。このプロパティを設定しない場合、セカンダリ・ライセンス・サーバの後でプライマリ・ライセンス・サーバを起動すると、制御がプライマリ・ライセンス・サーバに移りません。PRIMARY_IS_MASTER キーワードを設定しておくこと、プライマリ・ライセンス・サーバが実行しているときは、常にそれがマスタになります。

lmdown ユーティリティは 1 つのコマンドで 3 台のライセンス・サーバをすべて停止します。各ライセンス・サーバを別々に停止する必要はありません。

ライセンス・サーバ・マネージャ (lmgrd) の Windows サービスとしての実行

この構成では、ライセンス・サーバ・マネージャ (lmgrd 実行ファイル) をサービスとして実行する際の依存性や既知の問題はありません。

ロギングとデバッグ・ログ

3 サーバ冗長化を使用すると、マスタはそのローカル・デバッグ・ログとレポート・ログに (設定されている場合は Windows イベント・ログにも) 情報を記録します。このシステムが故障した場合は、別のライセンス・サーバがマスタになって、ローカルのデバッグ・ログとレポート・ログに情報を記録します。以降、プライマリ・ライセンス・サーバとセカンダリ・ライセンス・サーバには、デバッグ・ログとレポート・ログの異なるバージョンが存在し、それぞれに異なる情報が含まれている可能性があります。

ライセンス・サーバに対する 3 サーバ冗長化の設定

一部の設定手順は、ソフトウェア・パブリッシャとライセンス管理者の両方が行う必要があります。この項では、それぞれが行う手順について説明します。

ライセンス管理者用の設定

ライセンス管理者は、以下の手順に従ってください。

1. ライセンス管理者は、ライセンス・サーバのソフトウェア・パッケージを入手する前に、3 台のシステムを指定してセットアップする必要があります。システムを選択するときには、それらが安定していることを確認します。頻繁に再起動または停止するシステムは使用しないでください。
2. これらのシステムのホスト名とホスト ID をパブリッシャに送ります。ホスト ID に必要なシステム識別子については、パブリッシャに問い合わせてください。これにはイーサネット・アドレスやディスクのシリアル番号などが使用されます。パブリッシャは、これらのシステム用にライセンス・サーバのコンポーネントを作成します。
3. ライセンス・サーバ・パッケージをパブリッシャから受け取った後、必要に応じてライセンス・ファイルの SERVER 行のプロパティを次のように変更します。
 - ライセンス・サーバが通信を受信するために使用する **ポート番号**
 - **PRIMARY_IS_MASTER** キーワード
 - **HEARTBEAT_INTERVAL** プロパティ

hostid の値は変更しないでください。今後 hostid を変更した場合には、ソフトウェア・パブリッシャから新しいライセンス・ファイルを入手する必要があります。

4. ソフトウェア・パブリッシャが要求するその他の設定を行います。
5. ライセンス・サーバのソフトウェア・パッケージをシステム 3 台のそれぞれにコピーまたはインストールします。
6. ライセンス・サーバをプライマリ、セカンダリ、ターシャリの順に起動します。

ライセンス・ファイルの例

以下は、3 サーバ冗長化が設定されたライセンス・ファイルの例です。

```
SERVER pat 17003456 2837 PRIMARY_IS_MASTER
SERVER lee 17004355 2837
SERVER terry 17007ea8 2837
VENDOR demo
FEATURE f1 demo 1.0 1-jan-2018 10 SIGN="<...>"
FEATURE f2 demo 1.0 1-jan-2018 10 SIGN="<...>"
```

ライセンス・ファイルの以下の部分は、3 サーバ冗長化の構成に直接影響します。

- **SERVER** 行：この 3 行は使用される各システムを定義します。
 - **host** の値：これらは pat、lee、terry です。
 - **hostid** の値：これらは 17003456、17004355、17007ea8 です。この例では、lmhostid ユーティリティのデフォルトの hostid タイプが返す値を使用しています。デフォルトの hostid タイプはプラットフォームごとに異なります。

- **TCP/IP ports** : すべてのサーバが同じポート (たとえば 2837) を使用して通信を受信します。

ライセンス・ファイルの以下のプロパティは 3 サーバ冗長化の構成に直接影響しませんが、ライセンス権利の定義やライセンス・サーバの設定に使用されます。

- **VENDOR** 行 : これは必須で、パブリッシャのベンダ・デーモンを参照します。
- **FEATURE** 行 : f1 と f2 の 2 つの機能がライセンス権利を定義します。各 **FEATURE** 行の **SIGN** の値は、ライセンス・サーバのホスト ID の値を符号化します。

3 サーバ冗長化による他の機能の使用

次の項では、FLEXnet Publisher ライセンス・ツールキットのその他の機能、および 3 サーバ冗長化との相互作用について説明します。

ライセンス検索パスの設定

この設定は、ソフトウェア・パブリッシャまたはライセンス管理者が実行できます。FLEXenabled アプリケーションはライセンスをチェックアウトする前に、ライセンス権利の場所を知る必要があります。ライセンス検索パスはライセンス権利の場所を特定します。

3 サーバ冗長化が設定されているライセンス・サーバに接続するとき、FLEXenabled アプリケーションはライセンス検索パスで (ライセンス・ファイルの場所ではなく) port@host の表記を使用する必要があります。

ライセンス検索パスでは、ライセンス・サーバをライセンス・ファイルでの表示と同じ順序で列挙してください。このようにすると、マスタ・サーバを識別してチェックアウト要求に応答する時間を短縮できます。設定は、ライセンス検索パスにライセンス・サーバを 1 つだけ含めれば機能しますが、ライセンス・サーバがチェックアウト要求に応答する時間が長引く可能性があります。これは、ライセンス・サーバが他のライセンス・サーバをすべて識別してマスタを指定しなければならないからです。

また、port@host の各エントリは、セミコロン (Windows) やコロン (Unix/Mac) やアンパサンド (Java) ではなく、カンマで区切る必要があります。カンマは、ライセンス・サーバに 3 サーバ冗長化が設定されていることを示します。

前述のライセンス・ファイルを例にとると、ライセンス検索パスは次のようになります。

```
2837@pat,2837@lee,2837@terry
```

FLEXenabled アプリケーションは、ライセンス・サーバと接続を確立するか、リストの最後に達するまで、左から順に各ライセンス・サーバに接続を試みます。これによって FLEXenabled アプリケーションがクォーラムに接続できることを確認できます。

ライセンス・ファイルのキーワードの使用

SERVER 行に以下のキーワードやプロパティを使用すると、設定を変更できます。

- **Host**：これはシステムのホスト名です。パブリッシャは、ライセンス・ファイルを生成するときこの情報が必要です。この値は、ライセンス・ファイルの署名後に変更できます。
- **Port**：ライセンス・サーバが通信を受信するために使用するポート番号。単一ライセンス・サーバと異なり、各 **SERVER** 行にポート番号を含める必要があります。ここでは、システム上の他のプロセスが使用していない 1024 ~ 64000 の数字を入力できます。この値は、ライセンス・ファイルの署名後に変更できます。ladmin を使用している場合は、ライセンス・ファイルを編集する必要はありません。GUI インタフェースを使用してポート番号を設定できます。詳細については、オンライン・ヘルプを参照してください。

ライセンス・サーバを管理しやすくするため、各 **SERVER** 行に同じポート番号を定義することをおすすめします。この値は、ライセンス・ファイルの署名後に変更できます。

- **PRIMARY_IS_MASTER**：このキーワードは、プライマリ・サーバが実行し、他のどちらかのライセンス・サーバと通信している場合は、必ずマスタになるようにします。
 - これを設定すると、プライマリ・サーバがダウンしても、プライマリ・サーバが復活すると、プライマリ・サーバがマスタのままになります。
 - これを設定しないと、プライマリ・サーバがダウンした場合、セカンダリ・サーバがマスタとなり、プライマリ・サーバが復旧してもセカンダリ・サーバがマスタのままになります。セカンダリ・ライセンス・サーバが故障した場合にのみ、プライマリ・サーバは再びマスタになることができます。

このパラメータは省略可能で、ライセンス・ファイルの最初の **SERVER** 行に配置します。この値は、ライセンス・ファイルの署名後に変更できます。このキーワードを使用するには、ライセンス・サーバでバージョン 10.8 以降のベンダ・デーモンを実行している必要があります。

- **HEARTBEAT_INTERVAL=seconds**：これは、ライセンス・サーバが別のライセンス・サーバからハートビートを受信するまで待つ時間で、その後ベンダ・デーモンを停止します。この値は、実際のタイムアウト値を計算する次の等式で使用されます。

$$\text{timeout} = (3 * \text{seconds}) + (\text{seconds} - 1)$$

デフォルト値は 20 で、実際のタイムアウト値 79 秒と同じです。有効な値は 0 ~ 120 です。このパラメータは省略可能で、ライセンス・ファイルの最初の **SERVER** 行に配置します。この値は、ライセンス・ファイルの署名後に変更できます。このキーワードを使用するには、ライセンス・サーバでバージョン 10.8 以降のベンダ・デーモンを実行している必要があります。

オプション・ファイルのキーワードの使用

オプション・ファイルのキーワードは 3 サーバ冗長化には影響しません。

トラブルシューティングのヒントと制限事項

ライセンス・ファイルの内容の分離

SERVER 行の `hostid` の値は各機能定義行のシグニチャの計算に使用されるため、SERVER 行と機能定義行は、生成されたときのように一緒にしておいてください。つまり、機能定義行を別のファイルに移動した場合は、その SERVER 行と VENDOR 行も移動する必要があります。

ネットワーク・ファイル・サーバへのライセンス・ファイルの配置

ライセンス・ファイルをネットワーク・ファイル・サーバに配置しないでください。ネットワーク・ファイル・サーバに置くと、ファイル・サーバがシングル・ポイント障害の発生点になる可能性があるため、フェールオーバー保護を設定するメリットがなくなります。

大きいネットワーク・トラフィックでのライセンス・サーバの使用

過剰なトラフィックのネットワークでは、ライセンス・サーバがハートビートを逃して、ベンダ・デーモンの停止を引き起こす可能性があります。その場合、マスタがライセンスを供与しなくなります。過剰なネットワーク・トラフィックが原因であれば、`HEARTBEAT_INTERVAL` を大きい値に設定してください。ネットワーク通信が低速のときや、`FLEXenabled` クライアントがダイヤルアップ・リンクを介してネットワークに接続している場合は、企業でパフォーマンスの問題が発生する可能性があります。

複数のベンダ・デーモンの使用

ライセンス・サーバ・マネージャ (`ladmin` または `lmgrd`) は、3 サーバ冗長化が設定されていると、他のソフトウェア・パブリッシャのベンダ・デーモンを開始できません。ライセンス・サーバ・マネージャが管理できるベンダ・デーモンは 1 つだけです。システムの 1 つが複数のベンダを実行している場合、ライセンス管理者はそのシステムでライセンス・サーバの別のインスタンスを実行して、他のベンダ・デーモンをサポートする必要があります。ポート番号が競合しないように注意してください。

ライセンス・ファイルの読み込み

ライセンス・ファイルには、FLEXenabled アプリケーションのライセンスを管理するために必要な情報が記述されています。たとえば、次の情報があります。

- ライセンス・サーバの名前とホスト ID
- ベンダ名と、ベンダ・デーモン実行プログラムのパス
- 機能の情報

ライセンス・ファイルは、FLEXenabled アプリケーションまたはライセンス・サーバを実行するシステムにアクセス可能である必要があります。

ライセンス・ファイルの場所の指定

多くの場合、ソフトウェア・パブリッシャは指定の場所にライセンス・ファイルを置くことを推奨しています。すべてのマシンでライセンスを使用可能にする方法には、次のオプションがあります。

- ネットワーク内の、ライセンス・ファイルを必要とするすべてのシステムからアクセス可能なパーティションにライセンス・ファイルを置く。
- 個々のマシンのそれぞれにライセンス・ファイルをコピーする。
- 環境変数 `LM_LICENSE_FILE` または `VENDOR_LICENSE_FILE` (`VENDOR` はベンダ・デーモンの名前) を `port@host` に設定する (`host` と `port` は、ライセンス・ファイルの `SERVER` 行で指定されているホストとポート)。または、ライセンス・ファイルの `SERVER` 行で指定されている TCP/IP ポートがデフォルトのポート範囲内 (27000 ~ 27009) であるか、ポートが指定されていない (デフォルト範囲内のポートがライセンス・サーバ・マネージャによって選択される) 場合に、`@host` というショートカット形式で指定する。

3 サーバ冗長として構成されたライセンス・サーバの場合は、3 つの `port@host` をカンマで区切ったリストを指定することで、3 つのライセンス・サーバであることを示す。次に例を示します。

```
port1@host1,port2@host2,port3@host3
```

表 7-1 に、環境変数 `LM_LICENSE_FILE` と `VENDOR_LICENSE_FILE` の設定の例を示します。

表 7-1: 環境変数の指定の例

SERVER 行	LM_LICENSE_FILE または VENDOR_LICENSE_FILE 設定
SERVER myserver 17007ea8 40000	各パラメータの意味は、次のとおりです。 40000@myserver
	<ul style="list-style-type: none">• host = myserver• port = 40000
SERVER myserver 17007ea8 27001	各パラメータの意味は、次のとおりです。 @myserver
	<ul style="list-style-type: none">• host = myserver• port = 27001 (デフォルト範囲内)
SERVER myserver 17007ea8	各パラメータの意味は、次のとおりです。 @myserver
	<ul style="list-style-type: none">• host = myserver• port = 指定しない。27000 ~ 27009 の範囲内のデフォルトの TCP/IP ポート番号を使用する。

- Windows では、アプリケーションによってライセンス・ファイルを見つけることができない場合に、ライセンス・ファイルの場所、ライセンス・サーバ、またはインターネットからのライセンス・フルフィルメントをユーザに入力させるためのダイアログ・ボックスが表示される。

ライセンス使用状況がベンダ・デーモンによって常時監視されていることと、ライセンス・ファイルのデータが暗号化され、修正が不可能となっていることから、ユーザによるライセンス・ファイルの移動やコピーは自由に許可されています。

カウント・ライセンスの場合は、どのオプションを選択するかにかかわらず、FLEXenabled アプリケーションがネットワーク上でアクセス可能な場所に `lmadmin` または `lmgrd` とベンダ・デーモンをコピーしておく必要があります。

環境変数を使用したライセンス検索パスの設定

ほとんどのアプリケーションでは、ライセンス・ファイルを検索する場所が定められており、ライセンス・ファイルが自動的にインストールされます。ただし、`LM_LICENSE_FILE` 環境変数を `license_file_list` に設定することで、ライセンス・ファイルの場所を変更できます。`license_file_list` を指定した場合、次の要素で構成されます。

- ライセンス・ファイルのフル・パス
- `.lic` 拡張子を持つライセンス・ファイルが格納されているディレクトリ
- `port@host` の設定。`port` と `host` は、ライセンス・ファイルの `SERVER` 行で指定されている TCP/IP ポート番号とホスト名です。ライセンス・ファイルの `SERVER` 行で、デフォルトの TCP/IP ポートまたはデフォルトのポート範囲内のポート (27000 ~ 27009) が指定されている場合は、ショートカット形式の `@host` を使用することも可能です。
- 3 サーバ冗長化として構成されたライセンス・サーバであることを示す、カンマで区切られた3つの `port@host` 識別子のリスト。次に例を示します。

```
port1@host1,port2@host2,port3@host3
```

アプリケーションが受け入れる環境変数 (または Windows レジストリ) の名前は、`VENDOR_LICENSE_FILE` です。ここで、`VENDOR` は、`DEMO_LICENSE_FILE` などの、ベンダ・デーモンの名前です。この環境変数の有効範囲は、`VENDOR` 名を使用するソフトウェア・パブリッシャからのアプリケーションだけに限定されます。UNIX では、ライセンス検索パスの要素がコロンで区切られ、Windows では、要素がセミコロンで区切られます。

`lmgrd` や `lmutil` (`lmstat`、`lmdown` など) の `-c` オプションは、`LM_LICENSE_FILE` 環境変数の設定よりも優先されます。



注意：アプリケーションによっては、`LM_LICENSE_FILE` 環境変数を認識しないものがあります。特に、`FLEXenabled` Java アプリケーションでは認識されません。

参照

[複数のライセンス・ファイルの管理](#) – `LM_LICENSE_FILE` の詳細
環境変数

ライセンス・ファイルのフォーマットの概要

ライセンス・ファイルの先頭に `SERVER` 行が 1 行または `SERVER` 行が 3 行 (3 サーバ冗長化に設定されている場合) あり、その後に 1 行以上の `VENDOR` 行と、1 行以上の `FEATURE` 行または `INCREMENT` 行が続きます。場合によっては、ライセンス・ファイルに `SERVER` 行も `VENDOR` 行も必要ないことがあります。

8 ビットのラテン語ベースの文字はライセンス・ファイル、オプション・ファイル、ログ・ファイル、および `FLEXenabled` アプリケーション環境で完全にサポートされます。

`SERVER` 行および `VENDOR` 行の要件の詳細については、「[カウント・ライセンスと非カウント・ライセンスの比較](#)」を参照してください。

ライセンス・ファイル内の次の要素が修正可能です。

- SERVER 行の場合
 - SERVER 行のホスト名
 - TCP/IP ポート番号
 - HEARTBEAT_INTERVAL および PRIMARY_IS_MASTER プロパティ
- VENDOR 行の場合
 - ベンダー・デーモンへのパス
 - オプション・ファイルのパス
 - TCP/IP ポート番号 (ファイアウォールをサポートする場合のみ)
- USE_SERVER 行
- 機能定義行の場合
 - FEATURE 行の *keyword=value* ペアの値 (*keyword* が小文字で指定されている場合)
 - 長い行を分割する場合は、行継続文字である **¥** を使用できます。

参照

[3 サーバ冗長化の使用](#)

[カウント・ライセンスと非カウント・ライセンスの比較](#)

ライセンス・ファイルの構文

この項では、SERVER 行と VENDOR 行を含む、ライセンス・ファイルのコンテンツについて説明します。以下は、1 つの VENDOR 名が 2 つの機能を使用する場合のライセンス・ファイルの例です。

```
SERVER my_server 17007ea8 1700
VENDOR sampled
FEATURE f1 sampled 1.000 01-jan-2008 10 SIGN="<...>"
FEATURE f2 sampled 1.000 01-jan-2008 10 SIGN="<...>"
```

この例では、**hostid** が **17007ea8** のライセンス・サーバ **my_server** は、各機能 (**f1** および **f2**) について 10 のフローティング・ライセンスをネットワーク上のどのユーザにも提供できます。

SERVER 行

SERVER 行では、ライセンス・サーバのホスト名と **hostid**、およびライセンス・サーバ・マネージャ (**lmadmin** または **lmgrd**) の TCP/IP ポート番号を指定します。通常、各ライセンス・ファイルには 1 つの SERVER 行があります。SERVER 行が 3 行ある場合は、3 サーバ冗長化で構成されるライセンス・サーバを使用していることを示します。SERVER 行がない場合は、ライセンス・ファイル内のすべての機能定義行が非カウントであることを示します。

SERVER 行の **hostid** は、すべての機能定義行のライセンス・キーまたはシグニチャの計算に使用されます。このため、SERVER 行は、すべての機能定義行とともに、ソフトウェア・パブリッシャから送信されたときのままの状態でも保持する必要があります。

SERVER 行のフォーマットは次のとおりです。

```
SERVER host hostid [port] [PRIMARY_IS_MASTER] [HEARTBEAT_INTERVAL=seconds]
```

次に例を示します。

```
SERVER my_server 17007ea8 21987
```

次の表に、この行の属性を示します。

表 7-2: SERVER 行のフォーマット

フィールド	説明
host	システムのホスト名または IP アドレス。UNIX の hostname または uname -n コマンドから返される文字列。NT/2000/XP では ipconfig /all、Windows 95/98/ME では winipcfg /all からホスト名が返される。
hostid	通常、lmhostid コマンドから返される文字列。これは、ソフトウェア・サプライヤのみが変更可能。
port	<p>使用する TCP/IP ポート番号。有効な番号は、0 ~ 64000 の未使用のポート番号。UNIX では、ポート 1024 以上を選択する。1024 未満は権限付きのポート番号である。TCP/IP ポート番号が指定されない場合は、27000 ~ 27009 の範囲内のデフォルト・ポートのいずれかが使用される。</p> <p>SERVER 行が 3 サーバ冗長化として構成されたライセンス・サーバを定義する場合は、ポート番号を指定する必要がある。</p>
PRIMARY_IS_MASTER	<p>3 サーバ冗長化として構成されたライセンス・サーバとともに使用して、プライマリ・サーバとセカンダリ・サーバとの間でマスタの制御がどのように移行されるかを示す。</p> <ul style="list-style-type: none"> これを設定すると、プライマリ・サーバがダウンしても、プライマリ・サーバが復活すると、プライマリ・サーバがマスタのままとなる。 これを設定しないと、プライマリ・サーバがダウンした場合、セカンダリ・サーバがマスタとなり、プライマリ・サーバが復旧してもセカンダリ・サーバがマスタのままとなる。セカンダリ・ライセンス・サーバが故障した場合にのみ、プライマリ・サーバは再びマスタとなることができる。 <p>プライマリ・サーバとセカンダリ・サーバの両方がダウンした場合、ライセンスは提供されない。第 3 のサーバがマスタになることはない。</p> <p>このパラメータは省略可能で、ライセンス・ファイルの最初の SERVER 行に配置する。このパラメータを使用するには、バージョン 10.8 以降のベンダ・デーモンが実行されている必要がある。</p>

表 7-2: SERVER 行のフォーマット

フィールド	説明
HEARTBEAT_INTERVAL = <i>seconds</i>	<p>3 サーバ冗長化として構成されたライセンス・サーバとともに使用され、3 つの構成内の別のライセンス・サーバがシャットダウンする前に、そのサーバからハートビートの受け取りを待つ時間を示す。<i>seconds</i> の値は、タイムアウトを計算するために次の等式で使用される。</p> <ul style="list-style-type: none">timeout = (3 x <i>seconds</i>) + (<i>seconds</i> - 1) <p>指定しない場合、<i>seconds</i> のデフォルト値は 20 で、79 秒の実際のタイムアウト値と同等となる。<i>seconds</i> 値の有効な値は 0 ~ 120 まで。</p> <p>このパラメータは省略可能で、ライセンス・ファイルの最初の SERVER 行に配置する。このパラメータを使用するには、バージョン 10.8 以降のベンダ・デーモンが実行されている必要がある。</p>

参照

3 サーバ冗長化の使用

VENDOR 行

VENDOR 行では、デーモンの名前とパスを指定します。1mgrd ではこの行を使用してベンダ・デーモンを起動します。また、ベンダ・デーモンはこの行を読み取ってそのオプション・ファイルを見つけます。VENDOR 行のフォーマットは次のとおりです。

```
VENDOR vendor [vendor_daemon_path]*  
                [[OPTIONS=]options_file_path] [[PORT=]port]
```

各パラメータの意味は、次のとおりです。

表 7-3: VENDOR 行のフォーマット

フィールド	説明
<i>vendor</i>	ファイル内の機能処理のために使用されるベンダ・デーモンの名前。この名前は変更できない。
<i>vendor_daemon_path</i>	<p>このデーモンの実行プログラムへのオプションのパス。一般的に、ライセンス管理者は任意のディレクトリにベンダ・デーモンをインストールできる。ただし、ライセンス・サーバのローカル・ディレクトリにインストールすることが推奨される。</p> <p>指定しない場合、1mgrd は次の場所でベンダ・デーモンを探す。</p> <ul style="list-style-type: none">現在のディレクトリ1mgrd の \$PATH 環境変数で指定されているパス1mgrd が存在するディレクトリ <p><i>vendor_daemon_path</i> がブランクの場合は、オプションまたは TCP/IP ポート番号を指定するために OPTIONS= および PORT= 文字列が必要になる。</p>

表 7-3: VENDOR 行のフォーマット

フィールド	説明
<code>options_file_path</code>	このデーモンに対するオプション・ファイルへのフル・パス。オプション・ファイルは必要ない。 省略すると、ベンダ・デーモンはデフォルトでライセンス・ファイルと同じディレクトリにある <code>vendor.opt</code> という名前のファイル(ここで <code>vendor</code> はベンダ・デーモンの名前)を探す。
<code>port</code>	ベンダ・デーモンの TCP/IP ポート番号。 <code>port</code> が指定されない場合、実行時にオペレーティング・システムによってデフォルトが選択される。インターネット・ファイアウォールを使用するサイトでは、デーモンが使用する TCP/IP ポート番号を指定する必要がある。TCP/IP ポート番号を VENDOR 行で指定すると、ベンダ・デーモンの起動が遅くなる場合がある。

参照

オプション・ファイルの内容の詳細については、「[オプション・ファイルの管理](#)」を参照。

バージョン 6.0 以降

VENDOR sampled

USE_SERVER 行

USE_SERVER 行は引数を取らず、ライセンス・サーバにも影響を与えません。アプリケーションが USE_SERVER 行を確認すると、SERVER 行より前の内容を除きライセンス・ファイルの内容はすべて無視され、チェックアウトの検証がベンダ・デーモンに転送されます。

ライセンス・サーバの使用時にパフォーマンスが向上するため、USE_SERVER を使用することをおすすめします。非カウント機能の場合は、デーモンによる使用状況を強制的にログに記録するために USE_SERVER が使用されます。

FEATURE 行および INCREMENT 行

FEATURE 行および INCREMENT 行には、製品のライセンス・モデルが記述されます。特定の機能名の最初の FEATURE 行だけが、ベンダ・デーモンにより処理されます。同じ機能の追加のコピーが必要な場合(複数のノード・ロックされたカウント機能が必要な場合など)、複数の INCREMENT 行を使用する必要があります。INCREMENT 行は、次のフィールドに基づいてライセンス・グループ、つまり「プール」を作成します。

- 機能名
- バージョン
- DUP_GROUP
- FLOAT_OK
- HOST_BASED

- HOSTID
- PLATFORM
- USER_BASED
- VENDOR_STRING (パブリッシャによってプール・コンポーネントとして設定されている場合)

2つの行がこれらのフィールドのいずれかについて異なる場合、「ライセンス・プール」と呼ばれる新しいライセンス・グループがベンダ・デーモンに作成され、このグループは同じ機能名の他のライセンス・プールとは関係なくカウントされます。FEATURE 行にはライセンスの追加数は示されませんが、INCREMENT 行には常にライセンスの追加数が示されます。

基本的な機能定義行のフォーマットは次のとおりです。

```
{FEATURE|INCREMENT} feature vendor feat_version exp_date ¥
                        num_lic [optional_attributes] SIGN="<...>"
```

機能定義行キーワードの後には、順序が固定されている 6 つのフィールドが必要です。これらのフィールドは、ソフトウェア・パブリッシャによって定義され、変更できません。表 7-4 に、これらのフィールドを表示される順序で示します。

表 7-4: 機能定義行の必須フィールド

フィールド	説明
feature	ソフトウェア・パブリッシャにより機能に付けられた名前。
vendor	ベンダ・デーモンの名前、VENDOR 行にも表示される。指定されたデーモンはこの機能を実行する。
feat_version	このライセンスによってサポートされるこの機能のバージョン。このフィールドに yyyy.mmdd のフォーマットを使用した日付が含まれる場合、これにより、ライセンス・サーバ・マネージャ lmadm in でアラートとして設定可能な日付ベースのバージョンが定義される。
exp_date	dd-mmm-yyyy のフォーマット (07-may-2008 など) で表されるライセンスの有効期限。 注意: exp_date が文字列 "permanent" または年が 0 (00、000、0000) の場合、ライセンスは無期限になる。
num_lic	この機能の使用中のライセンス数。num_lic に文字列 "uncounted" または 0 を指定すると、この機能のライセンスはカウントされず、ライセンス・サーバは不要になるが、FEATURE 行に hostid が必要になる。「 カウント・ライセンスと非カウント・ライセンスの比較 」を参照。
SIGN= sign または	SIGN= この FEATURE 行を認証するためのシグニチャ。
AUTH= ...	パブリッシャが共通のベンダ・デーモン・テクノロジーを使用してこのベンダ・デーモンを配備している場合、シグニチャは AUTH= キーワード内に埋め込まれる。詳細については、パブリッシャに問い合わせること。

表 7-5 に、FEATURE 行または INCREMENT 行に表示される可能性のある属性を示します。これらの属性は、ライセンス・モデルを定義するためにソフトウェア・パブリッシャが指定します。FEATURE 行または INCREMENT 行にある属性は、そのまま残す必要があり、エンド・ユーザは変更できません。これらの属性の構文は keyword=value です。ここで keyword は大文字です。

value が二重引用符 (“...”) で囲まれている文字列の場合、文字列には引用符以外のどの文字でも含めることができます。

表 7-5: ソフトウェア・パブリッシャにより設定された属性

属性	説明
BORROW [= <i>n</i>]	特定の機能定義行に対するライセンスの借用を有効にする。 <i>n</i> は、ライセンスが借用される時間数を表す。デフォルトの借用期間は 168 時間 (1 週間)。
DUP_GROUP =...	構文は次のとおりです。 DUP_GROUP=NONE SITE [UHDV] U = DUP_USER H = DUP_HOST D = DUP_DISPLAY V = DUP_VENDOR_DEF UHDV の任意の組み合わせを使用でき、DUP_MASK は OR で組み合わせられる。たとえば、DUP_GROUP=UHD は、重複グループが (DUP_USER DUP_HOST DUP_DISPLAY) であり、そのため、同じホストと表示のユーザの場合、機能を追加使用しても追加のライセンスは消費されないことを意味する。
FLOAT_OK [= <i>server_hostid</i>]	特定の機能定義行に対して FLOAT_OK を使用して FLEXnet ID ドングル 経由のモバイル・ライセンスを有効にする。この機能定義行も、FLEXnet ID ドングル に対してノードロックされる必要がある。 FLOAT_OK= <i>server_hostid</i> が FEATURE 行で指定されている場合は、次のようになる。 <i>server_hostid</i> は、ライセンス・ファイルの SERVER 行に表示されるものと同じホストを参照する必要がある。 ライセンス・サーバは、lmhostid が返す hostid と FLOAT_OK を使用して指定された <i>server_hostid</i> が同じシステムでのみ実行される。
HOSTID = " <i>hostid1</i> [<i>hostid2</i> ... <i>hostidn</i>]"	機能行をバインドするホストの ID。 <i>hostid</i> は、lmhostid ユーティリティで決定される。このフィールドは、非カウント・ライセンスでは必須だが、カウント・ライセンスでも使用できる。詳細については、「 サポートされるプラットフォームの hostid 」を参照。
HOST_BASED [= <i>n</i>]	ホスト名はオプション・ファイルの INCLUDE 文で指定する必要がある。また、ホストの数は、 <i>num_lic</i> 、または = <i>n</i> で指定された数に制限される。
ISSUED = <i>dd-mmm-yyyy</i>	発行日。
ISSUER ="..."	ライセンスの発行者。
NOTICE ="..."	知的財産権の通知に使用されるフィールド。
ONE_TS_OK	ノードロックされた非カウント・ライセンスがリモート・デスクトップで実行されているアプリケーションにより使用される場合に検出する。

表 7-5: ソフトウェア・パブリッシャにより設定された属性 (続き)

属性	説明
OVERDRAFT=n	オーバードラフト・ポリシーを使用して、ソフトウェア・パブリッシャは、ユーザが購入したライセンスに加え、使用を許可される追加のライセンス数を指定できる。これにより、ユーザは「一時的なオーバードラフト」状態である場合にサービスを拒否されない。使用がこのライセンス制限を超えると、FLEXnet Manager レポート・ツールにより報告される。
PLATFORMS="..."	使用は、リストされているプラットフォームに制限される。
SN=serial_num	シリアル番号。FEATURE 行または INCREMENT 行の識別に使用される。
START=dd-mmm-yyyy	開始日付。
SUITE_DUP_GROUP=...	DUP_GROUP と似ているが、パッケージ・スイートに対して FEATURE 行を有効化している場合にのみ影響する。パッケージの合計ユーザ数をライセンス数に制限し、スイートをチェックアウトしているユーザ間でパッケージを共有できる。
SUPERSEDE= "f1 f2 ..."	これが表示される場合、ISSUED= で指定された日付より前に発行されたすべてのライセンスはこの行で置き換えられ、無効になる。
SUPERSEDE_SIGN= {f1:xxxx, f2:xxxx}	値として定義されたすべての機能定義行またはパッケージ行のライセンス・モデルを無効にする。
SUPERSEDE_SIGN= {p1:xxxx, p2:xxxx}	
TS_OK	FLEXnet Publisher ライセンス・ツールキットによって、ノードロックされた非カウント・ライセンスが Windows Terminal Server で実行されていることが検出される。アプリケーションを Terminal Server クライアント・ウィンドウ経由で実行するには、FEATURE 行に TS_OK を追加する必要がある。TS_OK がない場合、Terminal Server クライアントで実行するユーザはライセンスが拒否される。
USER_BASED[=n]	ユーザはオプション・ファイルの INCLUDE 文で指定する必要がある。また、ユーザの数は、num_lic または =n で指定された数に制限される。
VENDOR_STRING="..."	二重引用符で囲まれた、ソフトウェア・パブリッシャにより定義されたカスタム値。

表 7-6 に示されている属性はオプションで、ライセンス管理者が制御します。これらの属性の構文は `keyword=value` です。ここで `keyword` は小文字です。

表 7-6: オプションの機能定義行の属性

属性	説明
<code>asset_info="..."</code>	資産管理のライセンス管理者から提供される追加の情報。
<code>dist_info="..."</code>	ソフトウェア販売代理店から提供される追加の情報。
<code>sort=nnn</code>	ライセンス・ファイル行のソート順を指定する。「ソートの規則」を参照。
<code>user_info="..."</code>	ライセンス管理者から提供される追加の情報。
<code>vendor_info="..."</code>	ソフトウェア・パブリッシャから提供される追加の情報。

例

```
FEATURE sample_app sampled 2.300 31-dec-2008 20 ¥
SIGN("<...>")
INCREMENT f1 sampled 1.000 permanent 5 ¥
HOSTID=INTERNET=195.186.*.* NOTICE="Licensed to ¥
Sample corp" SIGN("<...>")
```

ソートの規則

機能定義行は、ライセンス・ファイルから読み込まれるときに自動的にソートされます。デフォルトのソート規則は次のとおりです。

1. ライセンス・ファイル。ライセンス検索パス内のファイルには自動ソートは実行されない。
2. 機能名。
3. FEATURE の後に INCREMENT。
4. 非カウントの後にカウント。
5. バージョン。新しいバージョンの後に古いバージョン。
6. 発行日。逆の順序で最も新しい日付が最初。日付は ISSUED= または START= から取得される。
7. 上記以外は元の順序が維持される。

自動ソートをオフにするには、`sort=nnn` を機能定義行に追加します。ここで `nnn` はすべての行で同じです。`nnn` は、相対的なソート順を指定します。デフォルトのソート順の値は 100 です。ソート順の値が 100 未満の行は、この属性が指定されていないすべての行の前にソートされ、ソート順の値が 100 を超える行は、属性が指定されていないすべての行の後に表示されます。同じ値のすべての行は、ファイル内に表示されている順序でソートされます。

FEATURE および INCREMENT 行のフォーマットの変更

機能定義行のフォーマットの重要な変更と、追加のキーワードが導入された場合の変更を次に示します。

- バージョン 7.1 以前の機能定義行のフォーマットでは、次のように *license_key* を使用する。

```
{FEATURE|INCREMENT} feature vendor feat_version exp_date ¥  
num_lic [optional_attributes] SIGN="<...>"
```

バージョン 7.1 以前のフォーマットは、現在のリリースで認識される。

- SIGN= キーワードは、7.1 バージョンで導入されている。
- バージョン 7.1 ~ バージョン 8.0 のクライアント・ライブラリとベンダ・デーモンの場合は、機能定義行に SIGN= シグネチャが必要で、バージョン 8.1 以前のバージョンとの下位互換性のために、次のように *license_key* を含めることができる。

```
{FEATURE|INCREMENT} feature vendor feat_version exp_date ¥  
num_lic [license_key] [optional_attributes] SIGN="<...>"
```

- license_key* は、バージョン 8.1 のクライアント・ライブラリとベンダ・デーモンで廃止されている。
- exp_date* の “permanent” キーワードは、バージョン 6 のクライアント・ライブラリから導入されている。
- num_lic* の “uncounted” キーワードは、バージョン 6 のクライアント・ライブラリから導入されている。
- BORROW キーワードは、バージョン 8.0 のクライアント・ライブラリとベンダ・デーモンから導入されている。
- FLOAT_OK キーワードは、バージョン 8.0 クライアント・ライブラリとベンダ・デーモンから導入されている。
- TS_OK キーワードは、バージョン 8.0 のクライアント・ライブラリとベンダ・デーモンから導入されている。
- AUTH キーワードは、バージョン 10.8 のクライアント・ライブラリとベンダ・デーモンから導入されている。

PACKAGE 行

PACKAGE 行は、次の 2 つの異なるニーズをサポートするために使用されます。

- 製品スイートのライセンスを付与する。
- 大部分が FEATURE 行の同じ引数を共有する多数の機能を含むライセンス・ファイルをより効率的な方法で配布する。

PACKAGE 行そのものは何のライセンスも付与しません。パッケージ全体のライセンスを付与するには、対応する機能定義行が必要です。PACKAGE 行は、ライセンスに関係なく、製品とともにソフトウェア・パブリッシャから提供されます。後でそのパッケージのライセンスを購入するときに、1 つ以上の対応する機能定義行によって PACKAGE 行が有効化されます。

例

```
PACKAGE package vendor [pkg_version] COMPONENTS=pkg_list ¥
  [OPTIONS=SUITE] [SUPERSEDE=["p1 p2 ..."]] ISSUED=date]
SIGN="<...>"
```

表 7-7 に、PACKAGE 行のフィールドを示しますフィールドはリストされている順序で表示されます。

表 7-7: PACKAGE 行のフィールド

フィールド	説明
package	パッケージの名前。対応する機能定義行の名前は同じにする必要がある。
vendor	このパッケージをサポートするベンダ・デーモンの名前。
pkg_version	パッケージ・バージョンを指定するオプションのフィールド。指定された場合、有効化を行う機能定義行のバージョンは同じにする必要がある。
COMPONENTS=pkg_list	<p>パッケージ・コンポーネントのリスト。フォーマットは次のとおり。 feature[:version[:num_lic]]</p> <p>パッケージは、1 つ以上のコンポーネントで構成される必要がある。バージョンとカウントはオプションで、指定しない場合は、対応する機能定義行から値が取得される。OPTIONS=SUITE が設定されていない場合は、num_lic のみ有効になる。この場合、結果のライセンス数は、COMPONENTS 行の num_lic に、機能定義行のライセンス数を掛けた値になる。例：</p> <pre>COMPONENTS="comp1 comp2 comp3 comp4" COMPONENTS="comp1:1.5 comp2 comp3:2.0:4"</pre>
OPTIONS=SUITE	<p>オプション・フィールド。パッケージ・スイートを示すために使用される。</p> <p>このオプションを設定すると、借り出されているコンポーネント機能に加えて、パッケージと同じ名前の対応する機能が借り出される。</p> <p>このオプションを設定しない場合、パッケージが有効になると、パッケージと同じ名前の対応する機能が削除される。コンポーネント機能がチェックアウトされても、この機能はチェックアウトされない。</p>
OPTIONS=SUITE_RESERVED	<p>オプション・フィールド。このオプションを設定すると、パッケージ・コンポーネントのセットが予約される。1 つのパッケージ・コンポーネントが借り出されると、他のすべてのコンポーネントが同じユーザに対して予約される。</p>
SUPERSEDE ["p1 p2 ..."]	<p>オプション・フィールド。ISSUED 日付とともに使用する。同じパッケージ名のすべての PACKAGE 行を、dd-mmm-yyyy より前の ISSUED 日付で置き換える。</p>
ISSUED= dd-mmm-yyyy	<p>オプション・フィールド。SUPERSEDE とともに使用する。同じパッケージ名のすべての PACKAGE 行を、dd-mmm-yyyy より前の ISSUED 日付で置き換える。</p>
SIGN=sign または AUTH=...	<p>SIGN= この FEATURE 行を認証するためのシグニチャ。</p> <p>パブリッシャが共通のベンダ・デーモン・テクノロジーを使用してこのベンダ・デーモンを配備している場合、シグニチャは AUTH= キーワード内に埋め込まれる。詳細については、パブリッシャに問い合わせること。</p>

例

```
PACKAGE suite sampled 1.0 SIGN="<...>" ¥  
  COMPONENTS="comp1 comp2" OPTIONS=SUITE  
FEATURE suite sampled 1.0 1-jan-2008 5 SIGN="<...>"
```

これは標準的な OPTIONS=SUITE の例です。“comp1”と“comp2”の2つの機能があり、どちらもバージョン 1.0 で、それぞれが 5 つの有効期限のないライセンスを使用できます。“comp1”または“comp2”がチェックアウトされると、“suite”もチェックアウトされます。

```
PACKAGE suite sampled 1.0 SIGN="<...>"¥  
  COMPONENTS="apple:1.5:2 orange:3.0:4"  
FEATURE suite sampled 1.0 1-jan-2008 3 SN=123 SIGN="<...>"
```

この例では、コンポーネントのバージョンが機能バージョンを無効にし、コンポーネントで使用できるライセンス数は、“suite”の3つのライセンスとそのコンポーネントのライセンス数を掛けた値になります。結果は次のようになります。

```
FEATURE apple sampled 1.5 1-jan-2008 6 SN=123 SIGN="<...>"  
FEATURE orange sampled 3.0 1-jan-2008 12 SN=123 SIGN="<...>"
```

- 個々のファイルに PACKAGE 行を保存する機能は、バージョン 6 のクライアント・ライブラリから導入されている。
- *pkg_version* フィールドは、バージョン 7.1 以前のクライアント・ライブラリが必要である。
- AUTH キーワードは、バージョン 10.8 のクライアント・ライブラリとベンダ・デーモンから導入されている。

UPGRADE 行

```
UPGRADE feature vendor from_feat_version to_feat_version ¥  
exp_date num_lic [options ... ] SIGN="<...>"
```

すべてのデータは、FEATURE 行または INCREMENT 行と同じですが、*from_feat_version* フィールドが追加されています。UPGRADE 行では、古いバージョンから指定されたライセンス数までが削除され (\geq *from_feat_version*)、同じライセンス数の新しいバージョンが作成されます。

たとえば、次の 2 行によって、**f1** の 3 つのバージョン 1.0 ライセンスと **f1** の 2 つのバージョン 2.0 ライセンスが指定されます。

```
INCREMENT f1 sampled 1.000 1-jan-2008 5 SIGN="<...>"  
UPGRADE f1 sampled 1.000 2.000 1-jan-2008 2 SIGN="<...>"
```

UPGRADE 行は、直前の FEATURE または INCREMENT に対して、 \geq *from_feat_version* および $<$ *to_feat_version* のバージョン番号を指定して機能します。



注意：UPGRADE 行は、ノードロックされた非カウント・ライセンスでは機能しません。

10 進形式の FEATURE 行

ライセンスは 10 進形式で表すことができます。10 進形式にすると入力が簡単になり、多くの場合ライセンスを短くできるという利点があります。次は、判読できるフォーマットの簡単なデモ・ライセンスです。

```
FEATURE f1 sampled 1.00 1-jan-2008 0 HOSTID=DEMO SIGN="<...>"
```

同じ内容を 10 進形式で表すと、次のようになります。

```
sampled-f1-00737-55296-1825
```

必要に応じて、ライセンス・ファイルで 10 進形式の行を判読可能なフォーマットの行と組み合わせることができます。10 進形式のライセンスを判読可能なフォーマットに変換するには、`lminstall` コマンドを使用します。

参照

`lminstall` コマンドの詳細については、「[lminstall](#)」を参照。

ライセンス・ファイルの行の順序

ライセンス・ファイル内の行の順序は重要ではありません。これらは処理時にソートされ、多くの場合、最適な結果になります。ただし、バージョン 7.0 以前のバージョンの `FLEXenabled` アプリケーションとライセンス・サーバでは、ライセンス・ファイルの行の順序が暗黙で決められています。ライセンス・ファイルの行の順序については、次の推奨事項があります。

- FEATURE 行は、同じ機能の INCREMENT 行の前に置く。

FEATURE 行については、最初のカウント FEATURE 行のみがライセンス・サーバで使用されること、および FEATURE 行と INCREMENT 行がある場合は FEATURE 行が最初に表示されるという規則がある。

- 同じ機能に対して複数のカウント FEATURE 行が存在する場合は、必要な FEATURE 行が最初に表示されるようにする。

最初の行以外はすべて無視される。

- ノードロックされた、非カウント行は、同じ機能のフローティング行の前に置く。このようにしない場合、ノードロックされたライセンスではなくフローティング・ライセンスが使用される可能性があり、その結果他のユーザは拒否される。
- `USE_SERVER` 行の配置は動作に影響する。`USE_SERVER` 行の使用が推奨される。通常、`USE_SERVER` 行は `SERVER` 行の直後に置かれるが、`SERVER` によって処理されない非カウント・ライセンスは `USE_SERVER` 行の前に配置する。非カウント・ライセンスを必要とする各ユーザが、ファイルの現在のコピーに直接アクセスできるようにする。`SERVER` の直後に `USE_SERVER` を置くと、ユーザにライセンス・ファイルの最新コピーが必要なくなるという利点がある。

参照

[ソートの規則](#)

ライセンス・モデル

ライセンス・ファイルは、ソフトウェア・パブリッシャによって作成されます。ライセンス・ファイルでは、フローティング (同時) 使用であるか、ノードロック (カウントと非カウントの両方) であるか、またはフローティング、カウント、非カウントの組み合わせであるかが指定されます。

フローティング (同時) ・ライセンス

「フローティング・ライセンス」とは、ライセンス・ファイル内で指定された上限に達するまでは、そのネットワーク上のすべてのユーザが **FLEXenabled** アプリケーションを使用できるライセンスです (「同時使用」または「ネットワーク・ライセンシング」とも呼ばれます)。フローティング・ライセンスの場合は、個々の **FEATURE** 行に **HOST ID** はありません。フローティング・ライセンスを使用するには、ライセンスの同時使用数をカウントするために、ライセンス・サーバ・マネージャとベンダ・デーモンが実行されている必要があります。

フローティング・ライセンスを指定するライセンス・ファイルの例を次に示します。

```
SERVER lulu 17007ea8
VENDOR sampled
FEATURE f1 sampled 1.00 1-jan-2008 SIGN="<...>"
FEATURE f2 sampled 1.00 1-jan-2008 6 SIGN="<...>"
FEATURE f3 sampled 1.00 1-jan-2008 1 SIGN="<...>"
```

このライセンス・ファイルでは、ライセンス・サーバ **lulu** にアクセス可能ならばネットワーク上のどこからでも、機能 **f1** の 2 ライセンス、機能 **f2** の 6 ライセンス、機能 **f3** の 1 ライセンスが使用可能であることが指定されています。ライセンス・サーバ・マネージャの **lmadmin** または **lmgrd** は、デフォルトの **TCP/IP** ポートの 1 つを使用します。

ノードロック・ライセンス

ノードロックとは、FLEXenabled アプリケーションを 1 台または複数のマシン上でのみ使用できることを意味します。ノードロック・ライセンスは、特定のホストにノードロックされているすべての FEATURE 行にホスト ID があります。ノードロック・ライセンスのタイプには、非カウントとカウントの 2 つがあります。

ライセンス数の値をゼロ (0) または uncounted に設定されている場合は、非カウントのライセンスであり、無制限に使用できます。この設定では、機能の同時使用数のカウントは行われないので、ライセンス・サーバは不要です。

次に示すライセンス・ファイルは、ホスト ID が **17007ea8** から **1700ab12** の範囲内のマシン上で機能 **f1** を無制限に使用できることを示します。

```
FEATURE f1 sampled 1.000 1-jan-2008 uncounted HOSTID=17007ea8 SIGN="<...>"  
FEATURE f1 sampled 1.000 1-jan-2008 uncounted HOSTID=1700ab12 SIGN="<...>"
```

あるいは、この 2 つの FEATURE 行が、次のように「ホスト ID リスト £ としてソフトウェア・パブリッシャから発行されている可能性もあります。

```
FEATURE f1 sampled 1.000 1-jan-2008 uncounted HOSTID="17007ea8 1700ab12" SIGN="<...>"
```

ここに示したものの以外の FEATURE 行がこのライセンス・ファイルにない場合は、ライセンス・サーバ・マネージャもベンダ・デーモンも不要で、起動する必要はありません。

次に示すライセンス・ファイルは、機能 **f1** の 3 ライセンスがホスト ID **1300ab43** のマシンにロックされていることを示します。ライセンス・サーバとライセンスが同じマシンにロックされているので、デーモンは、FLEXenabled アプリケーションを実行しているマシンと同じマシン上で実行されます。

```
SERVER lulu 1300ab43 1700  
VENDOR sampled /etc/sampled  
FEATURE f1 sampled 1.00 1-jan-2008 3 HOSTID=1300ab43 SIGN="<...>"
```

ノードロック・ライセンスとフローティング・ライセンスの混在

非カウントのノードロック・ライセンスと同時使用ライセンスを、同じライセンス・ファイル内に混在させることができます。

次に示すライセンス・ファイルは、機能 **f1** がマシン **17007ea8** 上および **1700ab12** 上で無制限に使用できることを示します。この他に、ネットワークのその他のマシン上で機能 **f1** の 2 ライセンスが使用可能です。

```
SERVER lulu 17001234 1700  
VENDOR sampled C:\flexlm\sampled.exe  
FEATURE f1 sampled 1.00 1-jan-2005 uncounted HOSTID=17007ea8 SIGN="<...>"  
FEATURE f1 sampled 1.00 1-jan-2005 uncounted HOSTID=1700ab12 SIGN="<...>"  
FEATURE f1 sampled 1.00 1-jan-2005 2 SIGN="<...>"
```

この設定では、3 つ目の FEATURE 行のライセンスがカウントされるので、ライセンス・サーバ・マネージャとベンダ・デーモンが必要です。

カウント・ライセンスと非カウント・ライセンスの比較

ライセンス・サーバが必要かどうかは、(ライセンス・ファイルで定義される)ライセンス・モデルによって決まります。機能定義のすべての行でライセンス・カウントがゼロ (0) または非カウントに設定されている場合、ライセンス・サーバは不要です。このタイプのライセンスを「非カウント」と呼びます。一方、機能のライセンス数が 0 以外の場合は、それらのライセンスをカウントするライセンス・サーバが必要です。ソフトウェア・パブリッシャが FLEXnet Publisher ライセンス・ツールキットをライセンス・サーバなしで使用したい場合は、非カウント・ライセンスを発行する必要があります。

ライセンス・サーバでは非カウント・ライセンスの処理も可能です。これは通常、次の目的で行います。

- すべてのライセンス要求のトランザクションをレポート・ログに記録する。この記録を使用して、FLEXnet Manager でレポートを作成できます。
- オプション・ファイルの制約をライセンスに適用する。

非カウント・ライセンスがサーバで処理されるようにするには、**SERVER** 行をライセンス・ファイルに追加し、**SERVER** 行の直後に **USE_SERVER** 行を記述します。ベンダ・デーモンによって、非カウント・ライセンスの処理が行われます。**USE_SERVER** 行は、ライセンスを要求するアプリケーションがライセンス・サーバにアクセスして承認を受ける必要があることを示します。

モバイル・ライセンス

エンド・ユーザが、ライセンス・サーバに常時接続していないコンピュータでアプリケーションを実行することがよくあります。次のような状況が考えられます。

- ラップトップ・コンピュータで作業する
- コンピュータを職場と自宅で使用する
- ライセンス・サーバに接続していない複数のコンピュータで作業する

FLEXnet Publisher ライセンス・ツールキットのライセンスは以下のモバイル・ライセンスのいずれかに対応しています。

- ラップトップにノードロックされたライセンス
- FLEXnet ID ドングルにノードロックされたライセンス
- FLOAT_OK キーワード付きで FLEXnet ID ドングルにノードロックされたライセンス
- BORROW キーワードによるライセンスの借用
- ユーザ名にノードロックされたライセンス
- プリペイド・ライセンス・プールからの充当

エンド・ユーザがこれら以外の方法を使用してライセンスを移動する場合は、ライセンスの再ホストを行うことになります。これは、新しいクライアント・コンピュータごとにソフトウェア・パブリッシャが新しいノードロック・ライセンス・ファイルを生成することを意味します。再ホストでは移動のたびにソフトウェア・パブリッシャが関与する必要があるため、管理上のオーバーヘッドが生じます。

ラップトップ・コンピュータへのノードロック

特定のラップトップ・コンピュータで排他的に使用するライセンスは、そのコンピュータに関連付けられているアドレスにノードロックされます。ライセンス・ファイルはラップトップ・コンピュータに置かれます。

FLEXnet ID ドングルへのノードロック

Windows マシンの間で移動するライセンスは、FLEXnet ID ドングル（パラレル・ポートまたは USB ポートに接続するドングル）にノードロックされます。このライセンスをマシン間で移動するときは、ライセンス・ファイルのコピーを各マシンにインストールしておき、FLEXnet ID ドングルを他のマシンに付け替えます。ライセンスは FLEXnet ID ドングルに関連付けられているため、FLEXnet ID ドングルが装着されているマシンだけがライセンスを使用できます。

FLOAT_OK キーワード付き FLEXnet ID ドングルへのノードロック

このライセンス移動方法では、FLEXnet ID ドングルがライセンス・サーバを定義し、ライセンスがネットワーク上を自由に移動するため、FLEXnet ID ドングルにノードロックされたライセンスを使用するよりも便利です。FLEXnet ID ドングルにノードロックされた FLOAT_OK キーワード付きライセンスは、Windows 上で FLEXenabled アプリケーションとライセンス・サーバの両方が実行されている場合にのみサポートされます。

ソフトウェア・パブリッシャは、FLEXnet ID ドングルにノードロックされたライセンス・ファイルを発行します。このファイルの FEATURE 行に FLOAT_OK キーワードと FLEXnet ID ドングルが含まれています。モバイル・ライセンスのインスタンスごとに、FLOAT_OK キーワードが含まれている 1 つの FEATURE 行と、FLEXnet ID ドングルが 1 つ必要です。FLEXnet ID ドングルをライセンス・サーバに接続すると、ライセンスはネットワーク上を移動します。FLEXnet ID ドングルをライセンス・サーバから取り外すと、ライセンスはそのスタンドアロン・コンピュータ以外で使用できなくなります。

この移動法は、パラレルまたは USB FLEXnet ID ドングルをサポートします。複数の USB ドングルをコンピュータに接続するほうが簡単なので、USB FLEXnet ID ドングルをおすすめします。

FLOAT_OK キーワード付きの FLEXnet ID ドングルの開始

ソフトウェア・パブリッシャは顧客に対して FLEXnet ID ドングル、ドングル・ドライバのインストーラ、およびその FLEXnet ID ドングルにノードロックされたライセンス・ファイルを発行します。このファイルの FEATURE 行に、FLOAT_OK キーワードが含まれています。その後、ライセンス管理者は次の操作を行います。

1. ライセンス・サーバにライセンス・ファイルをインストールする
2. すべての FLEXnet ID ドングルをライセンス・サーバに接続する
3. ライセンス・サーバに FLEXnet ID ドングルドライバをインストールする
4. ライセンス・サーバを起動するかライセンス・ファイルを再読み込みする

FLEXnet ID ドングルをライセンス・サーバに接続すると、関連付けられたノードロック・ライセンスはネットワーク上を移動します。カウントされていない FLOAT_OK ノードロック FEATURE 行は、ネットワーク上で利用可能な間、各行が 1 とカウントされます。

移動ライセンス・プールから、ネットワークに接続されていないコンピュータにライセンスを移動するには、次の手順を実行します。

1. FLOAT_OK 付きのノードロックされた FEATURE 行を含んでいるライセンス・ファイルを、ライセンス・サーバからクライアント上のライセンス・ファイルの場所 (FLEXenabled アプリケーションがライセンス・ファイルを探すディレクトリ) にコピーします。
2. ノードロックされた FEATURE 行と一致する FLEXnet ID ドングルをライセンス・サーバからクライアント・コンピュータに移します。FLEXnet ID ドングルをライセンス・サーバから取り外すと、このライセンスはネットワーク上で使用できなくなります。
3. FLEXnet ID ドングルドライバがインストールされていない場合は、クライアント・コンピュータにインストールします。
4. クライアント・コンピュータをネットワークから切断します。これで、コンピュータに FLEXnet ID ドングルが接続されていれば、ネットワークに接続していなくてもライセンスを使用できるようになります。

FLOAT_OK を使用する FLEXnet ID ドングルライセンスの返却



Task ライセンスをライセンス・サーバに返却し、再びネットワーク上で移動できるようにするには、次の手順に従います。

1. クライアント・マシンから FLEXnet ID ドングルを取り外し、ライセンス・サーバに付け替えます。
2. 移動ライセンスを提供するライセンス・サーバで `lmreread` を実行してライセンス・ファイルを再読み込みします。FLEXnet ID ドングルがライセンス・サーバに返却された後、FLOAT_OK ライセンスは `lmreread` が実行されるまでネットワーク上を移動しません。

FLOAT_OK 付きの FLEXnet ID ドングルの使用例

以下はライセンス・ファイルの例です。出荷時の FLEXnet ID ドングルは、FLEXID=7-b28520b9 と FLEXID=7-b2857678 の 2 つです。

```
SERVER myhost ANY
VENDOR sampled
FEATURE f1 sampled 1.0 permanent uncounted FLOAT_OK ¥
HOSTID=FLEXID=7-b28520b9 SIGN="<...>"
FEATURE f1 sampled 1.0 permanent uncounted FLOAT_OK ¥
HOSTID=FLEXID=7-b2857678 SIGN="<...>"
```

ユーザはライセンス・ファイルをライセンス・サーバにインストールし、2 つの FLEXnet ID ドングルライセンス・サーバに接続します。ライセンス・サーバに接続した非カウントの FLOAT_OK ライセンスは、それぞれネットワーク上を移動し、1 つ分の使用が許可されます。これにより、ネットワークにおいて最大 2 人のユーザまで **f1** を使用できます。ただし、ライセンス・サーバ上ではライセンスの使用が許可されていないため、使用できません。

ユーザが自宅で仕事をする場合、FLEXID=7-b28520b9 にノードロックされた FEATURE 行を含むライセンス・ファイルをインストールし (1 回のみ行う必要がある)、FLEXnet ID ドングル FLEXID=7-b28520b9 をライセンス・サーバからクライアントに転送し、FLEXnet ID ドングルドライバをクライアント・コンピュータにインストールします (これも 1 回のみ行う)。次にクライアント・コンピュータをネットワークから切断し、転送された FLOAT_OK ライセンスをクライアント・コンピュータ上で使用します。ライセンス・サーバは、残りの 1 つの FLOAT_OK ライセンスのみをネットワーク上で移動できるようにします。

ドングルをライセンス・サーバに返却した後、ライセンス管理者が `lmreread` を実行することにより、返却されたライセンスは再び移動可能になります。

- FLOAT_OK キーワードは、バージョン 8.0 のクライアント・ライブラリとライセンス・サーバ・マネージャ、ベンダ・デーモンで導入されました。FLOAT_OK を使用するには、すべてのコンポーネントをバージョン 8.0 以上にする必要があります。

BORROW によるライセンスの借用

ライセンス・サーバに一時的に接続されるコンピュータで使用するライセンスは、BORROW キーワードを使用して移動ライセンスとして発行できます。BORROW ライセンスは、特別なチェックアウトを行ってライセンス・サーバから借用し、後でライセンス・サーバに接続していないコンピュータ上でアプリケーションを実行することができます。ライセンスを借用するには、ライセンスの借用がソフトウェア・パブリッシャによって有効にされている必要があります。

ライセンスを借用する場合、ソフトウェア・パブリッシャは、FEATURE 行に BORROW キーワードが含まれる移動ライセンスを発行します。ユーザは、借用したライセンスの返却期限を指定し、ネットワークに接続されたときにクライアント・コンピュータに借用情報を出力するアプリケーションを実行します。借用ライセンスは、ライセンス・サーバではチェックアウトの状態に保たれます。FLEXenabled アプリケーションは、自動的にローカルの借用情報を使用して、借用期間中にチェックアウトを実行します。ソフトウェア・パブリッシャによって有効にされた借用ライセンスは、借用期間が終了する前に返却することもできます。借用期間が終了した場合、または借用ライセンスを早期返却した場合、ローカルの借用データによるチェックアウトはできなくなり、ライセンス・サーバは借用ライセンスを使用可能なライセンス・プールに返却します。ライセンス・サーバと FLEXenabled アプリケーションの実行マシンの間で時刻を合わせる必要はありません。

ライセンス借用の開始

FEATURE 行に BORROW キーワードを含むライセンス・ファイルがソフトウェア・パブリッシャが発行されてライセンス借用が有効になっている場合、ユーザは次の 3 とおりの方法でライセンスの借用を開始できます。

- アプリケーションの借用インターフェースを使用する
- `lmborrow` ユーティリティを実行して `LM_BORROW` を設定する
- `LM_BORROW` 環境変数を直接設定する

アプリケーション・インタフェース

アプリケーションに借用インタフェースがある場合にのみ、この方法でライセンス借用を開始できます。これに関する情報はソフトウェア・パブリッシャから提供されます。

lmborrow ユーティリティの実行

lmborrow は lmutil/lmtools ユーティリティの 1 つです。借用を開始するには、lmborrow をコマンド・ラインまたは lmtools から実行します。

```
lmborrow {vendor|all} enddate [time]
```

vendor は、借用するライセンスを提供するベンダ・デーモンです。また、*all* はライセンス・サーバ内のすべてのベンダ・デーモンを指定します。*enddate* は、ライセンスを返却する日付 (*dd-mm-yyyy*) です。*time* は、24 時間形式 (*hh:mm*) の FLEXenabled アプリケーションのローカル時刻で指定します。この値は省略可能です。*time* が指定されていない場合、チェックアウトは所定の終了日まで続きます。

次に例を示します。

```
lmborrow sampled 20-aug-2007 13:00
```

LM_BORROW 環境変数の直接設定

lmborrow ユーティリティは、LM_BORROW を設定するユーザ・インタフェースです。設定は、レジストリ (Windows) または \$HOME/.flexlmborrow (UNIX) に対して行われます。次のように、LM_BORROW を環境変数として直接設定することもできます。

```
today:{vendor|all}:enddate[:time]
```

各パラメータの意味は、次のとおりです。

表 8-1: LM_BORROW 環境変数の引数

引数	説明
today	当日の日付 (<i>dd-mm-yyyy</i>) です。この日にチェックアウトが実行されると、ローカル借用情報が生成されます。この日付以外の日にチェックアウトが実行された場合、ローカル借用情報は生成されません。
vendor	借用するライセンスを提供するベンダ・デーモンです。 <i>all</i> は、ライセンス・サーバ内のすべてのベンダ・デーモンを指定します。
enddate	ライセンスを返却する日付 (<i>dd-mm-yyyy</i>) です。
time	(省略可能) <i>time</i> は、24 時間形式 (<i>hh:mm</i>) の FLEXenabled アプリケーションのローカル時刻で指定します。 <i>time</i> が指定されていない場合、チェックアウトは所定の終了日まで続きます。

次に例を示します。

```
LM_BORROW=15-aug-2006:sampled:20-aug-2006:13:00
```

この例では、sampled ベンダ・デーモンにより提供される 1 つまたは複数のライセンスが 2001 年 8 月 15 日に借用され、2001 年 8 月 20 日午後 1 時に返却が予定されています。

ライセンスの借用

ユーザが `lmborrow` を実行するか `LM_BORROW` を設定した日付およびマシンで、特定の機能に関するライセンスをネットワークに接続したまま借用するには、ライセンスのチェックアウトと借用を行うためのアプリケーションを実行します。この日にアプリケーションを複数回実行しても、ライセンスの借用は重複しません。借用の開始を設定した日付以外の日にアプリケーションを実行しても、ライセンスを借用できません。

たとえば、PageWizard 機能のライセンスを今日から 1 週間借用するとします。PageWizard 機能は sampled ベンダ・デーモンによって提供されます。本日、ネットワークに接続している間に、`lmborrow` を実行するか `LM_BORROW` を直接設定します。次に例を示します。

```
lmborrow sampled enddate
```

今日 `lmborrow` を実行した後、ネットワークに接続している間に、PageWizard 機能のライセンスをチェックアウトするアプリケーションを実行します。ライセンスをチェックアウトした後、アプリケーションを終了して、マシンをネットワークから切断します。ここでチェックアウトしたライセンスは、借用期間が終了するまで、ライセンス・サーバからチェックアウトされた状態を保ちます。つまり、借用期間が終了するまで、ネットワークから切断されたマシン上でそのライセンスを使用できます。チェックアウトしたライセンスは、借用期間中、チェックアウト状態を保ちます。借用期間の更新は、期間が終了するまで行えません。

借用期間のクリア

現在の借用期間 (`LM_BORROW` 環境変数で定義)に必要なライセンスをすべて借用した後、その他の機能のライセンスが借用されないようにするためには、`lmborrow -clear` を実行します。これにより、レジストリ (Windows) または `$HOME/.flexlmborrow` (UNIX) の `LM_BORROW` 設定がクリアされます。`lmborrow -clear` を実行しても、借用済みライセンスのローカル情報はクリアされません。

借用ステータスのチェック



Task 借用した機能に関する情報を出力するには、次の手順に従います。

1. 借用が行われたマシンで次のコマンドを発行します。

```
lmborrow -status
```

ステータスを確認する場合、機能を借用したシステムがネットワークに接続している必要はありません。

借用ライセンスの早期返却



Task 借用したライセンスを借用期間が終了する前に返却するには、次の手順に従います。

1. 借用先システムをネットワークに再接続します。
2. 借用を開始したのと同じマシンから次のコマンドを発行します。

```
lmborrow -return [-c license_file_list] feature
```

このオプションが使用できるかどうかは、ソフトウェア・パブリッシャによって異なります。借用ライセンスの早期返却がサポートされているかどうかについては、ソフトウェア・パブリッシャに直接問い合わせてください。

ライセンスを早期返却すると、返却したライセンスを提供しているベンダ・デーモンの LM_BORROW 設定がクリアされます。

ライセンス借用のサポート

ライセンス借用をサポートするオプション・ファイルのユーティリティとキーワードの詳細については、以下の項を参照してください。

- **lmborrow** ユーティリティ
- **lmdown** ユーティリティ
- **lmstat** ユーティリティ
- **BORROW_LOWWATER** キーワード
- **EXCLUDE_BORROW** キーワード
- **INCLUDE_BORROW** キーワード



注意：BORROW キーワードは、バージョン 8.0 のクライアント・ライブラリとライセンス・サーバ・マネージャ、ベンダ・デーモンで導入されました。BORROW を使用するには、すべてのコンポーネントをバージョン 8.0 以上にする必要があります。

ユーザ名へのノードロック

異なるマシン上で 1 人のユーザがライセンスを排他的に使用する場合は、ライセンスをユーザのユーザ名にノードロックできます。ライセンス・ファイルは、このユーザが作業する複数のシステムにコピーされます。ユーザが使用するユーザ名は、各システムで同じにする必要があります。この方法を使用するためには、各ユーザのユーザ名を組織内で一意にする必要があります。

プリペイド・ライセンス・プールからの充当

この方法では、ユーザがプリペイドのライセンス日数をソフトウェア・パブリッシャから購入します。その後ユーザは、全ライセンス日数の一部を使用して、特定のマシンにノードロックされたライセンスを所定の借用期間にわたって充当できます。たとえば、ユーザは出張の準備中（あるいは出張中）に、ノードロックされた有効期限 5 日のライセンスを自分のラップトップに充当します。それぞれの充当は異なるシステムにノードロックできます（あるいは同じマシンに数回ノードロックすることもできます）。これにより、プリペイドのライセンス日数の範囲内でライセンスをモバイル使用できます。

充当が行われるたびにライセンス日数が減少するため、このモデルは従量課金と似ています。他の従量課金モデルと異なる点は、特定のマシンにノードロックされると、ライセンスの期限が終了するまでアプリケーションを無制限に使用できることです。この短期ライセンスは早期返却できません。一度充当したライセンス日数を払い戻すこともできません。アプリケーション使用回数をベースとする従量課金モデルもあります。

9

複数のソフトウェア・パブリッシャからのライセンスの管理

複数のソフトウェア・パブリッシャのライセンスを管理する必要がある場合があります。

複数ライセンス管理方法の概要

複数のソフトウェア・パブリッシャからの FLEXenabled アプリケーションを実行するときには、インストール中の競合を防ぐ対策を講じる必要があります。これを実現する方法は複数ありますが、そのうちの3つを次に示します。

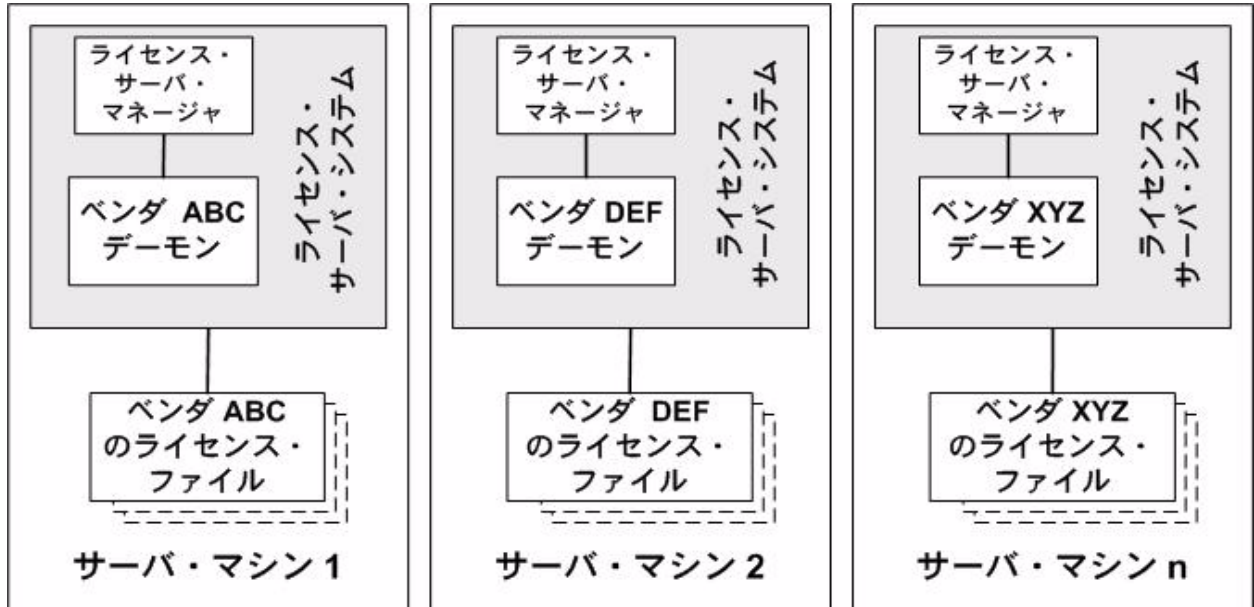
- 複数のマシンのそれぞれで、1つのライセンス・サーバ・マネージャと1つのベンダ・デーモンを実行し、1つのライセンス・ファイルを使用する。
- 1台のマシンで複数のライセンス・サーバ・マネージャを実行する。それぞれが1つのベンダ・デーモンを実行し、1つのライセンス・ファイルを使用する。
- 1台のマシンで1つのライセンス・サーバ・マネージャと複数のベンダ・デーモンを実行し、各ベンダ・デーモンが専用のライセンス・ファイルを使用する。ライセンス・ファイルは、共通のディレクトリを使用する。

この3つの方法のそれぞれについて、以降の各項で詳しく説明します。バリエーションについては、[その他の注意事項](#)で説明します。

複数のマシン

このシナリオでは、ライセンス・サーバの各インスタンス (ladmin または lmgrd、ベンダ・デーモン、ライセンス・ファイル、およびその他のファイル) が別々のマシンに配置されます。各マシンは、そのベンダ・デーモンに対応するライセンスのみを処理し、ローカルにコピーされたライセンス・サーバ・マネージャを実行します。図 9-1 に、この配置を示します。

図 9-1: 複数のライセンス・サーバ・システム



メリット

- 各ソフトウェア・パブリッシャのライセンス・ファイルは相互の依存関係がありません。
- 各マシンは独立して維持されます。マシンのいずれかがダウンした場合も、他のマシンは引き続きソフトウェア・パブリッシャのライセンス処理を続行します。
- 各サーバに専用のデバッグ・ログがあります。
- ライセンス要求は分散されます。

デメリット

- 管理コストが最も高くなります。

ライセンス・サーバの起動

次の例は、ライセンス・サーバ・マネージャとして `lmgrd` を使用します。



Task **ライセンス・サーバを起動するには、次の手順に従います。**

- 各マシンでライセンス・サーバ・マネージャを起動します。

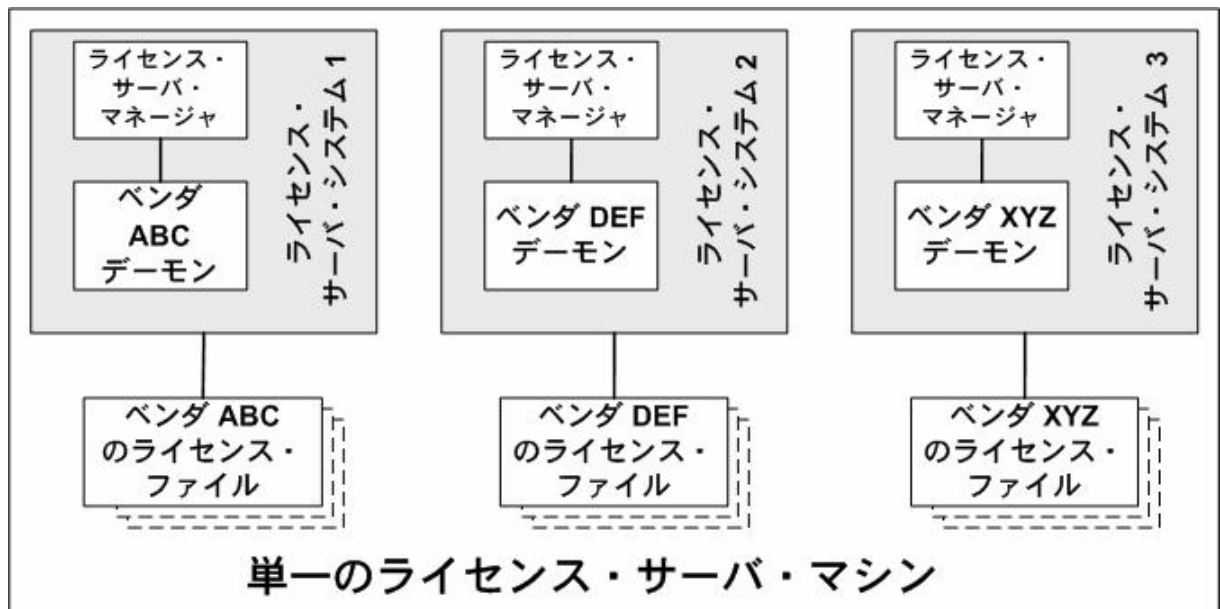
```
lmgrd -c server_system_n_license_list
```

`server_system_n_license_list` は、[複数のライセンス・ファイルの管理](#)で説明したライセンス・ファイルのリストです。各 `lmgrd` によって、そのライセンス・ファイルで参照されているベンダ・デーモンが起動します。

単一マシンと複数のライセンス・サーバ・インスタンス

このモデルでは、各ベンダ・デーモンおよび対応するライセンス・ファイルの処理が専用のライセンス・サーバ・マネージャによって実行され、すべてが 1 台のマシン上に含まれています。[図 9-2](#) にこの方式を示します。

図 9-2: 複数のライセンス・サーバ・マネージャ、複数のライセンス・ファイル



個々のライセンス・サーバを同じマシン上で管理する場合は、次の点に注意してください。

- TCP/IP ポート番号が SERVER 行で指定されている場合は、ライセンス・サーバのインスタンスごとに異なる番号を指定する必要があります。各ライセンス・ファイル内の TCP/IP ポート番号が他と重複しないように、標準的なテキスト・エディタを使用してポート番号を変更してください。実行しているインスタンスが 10 以下の場合、ポート番号をすべて省略することもできます。この場合は、デフォルトの 27000 ~ 27009 の範囲のユニークな番号が `lmadmin` または `lmgrd` によって自動選択されます。
- 個々のライセンス・ファイルに適合するバージョンの `lmadmin` または `lmgrd` を必ず使用してください。これは明示的なパスを使用して行います。「バージョン・コンポーネントの互換性」を参照してください。
- ライセンス・サーバのインスタンス数を制限する要件は、マシンの CPU、使用可能なメモリ、およびネットワークのみです。

メリット

- 各ソフトウェア・パブリッシャのライセンス・ファイルは相互の依存関係がありません。
- ライセンス・サーバは独立して維持されます。サーバのいずれかがダウンした場合も、他のサーバは引き続きそのライセンス処理を続行します。
- 各サーバに専用のデバッグ・ログがあります。

デメリット

- 管理コストが高くなります。
- システムがダウンすると、すべてのライセンスが無効になります。
- ライセンス要求の負荷が 1 台のマシンに集中します。

ライセンス・サーバの起動

次の例は、ライセンス・サーバ・マネージャとして `lmgrd` を使用します。



Task ライセンス・サーバを起動するには、次の手順に従います。

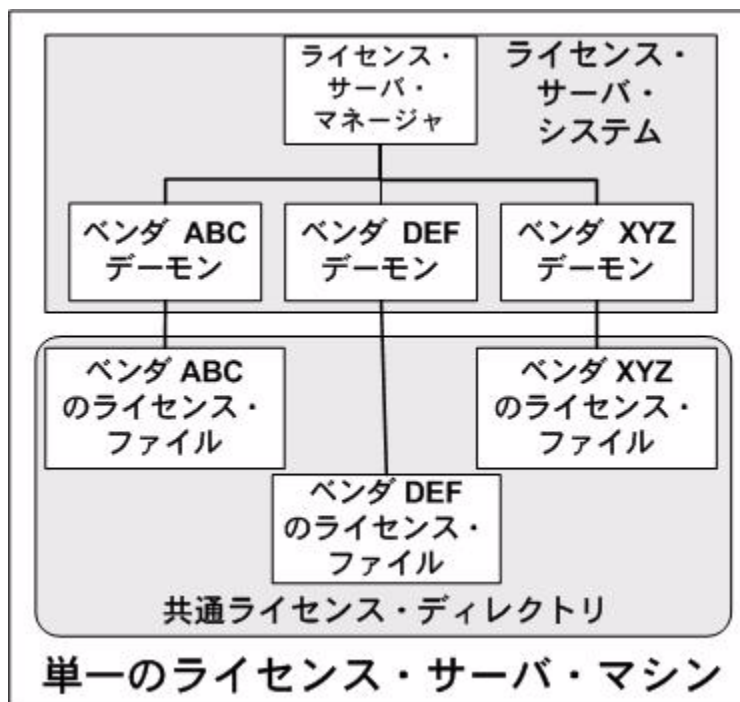
- 各ライセンス・サーバを次のように起動します。
 - a. サーバ 1 : `lmgrd -c vendor_ABC_license_dir_list`
 - b. サーバ 2 : `lmgrd -c vendor_DEF_license_dir_list`
 - c. サーバ 3 : `lmgrd -c vendor_XYZ_license_dir_list`

`vendor_nnn_license_list` は、[複数のライセンス・ファイルの管理](#)で説明したライセンス・ファイルのリストです。各 `lmgrd` によって、そのライセンス・ファイルで参照されているベンダ・デーモンが起動します。

単一マシンと単一ライセンス・サーバおよび複数ライセンス・ファイル

このシナリオでは、1台のマシンで1つのライセンス・サーバ・マネージャが実行され、それぞれに1つ以上のライセンス・ファイルがある1つまたは複数のベンダ・デーモンを処理します。ladmin を使用している場合は、異なるベンダのライセンス・ファイルを別々のディレクトリで管理できます。lmgd を使用している場合は、通常すべてのライセンス・ファイルが同じディレクトリに保管されます。ライセンス・ファイルの標準ファイル名拡張子は .lic です。ベンダ・デーモンの数は FLEXnet Publisher ライセンス・ツールキットの制限を受けません。図 9-3 に、このシナリオを示します。

図 9-3: 1つのライセンス・サーバ・マネージャ、複数のライセンス・ファイル



メリット

- 各ライセンス・ファイルを独立して維持できます。
- 管理コストが低くなります。

デメリット

- 1つのライセンス・サーバ・マネージャですべてのベンダ・デーモンを処理します。ライセンス・サーバ・マネージャがダウンした場合は、すべてのライセンスが使用不可になります。
- マシンがダウンすると、すべてのライセンスが使用不可になります。

- ベンダ・デーモンからの出力はすべて 1 つの共通デバッグ・ログに出力されます。ただし、各ベンダ・デーモンのオプション・ファイル内の `DEBUGLOG` で別のデバッグ・ログが指定されている場合を除きます。1 つの共通デバッグ・ログを使用していると、1 つのベンダ・デーモンに限られた問題のデバッグが困難になります。
- 1 台のマシンと 1 つのライセンス・サーバ・マネージャへのライセンスの負荷が最大になります。

ライセンス・サーバの起動

次の例は、ライセンス・サーバ・マネージャとして `lmgrd` を使用します。



Task **ライセンス・サーバを起動するには、次の手順に従います。**

- マシン上で、ライセンス・サーバ・マネージャを 1 回起動します。

```
lmgrd -c common_license_directory
```

`lmgrd` は、`common_license_directory` 内の `.lic` 拡張子のファイルをすべて処理し、それらのファイルで指定されているベンダ・デーモンをすべて起動します。したがって、個々のライセンス・ファイル名を `lmgrd` のコマンド・ラインで列挙する必要はありません。

参照

[複数のライセンス・ファイルの管理](#)
[特定のベンダ・デーモンのデバッグ・ログ出力の取得](#)

複数のライセンス・ファイルの管理

ライセンス・サーバ・マネージャとして `lmgrd` を使用すると、同じマシン上にある複数のライセンス・ファイルを 1 つのライセンス検索パスで管理できます。ライセンスの検索パスは、次の 2 とおりの方法で指定します。

- `lmgrd` の `-c` オプションを使用する。

```
lmgrd -c license_file_list [other lmgrd options]
```
- `lmgrd` プロセスの環境の有効範囲内で `LM_LICENSE_FILE` 環境変数を定義する。

マシン上の適切な場所にライセンス・ファイルをインストールしてから、`license_file_list` を定義します。

`license_file_list` は、次に示す要素のうち 1 つ以上から構成されるリストとして指定します。

- ライセンス・ファイルのフル・パス
- `.lic` 拡張子を持つライセンス・ファイルが格納されているディレクトリ
- `port@host` の設定。 `port` と `host` は、ライセンス・ファイルの `SERVER` 行で指定されている TCP/IP ポート番号とホスト名です。ライセンス・ファイルの `SERVER` 行で、デフォルトの TCP/IP ポートまたはデフォルトのポート範囲内のポート (27000 ~ 27009) が指定されている場合は、ショートカット形式の `@host` を使用することも可能です。

- 3 サーバ冗長化として構成されたライセンス・サーバであることを示す、カンマで区切られた 3 つの `port@host` 識別子のリスト。次に例を示します。

```
port1@host1,port2@host2,port3@host3
```



注意：ライセンス・ファイル名を区切るとき、UNIX ではコロン (:) を使用し、Windows ではセミコロン (;) を使用します。

lmgrd の起動時に、指定した順に各エントリが解析され、内部的にライセンスの検索パスが構築されます。

ライセンスの検索パスを使用するシナリオについては、「[複数のマシン、単一マシンと複数のライセンス・サーバ・インスタンス](#)」、「[単一マシンと単一ライセンス・サーバおよび複数ライセンス・ファイル](#)」を参照してください。

ライセンス・サーバ・マネージャとして lmadm を使用する場合は、各ベンダ・デーモンのライセンス・ファイルのリストを GUI で指定します。[Import License File] 画面 ([Vendor Daemon Configuration] 画面内からアクセス) でライセンス・ファイルを指定します。詳細については、lmadm のオンライン・ヘルプを参照してください。

参照

[環境変数を使用したライセンス検索パスの設定](#)
[3 サーバ冗長化の使用](#)
[環境変数](#)

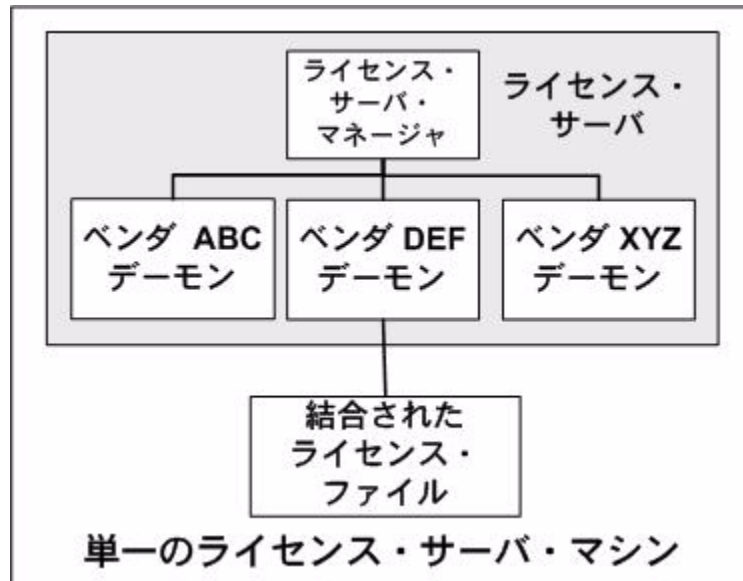
その他の注意事項

ライセンス・ファイルの結合

複数の製品があり、どちらのライセンスも同じマシンで使用されるように **SERVER** 行で指定されている場合は、これらのライセンス・ファイルに結合して 1 つのライセンス・ファイルにすることもできます。これは、ライセンス・サーバ・マネージャとして lmgrd を使用している場合に、好都合です。ライセンス・サーバ・マネージャとして lmadm を使用している場合は、ライセンス・ファイルを結合する必要はありません。複数のライセンス・ファイルを使用している場合、lmadm は各ライセンス・ファイルをインポートして lmadm を起動します。これにより、インポートされたライセンス・ファイルで定義されているベンダ・デーモンのそれぞれを起動します。

単一マシンと複数のライセンス・サーバ・インスタンスと単一マシンと単一ライセンス・サーバおよび複数ライセンス・ファイルで説明しているモデルのライセンス・ファイルは、一定の条件を満たしていれば結合が可能です。「[ライセンス・ファイル結合の条件](#)」を参照してください。図 9-4 に、結合したライセンス・ファイルを使用したシナリオを示します。

図 9-4: 1 つの lmgrd、1 つのライセンス・ファイル



メリット

- 管理するライセンス・ファイルは 1 つだけです。
- ファイルの結合後は、管理コストが低くなります。

デメリット

- 複数のソフトウェア・パブリッシャからのライセンス・ファイルを 1 つに結合する際には、開始時と開始後の両方について綿密な計画を立てる必要があります。

ライセンス・サーバの起動



Task ライセンス・サーバを起動するには、次の手順に従います。

- マシン上で、ライセンス・サーバ・マネージャを 1 回起動します。

```
lmgrd -c combined_license_file
```


ライセンス・ファイル結合の条件

製品のライセンス・ファイルは、SERVER 行のホスト名とホスト ID によってライセンス・サーバのマシンが定義されています。結合されるライセンス・ファイルは、以下の条件を満たしている必要があります。

- 各ファイル内の SERVER 行の数が同一である。
- ファイルの各 SERVER 行のホスト ID フィールドが、もう一つのファイルの各 SERVER 行のホスト ID フィールドと完全に一致している。

ライセンス・ファイルを結合できない理由として、以下が考えられます。

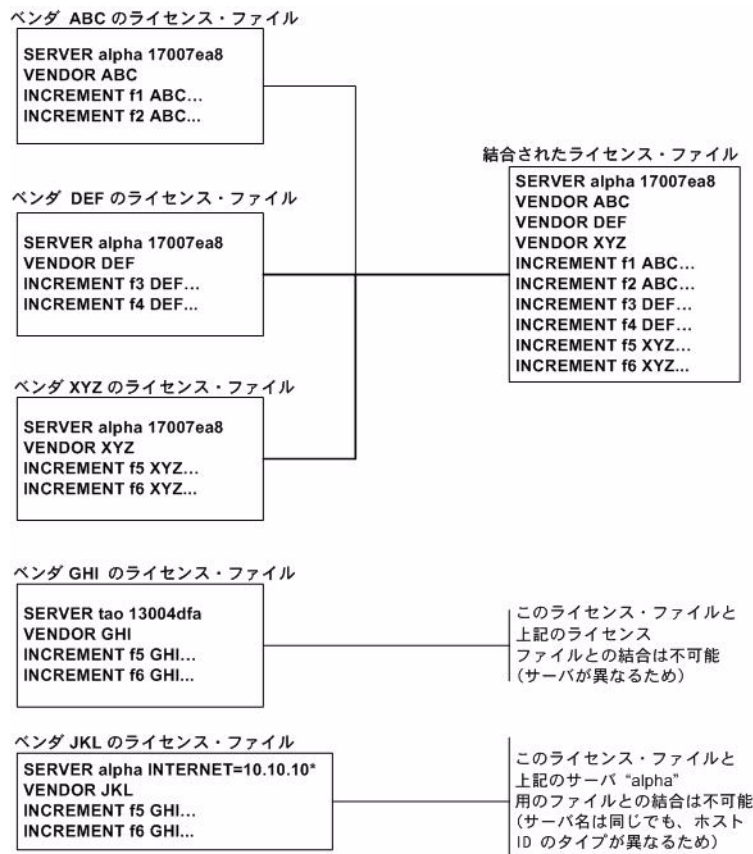
- ライセンス・ファイルは異なるサーバ・マシン上で実行するように設定されているため、ホスト ID が異なる。
- 1 つのファイルは単一ライセンス・サーバ用に設定されているが (SERVER 行が 1 行のみ)、他のファイルは 3 サーバ冗長性用に設定されている (SERVER 行が 3 行ある)。
- 同じシステムのホスト ID のホスト ID タイプがそれぞれ異なる。たとえば、あるライセンス・ファイルでは SERVER 行のホスト ID タイプとして INTERNET= が使用され、他のファイルでは Ethernet MAC アドレスがホスト ID タイプとして使用されている場合です。

ライセンス・ファイルが上記の条件を満たしていれば、[図 9-4](#) と後述の [ライセンス・ファイルの結合方法](#) で説明する手順に従ってファイルを結合できます。ただし、互換性のあるライセンス・ファイルの結合は必須ではありません。ファイルを結合しなくても、パフォーマンスやシステム負荷に悪影響が及ぶことはありません。

ライセンス・ファイルの結合方法

ライセンス・ファイルが結合の条件を満たしている場合は、任意のテキスト・エディタを使用してファイルを結合します。ライセンス・ファイルを結合するには、結合可能なすべてのライセンス・ファイルを 1 つのファイルに読み込みます。次に、不要な SERVER 行を削除して、SERVER 行のセットが 1 つだけ残るようにします。結果のデータを保存すると、結合されたライセンス・ファイルが完成します。図 9-5 に、ライセンス・ファイルの結合例を示します。

図 9-5: ライセンス・ファイルの結合



バージョン・コンポーネントの互換性

1 つの lmgrd プロセスで複数のベンダ・デーモンを管理するとき、これらのベンダ・デーモンで使用されている FLEXnet Publisher ライセンス・ツールキットのバージョンが異なる場合もあります。FLEXnet Publisher ライセンス・ツールキットのバージョン互換性規則については [コンポーネント間のバージョンの互換性](#) で説明していますが、この説明のとおり、FLEXnet Publisher ライセンス・ツールキットのコンポーネントはすべて互換性があります。

個々の FLEXenabled アプリケーションに対しては、複数のバージョンを管理できます。アプリケーションに対応するベンダ・デーモンは、FLEXenabled アプリケーションで使用される FLEXnet ライセンス・バージョンと最低限同じバージョンでなければなりません。

10

サポートされるプラットフォームの hostid

FLEXnet Publisher ライセンス・ツールキットでは、各システム・アーキテクチャに対して *hostid* という異なるシステム識別子を使用します。たとえば、すべての Sun Microsystems マシンには一意の *hostid* があります。このような理由から、一部のマシン・アーキテクチャではイーサネット・アドレスが *hostid* として使用されます。イーサネット・アドレスが 6 バイトの場合、各バイトは 2 桁の 16 進数として指定されます。イーサネット・アドレスを *hostid* として使用する場合は、12 桁の 16 進数をすべて指定します。たとえば、イーサネット・アドレスが “8:0:20:0:5:ac” の場合は、ホスト ID として “0800200005ac” と指定します。

hostid の形式

数値の 32 ビット *hostid* は通常 16 進形式で使用されます。一部のシステムでは、システム・コマンドによって 10 進形式の ID が返されます。10 進数であることを示す場合は、*hostid* の前に # 文字を付けます。たとえば、システム・コマンドが **2005771344** を返す場合、FLEXnet Publisher ライセンス・ツールキットが受け入れるようにするには **#2005771344** とします。あるいは、10 進数値を 16 進数に変換します。

マシンの *hostid* の取得

`Imhostid` ユーティリティでは、FLEXnet Licensing が特定のマシンで要求する正確な *hostid* が出力されます。*hostid* に ASCII の A ~ Z、a ~ z、または 0 ~ 9 以外の文字が含まれる場合、`-utf8` オプションを使用して `Imhostid` を指定します。*hostid* を正しく表示するには、UTF-8 エンコード文字列を表示できるメモ帳などのユーティリティを使用します。

次の表に、各マシン・アーキテクチャに必要なホスト ID を取得するための他の方法を示します。FLEXnet Publisher ライセンス・ツールキットは、特殊な **hostid** およびベンダ定義の **hostid** もサポートします。

表 10-1: 代替 Hostid 取得方法

ハードウェア・プラットフォーム	hostid	ライセンス・サーバ・システムで入力するコマンド	例
AIX (RS/6000、PPC)	32 ビット hostid	uname -m (000276513100 が返される)。最初と最後の 2 桁を削除し、残りの 8 桁を使用する。	02765131
HP (32 ビット版および 64 ビット版プラットフォーム、非 Itanium)	32 ビット hostid	uname -i。16 進に変換するか、# を前に付ける。	778DA450 または #2005771344
HP (64 ビット版 Itanium)	machine ID	getconf ¥ CS_PARTITION_IDENT then prefix with "ID_STRING="	ID_STRING=9c766319-db72-d411-af62-0060b05e4c05
Mac OS X	イーサネット・アドレス	/sbin/ifconfig eth0。ether の値からコロンを削除する。	000A277EA17E
	FLEXnet ID ドングル USB ポート・ドングル	lmhostid -flexid	FLEXID=9-b28520b9
Linux	イーサネット・アドレス	/sbin/ifconfig eth0。HWaddr からコロンを削除する。	00400516E525
	FLEXnet ID ドングル USB ポート・ドングル	lmhostid -flexid	FLEXID=9-b28520b9
SGI	32 ビット hostid	/etc/sysinfo -s。16 進数に変換するか、# のプレフィックスを付ける。	69064C3C または #1762020412
SUN	32 ビット hostid	hostid	170a3472
	イーサネット・アドレス	lmhostid -ether	00400516E525

表 10-1: 代替 Hostid 取得方法

ハードウェア・プラットフォーム	hostid	ライセンス・サーバ・システムで 入力するコマンド	例
Windows	イーサネット・アドレス	lmhostid	00B0A9DF9A32
	ディスクのシリアル 番号	DIR C: ([Volume Serial Number is] を探し、- を削除する)	DISK_SERIAL_NUM= 3e2e17fd
	FLEXnet ID ドングル パ ラレルまたは USB ポー ト・ドングル	lmhostid -flexid FLEXnet ID ドングル は、ソフトウェア・パブリッシャが利用可能にする。ソフトウェア・パブリッシャは、すべての FLEXnet ID ドングルのドライバをインストールするインストーラも提供できる。 パラレルの FLEXnet ID ドングルの場合、パラレル・ポートを双方向モードに設定する必要がある。	FLEXID=8-b28520b9

特殊な hostid

FLEXnet Publisher ライセンス・ツールキットには、すべてのプラットフォームに適用される特殊な **hostid** が多数含まれています。これらの **hostid** タイプは、**hostid** が必要な **SERVER** 行と **FEATURE** 行の両方で使用できます。具体的には、次のサブクエリです。

表 10-2: 特殊な **hostid** タイプ

hostid	説明
ANY	ソフトウェアを任意のシステムにロックする (つまり、何もロックしない)。
DEMO	ANY と同様だが、非カウントの FEATURE 行でのみ使用される。
COMPOSITE= <i>composite_hostid</i>	ソフトウェアを複合 hostid にロックする。複合 hostid は、1 つ以上の単純な hostid タイプの値をソフトウェア・パブリッシャの定義に従って組み合わせて作成されたハッシュ 12 文字の 16 進数値。複合 hostid は、 lmhostid 、 lmtools 、または lmadmin によって返されない。複合 hostid が使用されるときは、ソフトウェア・パブリッシャは、パブリッシャの複合 hostid を決定するユーティリティを提供する。システムによっては、複数の複合 hostid が提供され、ソフトウェアがロックされているシステムの識別に使用される場合もある。
DISPLAY= <i>display</i>	ソフトウェアを display にロックする。UNIX では、 display は /dev/ttyxx (アプリケーションがバックグラウンドで実行される場合は常に /dev/tty) または X-Display 名になる。Windows では、システム名か、(ターミナル・サーバ環境の場合は)ターミナル・サーバ・クライアント名になる (バージョン 8 以降の FLEXenabled アプリケーションのみ)。
HOSTNAME= <i>host</i>	ソフトウェアをコンピュータのホスト名 <i>host</i> にロックする。

表 10-2: 特殊な *hostid* タイプ

hostid	説明
ID=n	機能上は “ANY” <i>hostid</i> と同じで、どのシステムでも実行される。違いは、ライセンスが一意であり各ユーザを識別するために使用されるという点である。この <i>hostid</i> を使用してライセンス・サーバ (SERVER 行) または FLEXenabled アプリケーション (機能定義行) をロックする。番号にハイフンを含めて読みやすくすることができる。この場合、ハイフンは無視される。 例： <ul style="list-style-type: none">• ID=12345678• ID=1234-5678• ID=1-2-3-4-5-6-7-8 はすべて同じ
INTERNET=###.###.###.###	ソフトウェアをインターネットの IP アドレス、または IP アドレスのグループにロックする。ワイルドカードを使用できる。たとえば、198.156.*.* は、インターネット IP アドレスに一致するすべてのホストを意味する。主に、地域を表すサブネット別に使用アクセスを制限するために使用される。この場合は、機能定義行で <i>hostid</i> ロックとして使用される。
USER=user	ソフトウェアを指定したユーザ名 (<i>user</i>) にロックする。

例

```
FEATURE f1 demo 1.0 1-jan-2008 uncounted ¥  
HOSTID=FLEXID=6-a6300015f SIGN="<...>"
```

または

```
FEATURE f1 demo 1.0 1-jan-2008 uncounted ¥  
HOSTID=INTERNET=10.10.10.* SIGN="<...>"
```

トラブルシューティング

この項では、ライセンス・サーバについてこれまでにお客様から問い合わせがあった問題点を記載します。

一般的なトラブルシューティングのヒント

このリストは一般的なデバッグ情報を提供します。

- ライセンス・サーバを起動するときには、参照可能なローカル・ログ・ファイルへの出力を指定する。ログ・ファイルには有用な情報が数多く含まれている。問題が発生した場合は、ログ・ファイルを調べて、サポート担当者にお問い合わせの際に質問に回答できるよう準備する。
- ライセンス・サーバが正しく起動した場合は（ログ・ファイルから判断可能）、`lmstat -a` および `lmdiag` を実行して、そのプログラムでアプリケーションと同じ問題が発生するかどうかを確認する。
- アプリケーションがバージョン 4.1 以降（Windows ではバージョン 5 以降）の場合は、`FLEXLM_DIAGNOSTICS` 環境変数を使用できる。`FLEXLM_DIAGNOSTICS` を 1、2、3 のいずれかに設定する。3 に設定すると 2 よりも詳細な情報が示され、2 を指定すると 1 よりも詳細な情報が示される（特に、拒否された機能名）。詳細については、「[FLEXLM_DIAGNOSTICS](#)」を参照。
- サポート担当者にお問い合わせの場合は、以下の情報を事前に準備する。
 - ライセンス・サーバを実行しているマシンの種類
 - オペレーティング・システムのバージョン
 - アプリケーションを実行しているマシンとオペレーティング・システム
 - `FLEXenabled` アプリケーションで使用している `FLEXnet Publisher` ライセンス・ツールキットのバージョン

`lmver` スクリプトを使用する。または UNIX の場合は、ライセンス・サーバ・マネージャ、ベンダ・デーモン、およびアプリケーションで次のコマンドを実行する。

```
strings binary_name | grep Copy
```

あるいは、lmadmin の場合はコマンド `lmadmin -version`、lmgrd とベンダ・デーモンの場合は `-v` 引数を使用する (たとえば `lmgrd -v`)。

- ログ・ファイルに記録されているエラー・メッセージや警告メッセージ
- サーバが正しく起動したかどうか。次のようなメッセージがあるか確認する
`server xyz started for: feature1 feature2.`
- `lmstat -a` を実行して生成された出力
- 他の FLEXenabled 製品を実行しているかどうか
- 結合したライセンス・ファイルを使用しているか個別のライセンス・ファイルを使用しているか
- 3 台のサーバで構成される冗長ライセンス・サーバを使用しているかどうか (ライセンス・ファイルに複数の SERVER 行があるか)

FLEXLM_DIAGNOSTICS



注意 : FLEXnet Publisher ライセンス・ツールキットで診断出力を生成する機能は、ソフトウェア・パブリッシャが制御します。

FLEXLM_DIAGNOSTICS は、チェックアウトが拒否されたときにアプリケーションで診断情報を生成するための環境変数です。診断情報の形式は、将来変更される可能性があります。

UNIX では、診断出力は `stderr` に送られます。

Windows では、出力は `flexpid.log` (`pid` はアプリケーションのプロセス ID) という現在のディレクトリにあるファイルです。

レベル 1 の内容

FLEXLM_DIAGNOSTICS が 1 に設定されている場合は、標準の FLEXnet Publisher ライセンス・ツールキット・エラー・メッセージと、アプリケーションが使おうとしたライセンス・ファイルの完全なリストが表示されます。次に例を示します。

```
setenv FLEXLM_DIAGNOSTICS 1
FLEXnet checkout error: Cannot find license file (-1,73:2) No such file or directory
license file(s): /usr/myproduct/licenses/testing.lic license.lic
```


レベル 2 の内容

FLEXLM_DIAGNOSTICS が 2 に設定されている場合は、レベル 1 の出力に加えてチェックアウト引数が示されます。次に例を示します。

```
setenv FLEXLM_DIAGNOSTICS 2
FLEXnet checkout error: No such feature exists (-5,116:2) No such file or directory
license file(s): /usr/myproduct/licenses/testing.lic license.lic
lm_checkout("f1", 1.0, 1, 0x0, ..., 0x4000)
```

エラー・メッセージには実際には次の 2 つの問題が示され、どちらもチェックアウト中に発生しています。

- ライセンス内で該当する機能が見つからなかった。
- 他のライセンス・ファイルが見つからなかったこと。これによって、**No such file or directory** というメッセージが表示される。

以下に、**lm_checkout** 引数の説明を示します。

```
lm_checkout(feature, version, num_lic, queue_flag, ..., dupgroup_mask)
```

各パラメータの意味は、次のとおりです。

表 11-1: lm_checkout 引数

引数	説明
feature	要求された機能。
version	要求されたバージョン。ライセンス・ファイルには、要求されたバージョン以上のバージョンが含まれている必要がある。
num_lic	要求されたライセンス数。通常は 1。
queue_flag	0 に設定すると、キューイングはなし。 1 に設定した場合は、ライセンスのキュー（「ブロッキング」キュー）。 2 に設定した場合は、ライセンスのキューであるが、アプリケーションに返される（「非ブロッキング」キュー）。
dupgroup_mask	ライセンス共有とも呼ばれる重複グループを示す。ユーザ、ホスト、および表示は、 <code>lmstat -a</code> で示される。

レベル 3 の内容 (バージョン 6.0 以降のみ)

FLEXLM_DIAGNOSTICS が 3 に設定されている場合は、レベル 1 とレベル 2 の出力に加え、チェックアウトに成功した場合は、ライセンスの付与方法を説明する情報が出力されます。

```
setenv FLEXLM_DIAGNOSTICS 3
app
Checkout succeeded: f0/14263EAEA8E0
License file: ./servtest.lic
No server used
app2
Checkout succeeded: f1/BC64A7B120AE
License file: @localhost
License Server Machine: @localhost
app3
Checkout succeeded: f1/BC64A7B120AE
License file: servtest.lic
License Server Machine: @speedy
```

機能名とライセンス・キーに加え、ライセンス・ファイルの場所 (または `@host` が使用されている場合はホスト名) および該当する場合はサーバのホスト名が出力されます。

オプション・ファイルの管理

オプション・ファイルを使用すると、ライセンス管理者はライセンス・モデルの制約内でさまざまな処理パラメータを制御できます。ユーザの識別には、ユーザ名、ホスト変数、ディスプレイ、IP アドレス、または PROJECT (LM_PROJECT 環境変数で設定) が使用されます。

同時 (フローティング) ライセンスの場合、ライセンス管理者は以下のことができます。

- 機能の使用許可
- 機能の使用拒否
- ライセンスの予約

同時ライセンスはライセンス・ファイルか、信頼できる記憶領域の充当レコードに保管できます。

アクティブ化可能ライセンスについて、ライセンス管理者は以下のことができます。

- 特定の充当レコードのライセンス・アクティブ化の許可
- 特定の充当レコードのライセンス・アクティブ化の拒否

すべてのライセンスについて、ライセンス管理者は以下のことができます。

- 使用可能なライセンス数の制限
- ライセンス使用状況についてログに記録される情報の量の制御
- レポート・ログ・ファイルの有効化

ライセンス管理者はオプション・ファイルを使用して、ライセンス制限の強化や緩和を必要に応じて行うことができます。

オプション・ファイルの行は最大 2048 文字に制限されています。オプション・ファイルでは、行をつなげる文字として ¥ 文字を使用します。

- オプション・ファイルでの PROJECT による識別 (LM_PROJECT で設定) は、バージョン 7.0 のベンダ・デーモンで導入されました。

- 信頼できる記憶領域の充当レコードに保管されているライセンスのオプション・ファイル制御は、11.3 ベンダ・デーモンで導入されました。

オプション・ファイルの作成



Task オプション・ファイルを作成するには、次の手順に従います。

1. 任意のテキスト・エディタで、[オプション・ファイルの構文](#) で示す適切なオプションを使用してベンダ・デーモンのオプション・ファイルを作成します。
2. オプション・ファイルを保存します。保存場所は任意ですが、ライセンス・ファイルと同じディレクトリに保存することをおすすめします。
3. ライセンス・ファイルで、アプリケーションのベンダ・デーモンに対応する **VENDOR** 行の 4 つ目のフィールドとして、オプション・ファイルへのパスを追加します。次に例を示します。

```
VENDOR sampled /etc/sampled ¥  
    [options=]/sample_app/sampled/licenses/sampled.opt
```

上のように記述すると、指定したオプション・ファイルが sampled ベンダ・デーモンによって参照できます。

パスを省略した場合、ベンダ・デーモンは、以下の基準に従ってファイルを自動検索します。

- ファイル名が `vendor.opt` である (`vendor` はベンダ・デーモン名前)
- ライセンス・サーバ・マネージャが使用するライセンス・ファイルが入っているディレクトリ



注意: デフォルトのオプション・ファイル名 `vendor.opt` は、バージョン 6 のベンダ・デーモンで導入されました。

オプション・ファイルの構文

次に、オプション・ファイルの構文の概要を示します。例と詳細については、「[オプション・ファイルの例](#)」を参照してください。

ファイルの各行がそれぞれ 1 つのオプションを制御します。表 12-1 にオプション・キーワードの一覧を示します。

表 12-1: オプション・キーワード

オプション・キーワード	説明
BORROW_LOWWATER	借用できない BORROW ライセンスの数を設定する。
DEBUGLOG	指定したファイルにこのベンダ・デーモンのデバッグ・ログ情報を書き込む (バージョン 8.0 以降のベンダ・デーモン)。

表 12-1: オプション・キーワード (続き)

オプション・キーワード	説明
EXCLUDE	機能へのユーザ・アクセスを拒否する。
EXCLUDE_BORROW	ユーザが BORROW ライセンスを借用できないように拒否する。
EXCLUDE_ENTITLEMENT	信頼できる記憶領域の充当レコードに保管されているライセンスをユーザがアクティブ化できないようにする。
EXCLUDEALL	このベンダ・デーモンによって処理されるすべての機能へのユーザ・アクセスを拒否する。
FQDN_MATCHING	ホスト名マッチングのレベルを設定する。
GROUP	任意のオプションに対して使用するユーザのグループを定義する。
GROUPCASEINSENSITIVE	GROUP キーワードと HOST_GROUP キーワードで指定するユーザとホストのリストに、大文字と小文字の区別を設定する。
HOST_GROUP	任意のオプションに対して使用するホストのグループを定義する (バージョン 4.0 以降)。
INCLUDE	ユーザに機能の使用を許可する。
INCLUDE_BORROW	ユーザに BORROW ライセンスの借用を許可する。
INCLUDE_ENTITLEMENT	信頼できる記憶領域の充当レコードに保管されているライセンスをユーザがアクティブ化できるようにする。
INCLUDEALL	このベンダ・デーモンが処理するすべての機能の使用をユーザに許可する。
LINGER	特定の機能について、チェックイン期間を過ぎた後もユーザが残留間隔を延長できる。
MAX	特定の機能／グループの使用を制限し、使用に関してユーザ間の優先順位を付ける。
MAX_BORROW_HOURS	指定の機能の最長の借用期間を変更する。
MAX_OVERDRAFT	超過使用の量をライセンスで指定した量以下に制限する。
NOLOG	デバッグ・ログ・ファイルへの特定の項目のロギングをオフにする。
REPORTLOG	FLEXnet Manager ライセンス使用状況レポート・ツールによる使用に適したレポート・ログ・ファイルの書き込みを指定する。
RESERVE	特定のユーザまたはユーザ／ホストのグループのライセンスを予約する。
TIMEOUT	機能のアイドル・タイムアウトを指定し、フリー・プールに返却して他のユーザが使用できるようにする。
TIMEOUTALL	すべての機能のタイムアウトを設定する。

コメント

オプション・ファイルにコメントを記述するには、コメント行の先頭にシャープ記号 # を入力します。

機能の指定

オプション・ファイルのエントリ内で使用すると、オプション・キーワードと値のペアで機能名を完全修飾できます。この表記は、単一の機能に対して複数の **FEATURE** 行がある場合に、ライセンスの特定のグループを区別するために使用します。次の構文を使用します。

feature: keyword=value

次に例を示します。

f1:VERSION=2.0

上の表記では、機能 f1 について、バージョン 2.0 のライセンス・プールを指定しています。



注意：コロン (:) は、機能名の文字としても使用されます。機能名にコロンが含まれている場合は、引用符とスペースを使用した次の代替構文でライセンスのグループを指定します。

"feature keyword=value"

次に、ライセンスの特定のグループを表す機能名の変更子として使用されるオプション・キーワードを示します。

- VERSION=
- HOSTID=
- EXPDATE=
- KEY=
- SIGN=
- ISSUER=
- NOTICE=
- VENDOR_STRING= (パブリッシャによってプール・コンポーネントとして設定されている場合)
- dist_info=
- user_info=
- asset_info=

機能行に **USER_BASED** キーワードまたは **HOST_BASED** キーワードが使用されている場合は、この機能指定の構文を使用して機能を修飾する必要があります。

機能名の代わりにパッケージ名を指定すると、そのパッケージのすべてのコンポーネントにオプションが適用されます。

Type を使用したライセンス制限の指定

一部のオプション・キーワードは、ライセンスの使用者やライセンスの使用場所を制限します。これらのオプションには **type** 引数があり、制限の基準になるものを指定します。

オプション・キーワードの EXCLUDE、EXCLUDE_ENTITLEMENT、EXCLUDEALL、EXCLUDE_BORROW、INCLUDE、INCLUDE_ENTITLEMENT、INCLUDEALL、INCLUDE_BORROW、MAX、および RESERVE を使用する場合は、type に次の値を使用できます。

- **USER** – FLEXenabled アプリケーションを実行しているユーザのユーザ名。ユーザ名の大きい文字と小さい文字は区別される。
- **HOST** – アプリケーションを実行しているシステムのホスト名または IP アドレス。ホスト名の大きい文字と小さい文字は区別される。IP アドレスにはワイルド・カード文字を指定できる。

IP アドレスにはワイルド・カード文字を指定できる。

オプション・キーワードの EXCLUDE、EXCLUDEALL、EXCLUDE_BORROW、INCLUDE、INCLUDEALL、INCLUDE_BORROW、MAX、および RESERVE を使用する場合は、type に以下の値を使用できます。

- **DISPLAY** – アプリケーションが表示されるディスプレイ。UNIX では、DISPLAY は `/dev/ttyxx` (アプリケーションがバックグラウンドで実行されるときは常に `/dev/tty`) または X-Display 名になる。Windows では、システム名か、(ターミナル・サーバ環境の場合は)ターミナル・サーバ・クライアント名になる。表示名の大きい文字と小さい文字は区別される。
- **INTERNET** – アプリケーションを実行しているマシンの IP アドレス (IP アドレスにはワイルド・カード文字を使用できる)。
- **PROJECT** – FLEXenabled アプリケーションを実行しているユーザによって設定される LM_PROJECT 環境変数。プロジェクト名の大きい文字と小さい文字は区別される。

Windows (ターミナル・サーバを除く) では、HOST と DISPLAY はいずれもシステム名に設定します。ターミナル・サーバからのチェックアウトを許可するライセンス (機能行に TS_OK キーワードを指定) では、USER、HOST、および DISPLAY の名前がそれぞれ異なる場合があります。

上記の型は単一のメンバを取ります。次に例を示します。

```
EXCLUDE coolsoft USER joe
```

複数のユーザまたはホストのリストを指定するには、GROUP または HOST_GROUP オプション行を使用してリストを定義してから、GROUP 型または HOST_GROUP 型を使用してグループ名を指定します。次に例を示します。

```
GROUP stars joe barbara susan  
EXCLUDE coolsoft GROUP stars
```

- HOST の指定での IP アドレスの使用はバージョン 8 のベンダ・デーモンで導入されました。
- 機能名でのコロンの使用はバージョン 8 のベンダ・デーモンで導入されました。

BORROW_LOWWATER

このオプションは、ライセンス・ファイルに保管されているライセンスに使用されます。ライセンスが信頼できる記憶領域にある場合、通常は BORROW の代わりにアクティブ化が提供されます。

```
BORROW_LOWWATER feature[:keyword=value] n
```

BORROW 機能で借用できないライセンスの数を設定します。

表 12-2: BORROW_LOWWATER の用語

用語	説明
feature	対象となる機能の名前です。
keyword=value	ライセンスのグループを表す、機能名の変更子です。詳細については、「 機能の指定 」を参照してください。
n	ライセンス借用で借用できないライセンスの数です。

たとえば、機能 “f1” の数が 10 で、アプリケーションで借用が有効になっており、FEATURE 行が次のとおりだとします。

```
FEATURE f1 ... 10 ... BORROW SIGN=...
```

この場合、オプション・ファイルに次の行を指定すると、7つのライセンスのみを借用できます。

```
BORROW_LOWWATER f1 3
```

DEBUGLOG

```
DEBUGLOG [+]debug_log_path
```

このオプション・ファイルに関連付けられたベンダ・デーモンからのデバッグ・ログの出力先を指定します。debug_log_path の前に + 文字を指定した場合は、ログのエントリが追加され、それ以外の場合はデーモンの開始ごとにファイルが上書きされます。この設定が適用されるのは、このオプション・ファイルに関連付けられたベンダ・デーモンからの出力のみです。ladmin または lmgrd のデバッグ・ログ出力、および同じライセンス・ファイルのその他のベンダ・デーモンのデバッグ・ログ出力は、このファイルに出力されません。

Windows では、パス名にスペースが含まれる場合は、二重引用符で囲む必要があります。lmgrd がサービスとして起動される場合、レポート・ログ・ファイルのデフォルトの場所は c:¥winnt¥System32 フォルダです（完全修飾パスを指定する場合を除く）。

参照：

[ライセンス・サーバ・マネージャを Windows サービスとして設定する](#)

lmswitch

[デバッグ・ログ・ファイル](#) – 当該ベンダ・デーモンのファイルのみに制限されたデバッグ・ログの出力は、バージョン 8 のベンダ・デーモンで導入されました。

EXCLUDE

このオプションは、ライセンス・ファイルと信頼できる記憶領域に保管されている同時ライセンスに適用されます。

```
EXCLUDE feature[:keyword=value] type {name | group_name}
```

機能の使用を許可する対象のリストから、特定のユーザや事前に定義されたユーザ・グループなどを除外します。EXCLUDE は INCLUDE より優先度が上です。つまり、EXCLUDE のリストと INCLUDE のリストが競合する場合は、EXCLUDE が優先して解決されます。

表 12-3: EXCLUDE の用語

用語	説明
feature	対象となる機能やパッケージの名前です。
keyword=value	ライセンスのグループを表す、機能名の変更子です。詳細については、「 機能の指定 」を参照してください。
type	USER、HOST、DISPLAY、INTERNET、PROJECT、GROUP、または HOST_GROUP のいずれかです。詳細については、「 Type を使用したライセンス制限の指定 」を参照してください。
name	ライセンスの使用から除外する <i>type</i> の型の項目の名前です。
group_name	除外するグループの名前です。



Task 機能 **f1** を使用できるユーザのリストから、ユーザ **hank** を除外するには、次のように指定します。

```
EXCLUDE f1 USER hank
```

EXCLUDE_BORROW

このオプションは、ライセンス・ファイルに保管されているライセンスに使用されます。ライセンスが信頼できる記憶領域にある場合、通常は BORROW の代わりにアクティブ化が提供されます。

```
EXCLUDE_BORROW feature[:keyword=value] type ¥  
{name | group_name}
```

この BORROW 機能のライセンスの借用を許可する対象のリストから、特定のユーザや事前に定義されたユーザ・グループを除外します。EXCLUDE_BORROW は INCLUDE_BORROW より優先度が上です。つまり、EXCLUDE_BORROW のリストと INCLUDE_BORROW のリストが競合する場合は、EXCLUDE_BORROW が優先して解決されます。

表 12-4: EXCLUDE_BORROW の用語

用語	説明
<code>feature</code>	対象となる機能の名前です。
<code>keyword=value</code>	ライセンスのグループを表す、機能名の変更子です。詳細については、「 機能の指定 」を参照してください。
<code>type</code>	USER、HOST、DISPLAY、INTERNET、PROJECT、GROUP、または HOST_GROUP のいずれかです。詳細については、「 Type を使用したライセンス制限の指定 」を参照してください。
<code>name</code>	ライセンスの借用から除外する、 <code>type</code> の型の項目の名前です。
<code>group_name</code>	借用から除外するグループの名前です。



Task

機能 **f1** が **BORROW** 属性を持つと考えられる場合に、この機能を借用できるユーザのリストからユーザ **fred** を除外するには、次のように指定します。

```
EXCLUDE_BORROW f1 USER fred
```

EXCLUDE_ENTITLEMENT

このオプションは、信頼できる記憶領域に保管され、アクティブ化を使用して提供されるライセンスにのみ適用されます。

```
EXCLUDE_ENTITLEMENT entitlementId type {name | group_name}
```

信頼できる記憶領域に保管されている充当レコードに含まれたライセンスのアクティブ化を許可する対象のリストから、特定のユーザや事前に定義されたユーザ・グループなどを除外します。

EXCLUDE_ENTITLEMENT は **NCLUDE_ENTITLEMENT** より優先度が上です。つまり、**EXCLUDE_ENTITLEMENT** のリストと **INCLUDE_ENTITLEMENT** のリストが競合する場合は、**EXCLUDE_ENTITLEMENT** が優先して解決されます。

表 12-5: EXCLUDE_ENTITLEMENT の用語

用語	説明
<code>entitlementId</code>	ライセンスのアクティブ化要求時に使用される使用権利 Id です。
<code>type</code>	USER、HOST、GROUP、または HOST_GROUP のいずれかです。詳細については、「 Type を使用したライセンス制限の指定 」を参照してください。
<code>name</code>	ライセンスの使用から除外する <code>type</code> の型の項目の名前です。
<code>group_name</code>	除外するグループの名前です。



Important: 使用権利 ID “AB456” で指定された充当レコードで提供されるライセンスをアクティブ化できるユーザのリストからユーザ “pete” を除外するには、次のように指定します。

```
EXCLUDE_ENTITLEMENT AB456 USER pete
```

EXCLUDEALL

このオプションは、ライセンス・ファイルと信頼できる記憶領域に保管されている同時ライセンスに適用されます。

```
EXCLUDEALL type {name | group_name}
```

このベンダ・デーモンが処理するすべての機能を使用を許可する対象のリストから、特定のユーザまたは事前に定義されたユーザ・グループを除外します。

表 12-6: EXCLUDEALL の用語

用語	説明
type	USER、HOST、DISPLAY、INTERNET、PROJECT、GROUP、または HOST_GROUP のいずれかです。詳細については、「 Type を使用したライセンス制限の指定 」を参照してください。
name	ライセンスの使用から除外する <i>type</i> の型の項目の名前です。
group_name	除外するグループの名前です。

このベンダ・デーモンが処理するすべての機能を使用できる対象から、マシン **chaos** のすべてのユーザを除外するには、次のように指定します。

```
EXCLUDEALL HOST chaos
```

FQDN_MATCHING

このオプションは、ライセンス・ファイルまたは信頼できる記憶領域に保管されているすべてのライセンスに適用されます。

```
FQDN_MATCHING exact | lenient
```

HOST 型指定子で使用されたホスト名と、FLEXenabled アプリケーションによって送信されたホスト名との一致のレベルを設定します。アプリケーションからは、その設定に応じて、ホスト名または完全修飾ドメイン名 (FQDN) がベンダ・デーモンに送信され、HOST 型指定子に照らして検証されます。完全修飾ドメイン名のサポートの有無については、ソフトウェア・パブリッシャにお問い合わせください。

表 12-7: FQDN_MATCHING の用語

用語	説明
exact	HOST 型指定子のホスト名と、アプリケーションから送信されたホスト名は、内容と形式が一致する必要があります。デフォルトの設定です。
lenient	アプリケーションから送信されたホスト名は、HOST 型指定子またはアプリケーションで指定された範囲でのみ一致すればよいため、制限が緩和されます。

FQDN_MATCHING キーワードは、オプション・ファイル内で最後に指定されているもののみが適用され、それ以外はすべて無視されます。

表 12-8 に、オプション・ファイルの HOST 型指定子と、アプリケーションから送信されたホスト名とのマッチング処理の結果を示します。

表 12-8: ホスト名のマッチング表

		FQDN が設定されているアプリケーション – sends.myhost.abc.com	FQDN が設定されていないアプリケーション – sends.myhost.abc.com
FQDN_MATCHING exact	HOST myhost	×	○
	HOST myhost.abc.com	○	×
FQDN_MATCHING lenient	HOST myhost	○	○
	• オプション・ファイル HOST myhost.abc.com	○	○

例

次の例は、ホスト名の厳密な一致を示します。

```
INCLUDE f1 HOST myhost.abc.com
```

```
FQDN_MATCHING exact
```

これは、機能 f1 を使用できるホストのリストに myhost.abc.com を含めます。さらに、アプリケーションから送信されるホスト名は、myhost.abc.com に完全に一致する完全修飾ドメイン名である必要があります。

これに対し、次の例では、一致の条件が緩和されています。

```
INCLUDE f2 HOST myhost.abc.com
```

```
FQDN_MATCHING lenient
```

これは、機能 f2 を使用できるホストのリストに myhost.abc.com を含めます。送信されるホスト名が myhost.abc.com または単に myhost のみの場合は一致しますが、myhost.xyz.com、yourhost、yourhost.abc.com などとは一致しません。

次の例では、条件がさらに緩和されています。

```
INCLUDE f2 HOST myhost
FQDN_MATCHING lenient
```

これは、機能 f3 を使用できるホストのリストに myhost を含めます。lenient を指定しているため、myhost、myhost.abc.com、myhost.xyz.com などのホスト名は一致しますが、yourhost や yourhost.abc.com などは一致しません。

参照

[Type を使用したライセンス制限の指定](#)

FQDN_MATCHING はバージョン 9.3 のクライアント・ライブラリとベンダ・デーモンで導入されました。

GROUP

```
GROUP group_name user_list
```

INCLUDE、INCLUDEALL、INCLUDE_ENTITLEMENT、EXCLUDE、EXCLUDEALL、EXCLUDE_ENTITLEMENT、RESERVE の各オプション行で使用するユーザのグループを定義します。

表 12-9: GROUP の用語

用語	説明
group_name	定義するグループの名前です。グループ名の太文字と小文字は区別されます。
user_list	そのグループに属するユーザ名のリストです。名前の太文字と小文字は区別されます。太文字と小文字を区別しない場合は、オプション・ファイルの GROUPCASEINSENSITIVE キーワードを設定します。詳細については、「 GROUPCASEINSENSITIVE 」を参照してください。

同じグループ名に複数の GROUP 行を指定すると、指定したすべてのユーザがそのグループに追加されます。

bob、**howard**、**james** で構成されるグループ **Hackers** を定義するには、次のように指定します。

```
GROUP Hackers bob howard james
```



注意： USER_GROUP は GROUP のエイリアスです。

GROUPCASEINSENSITIVE

```
GROUPCASEINSENSITIVE OFF|ON
```

ON に設定した場合、オプション・ファイルの GROUP キーワードで指定するユーザ名と HOST_GROUP キーワードで指定するホスト名の太文字と小文字は区別されません。

デフォルトでは、GROUPCASEINSENSITIVE は OFF で、ユーザ名とホスト名の太文字と小文字は区別されます。

HOST_GROUP

```
HOST_GROUP group_name host_list
```

INCLUDE、INCLUDEALL、INCLUDE_ENTITLEMENT、EXCLUDE、EXCLUDEALL、EXCLUDE_ENTITLEMENT、RESERVE の各オプション行で使用するホストのグループを定義します。複数の HOST_GROUP 行を指定した場合、指定したホストすべてがグループに追加されます。

表 12-10: HOST_GROUP の用語

用語	定義
group_name	定義するグループの名前です。ホスト・グループ名では、大文字と小文字が区別されます。
host_list	そのグループのホスト名のリストです。名前の大文字と小文字は区別されます。大文字と小文字を区別しない場合は、オプション・ファイルの GROUPCASEINSENSITIVE キーワードを設定します。詳細については、「 GROUPCASEINSENSITIVE 」を参照してください。

tokyo、**seattle**、**auckland** で構成されるホスト・グループ **Pacific** を定義するには、次のように指定します。

```
HOST_GROUP Pacific tokyo seattle auckland
```

オプション・ファイル内のどの部分でも、ホスト名の代わりに IP アドレスを使用できます。

INCLUDE

このオプションは、ライセンス・ファイルと信頼できる記憶領域に保管されている同時ライセンスに適用されます。

```
INCLUDE feature[:keyword=value] type {name | group_name}
```

この機能のライセンスの使用を許可する対象のリストに、特定のユーザや事前に定義されたユーザ・グループを含めます。INCLUDE 文で指定されていない人は、その機能を使用できません。EXCLUDE は INCLUDE より優先度が上です。つまり、EXCLUDE のリストと INCLUDE のリストが競合する場合は、EXCLUDE が優先して解決されます。

表 12-11: INCLUDE の用語

用語	定義
feature	対象となる機能やパッケージの名前です。
keyword=value	ライセンスのグループを表す、機能名の変更子です。詳細については、「 機能の指定 」を参照してください。
type	USER、HOST、DISPLAY、INTERNET、PROJECT、GROUP、または HOST_GROUP のいずれかです。詳細については、「 Type を使用したライセンス制限の指定 」を参照してください。
name	ライセンスの使用に追加する type の型の項目の名前です。
group_name	ライセンスの使用に追加するグループの名前です。

機能 **f1** を使用できるユーザのリストにユーザ **bob** を追加するには、次のように指定します。

```
INCLUDE f1 USER bob
```



注意： *INCLUDE* は *USER_BASED* または *HOST_BASED* の機能に必要です。ライセンス管理者は、製品の使用を許可するユーザを *INCLUDE* で指定します。 *INCLUDE* で追加できるユーザの数はライセンスによって制限されます。

INCLUDE_BORROW

このオプションは、ライセンス・ファイルに保管されているライセンスに使用されます。ライセンスが信頼できる記憶領域にある場合、通常は *BORROW* の代わりにアクティブ化が提供されます。

```
INCLUDE_BORROW feature[:keyword=value] type {name | group_name}
```

BORROW 機能の借用を許可する対象のリストに、特定のユーザや事前に定義されたユーザ・グループを含めます。 *INCLUDE_BORROW* 文で指定されていないユーザは、ライセンスを借用できません。 *EXCLUDE_BORROW* は *INCLUDE_BORROW* より優先度が上です。つまり、 *EXCLUDE_BORROW* のリストと *INCLUDE_BORROW* のリストが競合する場合は、 *EXCLUDE_BORROW* が優先して解決されます。

表 12-12: INCLUDE_BORROW の用語

用語	定義
feature	対象となる機能の名前です。
keyword=value	ライセンスのグループを表す、機能名の変更子です。詳細については、「 機能の指定 」を参照してください。
type	USER、HOST、DISPLAY、INTERNET、PROJECT、GROUP、または HOST_GROUP のいずれかです。詳細については、「 Type を使用したライセンス制限の指定 」を参照してください。
name	ライセンスの借用に追加する type の型の項目の名前です。

表 12-12: INCLUDE_BORROW の用語

用語	定義
<code>group_name</code>	ライセンスの借用に追加するグループの名前です。

機能 **f1** を借用できるユーザのリストにユーザ **tom** を追加するには、次のように指定します。

```
INCLUDE_BORROW f1 USER tom
```

INCLUDE_ENTITLEMENT

このオプションは、信頼できる記憶領域に保管されているライセンスにのみ適用されます。

```
INCLUDE_ENTITLEMENT entitlementId type {name | group_name}
```

信頼できる記憶領域に保管されている充当レコードに含まれたライセンスのアクティブ化を許可する対象のリストに、特定のユーザや事前に定義されたユーザ・グループを含めます。

EXCLUDE_ENTITLEMENT は INCLUDE_ENTITLEMENT より優先度が上です。つまり、EXCLUDE_ENTITLEMENT のリストと INCLUDE_ENTITLEMENT のリストが競合する場合は、EXCLUDE_ENTITLEMENT が優先して解決されます。

表 12-13: INCLUDE_ENTITLEMENT の用語

用語	定義
<code>entitlementId</code>	ライセンスのアクティブ化要求時に最初に使用された使用権利 Id です。
<code>type</code>	USER、HOST、GROUP、または HOST_GROUP のいずれかです。詳細については、「 Type を使用したライセンス制限の指定 」を参照してください。
<code>name</code>	ライセンスの使用に追加する <code>type</code> の型の項目の名前です。
<code>group_name</code>	追加するグループの名前です。

使用権利 Id 「AB456」で指定された充当レコードで提供されるライセンスをアクティブ化できるユーザのリストにユーザ **claire** を含めるには、次のように指定します。

```
INCLUDE_ENTITLEMENT AB456 USER claire
```


INCLUDEALL

このオプションは、ライセンス・ファイルと信頼できる記憶領域に保管されている同時ライセンスに適用されます。

`INCLUDEALL type {name | group_name}`

このベンダ・デーモンが処理するすべての機能を使用できるユーザーのリストに、特定のユーザーや事前に定義されたユーザー・グループを含めます。INCLUDEALL 文で指定されていない人は、これらの機能を使用できません。

表 12-14: INCLUDEALL の用語

用語	定義
type	USER、HOST、DISPLAY、INTERNET、PROJECT、GROUP、または HOST_GROUP のいずれかです。詳細については、「 Type を使用したライセンス制限の指定 」を参照してください。
name	ライセンスの使用に追加する <i>type</i> の型の項目の名前です。
group_name	追加するグループの名前です。

ユーザー **jane** に対し、このベンダ・デーモンが処理するすべての機能を使用を許可するには、次のように指定します。

`INCLUDEALL USER jane`

LINGER

このオプションは、ライセンス・ファイルと信頼できる記憶領域に保管されている同時ライセンスに適用されます。

`LINGER feature[:keyword=value] seconds`

残留ライセンスは、ライセンスのチェックイン期間の経過後または **FLEXenabled** アプリケーションの終了後のいずれの場合も、指定した時間内はチェックアウトされた状態が継続されます。このオプションを使用すると、ソフトウェア・パブリッシャが **FLEXenabled** アプリケーションに設定したデフォルトの残留時間が延長されます。



注意：この機能が動作するためには、ソフトウェア・パブリッシャが **FLEXenabled** アプリケーションでこの機能を有効にしている必要があります。この機能が実装されているかどうかについては、ソフトウェア・パブリッシャにお問い合わせください。

表 12-15: LINGER の用語

用語	定義
feature	機能の名前です。
keyword=value	ライセンスのグループを表す、機能名の変更子です。詳細については、「 機能の指定 」を参照してください。
seconds	ライセンスが残留する秒数です。最小値はソフトウェア・パブリッシャが設定します。最小値より小さい値を <i>seconds</i> に指定した場合は、最小値が使用されます。

機能 *f1* の残留値を 1 時間 (3600 秒) に設定するには、次のように指定します。

```
LINGER f1 3600
```

ベンダ・デーモンは、すべての残留ライセンスを毎分 1 回のみチェックするため、実際の残留時間は多少異なります。ただし、新しいライセンスが要求され、残留ライセンスを使用する以外には拒否するか方法がない場合には、新しい要求を満たすために残留ライセンスのチェックが直ちに行われます。

MAX

このオプションは、ライセンス・ファイルと信頼できる記憶領域に保管されている同時ライセンスに適用されます。

```
MAX num_lic feature[:keyword=value] type {name | group_name}
```

グループまたはユーザに対し、使用制限を設定します。

表 12-16: MAX の用語

用語	説明
num_lic	このユーザまたはグループに対する使用制限です。
feature	この制限を適用する機能またはパッケージです。
keyword=value	ライセンスのグループを表す、機能名の変更子です。詳細については、「 機能の指定 」を参照してください。
type	USER、HOST、DISPLAY、INTERNET、PROJECT、GROUP、または HOST_GROUP のいずれかです。詳細については、「 Type を使用したライセンス制限の指定 」を参照してください。
name	使用を制限する <i>type</i> の型の項目の名前です。
group_name	制限するグループの名前です。

たとえば、機能 *f1* について、ユーザ *jan* を 5 ライセンスに制限するには、オプション・ファイルに次の行を指定します。

```
MAX 5 f1 USER jan
```

MAX_BORROW_HOURS

このオプションは、ライセンス・ファイルに保管されているライセンスに使用されます。ライセンスが信頼できる記憶領域にある場合、通常は BORROW の代わりにアクティブ化が提供されます。

MAX_BORROW_HOURS *feature[:keyword=value] num_hours*

ライセンスを借用できる最長期間を、*feature* のライセンス・ファイルで指定された期間から変更します。新しい期間はライセンス・ファイルで指定された期間より短くしてください。オプション・ファイルに複数の MAX_BORROW_HOURS キーワードを指定した場合、最後に指定したもののみが *feature* に適用されます。

表 12-17: MAX_BORROW_HOURS の用語

用語	説明
feature	この借用期間を適用する機能です。 <i>feature</i> で BORROW が有効になっている必要があります。
keyword=value	ライセンスのグループを表す、機能名の変更子です。詳細については、「 機能の指定 」を参照してください。
num_hours	新しい借用期間の時間数です。この値は、ライセンス・ファイルで指定されている値（指定がない場合のデフォルトは 168 時間）よりも短くしてください。

MAX_OVERDRAFT

このオプションは、ライセンス・ファイルと信頼できる記憶領域に保管されている同時ライセンスに適用されます。

MAX_OVERDRAFT *feature[:keyword=value] num_lic*

OVERDRAFT ライセンスの使用量を、ライセンス・ファイルで許可されている OVERDRAFT の量未満に制限します。

表 12-18: MAX_OVERDRAFT の用語

用語	説明
feature	この制限を適用する機能です。
keyword=value	ライセンスのグループを表す、機能名の変更子です。詳細については、「 機能の指定 」を参照してください。
num_lic	このユーザまたはグループに対する使用制限です。

NOLOG

NOLOG { IN | OUT | DENIED | QUEUED | UNSUPPORTED }

指定した種類のイベントについて、デバッグ・ログ・ファイルへのロギングを中止します。

表 12-19: NOLOG の用語

入力値	説明
NOLOG IN	チェックインのロギングをオフにします。チェックアウトとキューイングされた要求のロギングをオフにする場合は、2 つの NOLOG 行が必要です。
NOLOG DENIED NOLOG QUEUED	チェックアウトとキューイングされた要求のロギングをオフにします。ライセンス管理者は、デバッグ・ログ・ファイルのサイズを小さくする目的でこのオプションを使用します。ただし、ライセンス・サーバの問題をデバッグするときに、デバッグ・ログの有効性が低下する可能性があります。
NOLOG UNSUPPORTED	デバッグ・ログで “UNSUPPORTED” メッセージを抑制します。これにより、機能がサポートされていないために発生するエラーを報告するエラー・メッセージがデバッグ・ログに表示されなくなります。

参照

[lmswitch](#)

REPORTLOG

REPORTLOG [+]*report_log_path*

REPORTLOG では、このベンダ・デーモンのレポート・ログ・ファイルを指定します。*report_log_path* の前にプラス記号 (+ 文字) を指定して、ログのエントリを追加することをおすすめします。これを指定しない場合、デーモンの開始ごとにファイルが上書きされます。

Windows では、パス名にスペースが含まれる場合は二重引用符で囲む必要があります。lmgrd がサービスとして起動される場合、レポート・ログ・ファイルのデフォルトの場所は `c:\winnt\System32` フォルダです (完全修飾パスを指定する場合を除く)。



注意: レポート・ログ・ファイルの処理には、Acrecco ソフトウェア の別製品である *FLEXnet Manager* が使用されます。*FLEXnet Manager* が処理するのはレポート・ログ・ファイルのみで、デバッグ・ログ・ファイルは処理しません。

LM_PROJECT によるプロジェクトについてのレポート

FLEXnet Manager レポート・ライタがレポートする対象はプロジェクトです。プロジェクトは、同じプロジェクトに関与しているすべてのユーザの LM_PROJECT 環境変数（または Windows のレジストリ）に、プロジェクトを表す文字列に設定することによってセットアップされます。FLEXnet Manager は、アプリケーションの実行時の LM_PROJECT の設定による定義に従って、使用をプロジェクトごとに分類します。

参照

[ライセンス・サーバ・マネージャを Windows サービスとして設定する環境変数](#)
[レポート・ログ・ファイル](#)

RESERVE

このオプションは、ライセンス・ファイルと信頼できる記憶領域に保管されている同時ライセンスに適用されます。

```
RESERVE num_lic feature[:keyword=value] type {name | group_name}
```

特定のユーザのためにライセンスを予約します。

表 12-20: RESERVE の用語

用語	説明
num_lic	このユーザまたはグループに予約するライセンスの数です。
feature	この予約を適用する機能またはパッケージです。
keyword=value	ライセンスのグループを表す、機能名の変更子です。詳細については、「 機能の指定 」を参照してください。
type	USER、HOST、DISPLAY、INTERNET、PROJECT、GROUP、または HOST_GROUP のいずれかです。詳細については、「 Type を使用したライセンス制限の指定 」を参照してください。
name	ライセンスの使用を予約する <i>type</i> の型の項目の名前です。
group_name	ライセンスの使用を予約するグループの名前です。

ユーザ mel に対し、機能 f1 の 1 ライセンスを予約するには、次のように指定します。

```
RESERVE 1 f1 USER mel
```

複数のユーザまたはグループのそれぞれに対してライセンスを予約する場合は、ユーザまたはグループごとに RESERVE 行を指定する必要があります。パッケージ名を指定した場合、そのパッケージを構成するすべてのコンポーネントが予約されます。



注意：特定のユーザに対して予約したライセンスは、そのユーザ専用になります。そのユーザがライセンスをアクティブに使用していない場合でも、他のユーザは使用できません。しかし、RESERVE で予約したライセンスの使用状況は、実際に使用されていない場合でも、FLEXnet Manager によってレポートされません。

TIMEOUT

このオプションは、ライセンス・ファイルと信頼できる記憶領域に保管されている同時ライセンスに適用されます。

`TIMEOUT feature[:keyword=value] seconds`

アクティブでないライセンスがベンダ・デーモンによって解放されて再利用されるまでの時間を設定します。



注意：この機能が動作するためには、ソフトウェア・パブリッシャが FLEXenabled アプリケーションでこの機能を有効にしている必要があります。この機能が実装されているかどうかについては、ソフトウェア・パブリッシャにお問い合わせください。

表 12-21: TIMEOUT の用語

用語	説明
feature	機能の名前です。
keyword=value	ライセンスのグループを表す、機能名の変更子です。詳細については、「 機能の指定 」を参照してください。
seconds	アクティブでなくなったライセンスが再利用されるまでの秒数です。最小値はソフトウェア・パブリッシャが設定します。最小値より小さい値を <code>seconds</code> に指定した場合は、最小値が使用されます。

機能 f1 のタイムアウトを 1 時間 (3600 秒) に設定するには、次のように指定します。

`TIMEOUT f1 3600`

TIMEOUT では、FLEXenabled アプリケーションがアクティブでない時間が指定時間を超えた場合に、ライセンスがチェックインされます。デーモンは、プロセスのハートビートを受信しなくなると、そのプロセスはアクティブでなくなったことを宣言します。アクティブな FLEXenabled アプリケーションはハートビートを送信します。

この機能を利用するには、オプション・ファイルに TIMEOUT 行を指定する必要があります。

TIMEOUTALL

このオプションは、ライセンス・ファイルと信頼できる記憶領域に保管されている同時ライセンスに適用されます。

TIMEOUTALL *seconds*

TIMEOUT と同じですが、すべての機能に該当します。

ベンダ・デーモンによるオプション・ファイルの使用方法

ベンダ・デーモンは、`lmadmin` または `lmgrd` によって起動されると、そのオプション・ファイルを読み込みます。各ベンダ・デーモンに対応するオプション・ファイルは 1 つのみで、ベンダ・デーモンごとに個別のオプション・ファイルが必要です。オプション・ファイルの変更を有効にするためには、ベンダ・デーモンがそのオプション・ファイルを読み込む必要があります。`lmreread` ユーティリティを使用すると、ベンダ・デーモンはオプション・ファイルを再読み込みします。

- `lmreread` ユーティリティは、バージョン 8.0 のベンダ・デーモンで強化され、ベンダ・デーモンのオプション・ファイルを再読み込みできるようになりました。これより前のバージョンを使用している場合、オプション・ファイルを再読み込みするには、ベンダ・デーモンを停止して再起動する必要があります。

オプション・ファイル内での優先度の規則

1 つのオプション・ファイル内で `INCLUDE` 文と `EXCLUDE` 文を組み合わせ、ライセンス・ファイルまたは（信頼できる記憶領域の）充当レコードの同じ機能へのアクセスを制御する場合には、優先度の規則が適用されます。以下に、両方の種類の文を使用する場合の優先度の定義を示します。

- `EXCLUDE` リストのみを指定した場合は、そのリストに含まれていないすべての人が機能を使用できる。
- `INCLUDE` リストのみを指定した場合は、そのリストに含まれている人のみが機能を使用できる。
- どちらのリストも指定していない場合は、すべての人が機能を使用できる。
- `EXCLUDE` リストは `INCLUDE` リストより先に適用される。つまり、両方のリストに含まれている人は機能を使用できない。

`INCLUDE` リストまたは `EXCLUDE` リストを作成すると、それ以外のすべての人は暗黙的にそのグループに含まれていないとみなされます。これにより、ライセンス管理者はアクセスを許可または拒否する各ユーザをリストで明示的に指定しなくても、ライセンスを制御できます。つまり、次の 2 つの方法のいずれかを行えます。

- 多数のユーザにアクセス許可を与え、除外するユーザのみをリストに指定する。
- アクセスを厳しく制限し、アクセス許可のあるユーザだけをリストに含める。

オプション・ファイルの例

以下は、ライセンスへのアクセスを効果的に制御する方法を示すオプション・ファイルの例です。

簡単なオプション・ファイルの例

```
RESERVE 1 compile USER robert  
RESERVE 3 compile HOST mainline  
EXCLUDE compile USER lori  
NOLOG QUEUED
```

このオプション・ファイルは同時ライセンスの使用を次のように制限します。

- 機能 **compile** について、ユーザ **robert** に 1 つのライセンスを予約する。
- 機能 **compile** について、ホスト名 **mainline** のシステム上のすべてのユーザに 3 つのライセンスを予約する。
- ユーザ **lori** がネットワーク上のいずれのマシンでも **compile** 機能を使用できないようにする。
- デバッグ・ログ・ファイルで **QUEUED** メッセージを省略する。

予約するライセンスの合計数は、**FEATURE** 行で指定されたライセンス数以下であることが必要です。上の例では、**compile** の **FEATURE** 行に最低 4 つのライセンスが指定されている必要があります。使用できるライセンスの数がそれより少ない場合、最初の一連の予約（ライセンス数の制限値まで）が使用されます。

このデータが `/a/b/sampled/licenses/sampled.opt` ファイルにある場合は、ライセンス・ファイルの **VENDOR** 行を次のように変更します。

```
VENDOR sampled /etc/sampled /sample_app/sampled/licenses/sampled.opt
```

複数ユーザのアクセスの制限

INCLUDE、**INCLUDEALL**、**INCLUDE_BORROW**、**INCLUDE_ENTITLEMENT**、**EXCLUDE**、**EXCLUDEALL**、**EXCLUDE_BORROW**、**EXCLUDE_ENTITLEMENT**、**MAX**、**RESERVE** の各行に、単一のユーザ名（またはグループ）を指定する必要があります。複数のユーザ名を対象とする場合は、**GROUP** を使用してグループを作成します。たとえば、**bob**、**howard**、**james** が **toothbrush** という機能の使用から除外するには、オプション・ファイルを次のように作成します。

```
EXCLUDE toothbrush USER bob  
EXCLUDE toothbrush USER howard  
EXCLUDE toothbrush USER james
```

しかし、これより簡単な方法があります。**GROUP** を使用してグループを作成し、機能の使用対象からユーザのリストを除外する方法です。次のオプション・ファイルは、上の例と同様に、**bob**、**howard**、**james** を **toothbrush** という機能の使用から除外します。

```
# First define the group "Hackers"  
GROUP Hackers bob howard james  
# Then exclude the group  
EXCLUDE toothbrush GROUP Hackers
```

このようにすることで、そのグループに対していずれかの機能へのアクセスを許可または拒否するときに、エイリアスのリストを使用して簡単に指定できます。

複数のホストに対してライセンスを許可、拒否、または予約するには、**HOST_GROUP** を使用します。たとえば、ホスト **fred** と **barney** でログインしているすべてのユーザを、**f1** という機能の使用から除外するには、オプション・ファイルに下記の行を追加します。

```
HOST_GROUP writers fred barney
EXCLUDE f1 HOST_GROUP writers
```

参照

グループの定義の詳細については、「**HOST_GROUP**」を参照してください。

EXCLUDE の例

```
#First Define the group "painters"
GROUP painters picasso mondrian klee
EXCLUDE spell GROUP painters
EXCLUDE spell USER bob
EXCLUDE spell INTERNET 123.123.123.*
```

このオプション・ファイルでは、次のように指定しています。

- ユーザ **picasso**、**mondrian**、**klee** は、ネットワーク上のどのシステムでも機能 **spell** を使用できない。
- ユーザ **bob** は、ネットワーク上のいずれのマシンでも機能 **spell** を使用できない。
- IP アドレスが 123.123.123.0 ~ 123.123.123.255 の範囲のホストにログインしているユーザは、機能 **spell** を使用できない。
- その他のユーザは、除外する IP アドレスに含まれておらず、かつ **painters** グループのメンバーではなく、**bob** でない場合に限り、機能 **spell** を使用できる（暗黙的な指定）。

ここでは、**bob** を **painters** グループに追加するという方法もありました。しかし、今後 **painters** を他の目的で使用する可能性があるため、ライセンス管理者は **bob** を特例として処理する方法を選択しました。この場合、2つの EXCLUDE 文を合わせて、4人のユーザのリストが作成されます。

EXCLUDE_ENTITLEMENT の例

```
#First Define the group "admin"
GROUP admin johns adrianp maryt
EXCLUDE_ENTITLEMENT qf573k GROUP admin
EXCLUDE_ENTITLEMENT qf573k USER bob
EXCLUDE_ENTITLEMENT qf573k HOST cordelia
```

このオプション・ファイルでは、次のように指定しています。

- ユーザ **johns**、**adrianp**、**maryt** は、ネットワーク上のどのシステムでも、使用権利 Id **qf573k** を使用して取得した充当レコードに含まれているライセンスをアクティブ化できない。
- ユーザ **bob** は、ネットワーク上のどのシステムでも、使用権利 Id **qf573k** を使用して取得した充当レコードに含まれているライセンスをアクティブ化できない。
- **cordelia** というシステムのユーザは、使用権利 Id **qf573k** を使用して取得した充当レコードに含まれているライセンスをアクティブ化できない。
- 暗黙的には、**cordelia** 以外のシステムのユーザはすべて、使用権利 Id **qf573k** を使用して取得した充当レコードに含まれているライセンスをアクティブ化できる。

INCLUDE の例

```
INCLUDE paint USER picasso  
INCLUDE paint USER mondrian  
INCLUDE paint HOST bigbrush
```

このオプション・ファイルでは、次のように指定しています。

- ユーザ **picasso** は、ネットワーク上の任意のマシンで機能 **paint** を使用できる。
- ユーザ **mondrian** は、ネットワーク上の任意のマシンで機能 **paint** を使用できる。
- ホスト **bigbrush** のすべてのユーザは、機能 **paint** を使用できる。
- **picasso**、**mondrian**、ホスト **bigbrush** のユーザ以外のすべてのユーザは、機能 **paint** にアクセスできない（暗黙的な指定）。

INCLUDE_ENTITLEMENT の例

```
INCLUDE_ENTITLEMENT gy7210 USER tom  
INCLUDE paint USER anthony  
INCLUDE paint HOST jupiter
```

このオプション・ファイルでは、次のように指定しています。

- ユーザ **tom** は、ネットワーク上のどのシステムでも、使用権利 Id **gy7210** を使用して取得した充当レコードに含まれているライセンスをアクティブ化できる。
- ユーザ **anthony** は、ネットワーク上のどのシステムでも、使用権利 Id **gy7210** を使用して取得した充当レコードに含まれているライセンスをアクティブ化できる。
- ホスト **jupiter** のユーザはすべて、使用権利 Id **gy7210** を使用して取得した充当レコードに含まれているライセンスをアクティブ化できる。
- 暗黙的には、**tom**、**anthony**、またはホスト **jupiter** のユーザ以外は、使用権利 Id **gy7210** を使用して取得した充当レコードに含まれているライセンスをアクティブ化できない。

環境変数

アプリケーションを使用するために環境変数は不要です。FLEXenabled 環境変数は通常デバッグを行う場合や、ライセンスのデフォルトの場所を変更するために使用します。

環境変数の設定方法

FLEXnet Publisher ライセンス・ツールキット環境変数は、次の 2 とおりの方法で設定します。

- プロセスの環境で設定する。
- レジストリ (Windows バージョン 6.0 以前)、または UNIX でレジストリと同様に機能する `$HOME/.flexlmrc` (UNIX バージョン 7.0 以前) で設定する。

Windows レジストリ

Windows Vista 以外の Windows システムでは、レジストリは次の場所にあります。

HKEY_LOCAL_MACHINE\Software\FLEXlm License Manager

UNIX では、同じ情報が `$HOME/.flexlmrc` に保存されます。このファイルでは、構文は `variable=value` になります。

Windows Vista では、次の場所にあります。

HKEY_CURRENT_USER\Software\FLEXlm License Manager

優先度

変数が `LM_LICENSE_FILE` または `VENDOR_LICENSE_FILE` の場合、環境およびレジストリの両方が使用され、最初に環境が使用されて、レジストリがパスに追加されます。

他の変数では、環境が設定されると環境のみ使用され、それ以外の場合はレジストリが使用されます。つまり、レジストリは環境が設定されていないときにだけ使用されます。

環境変数

次の表に、さまざまな環境変数とその定義を示します。

表 13-1: 環境変数

変数	定義
FLEXLM_BATCH	Windows のみ：対話型のポップアップが表示されないようにします。バッチ・アプリケーションの場合は 1 に設定します (バージョン 7.0 以降のクライアント)。
FLEXLM_DIAGNOSTICS	アプリケーションからエラー・メッセージ・テキストが出力されない場合のデバッグに使用されます。必要な診断情報量に応じて、1、2、または 3 に設定します。「 FLEXLM_DIAGNOSTICS 」を参照してください (バージョン 5.0 以降のクライアント)。
FLEXLM_TIMEOUT	Windows のみ：27000 ~ 27009 の範囲のライセンス・サーバ・ポートに接続するときに FLEXenabled アプリケーションで使用するタイムアウト値を設定します。 値は、0 ~ 2,147,483,647 のマイクロ秒単位で指定します。デフォルト設定は 100,000 マイクロ秒です。
LM_BORROW	ライセンスの借用の開始および借用期間の設定に使用されます。詳細については、「 ライセンス借用の開始 」を参照してください。UNIX プラットフォームでは、レジストリに \$HOME/.flexlmrc ではなく \$HOME/.flexlmborrow が使用されます。
LM_PROJECT	LM_PROJECT の値は、レポート・ログ・ファイルに記録され、後で FLEXnet Manager によって報告されます。30 文字以内に制限されています (バージョン 5.0 以降のクライアントが必要です)。 これは、PROJECT でライセンスを RESERVE したり、INCLUDE したりする場合にも使用できます。次に例を示します。 RESERVE 1 f1 PROJECT airplane この機能には、バージョン 5.0 以降のクライアントとバージョン 7.0 以降のベンダ・デーモンが必要です。
LM_SERVER_HIGHEST_FD	ライセンス・サーバがアクセスできるファイル記述子の最大値を設定するために使用されます。

表 13-1: 環境変数 (続き)

変数	定義
LM_UTIL_CASE_SENSITIVE	FLEXlm ユーティリティで使用されます。1 に設定した場合は、ライセンス・ファイルの行が大文字と小文字を区別して処理されます。デフォルトでは、この変数は 0 に設定され、ライセンス・ファイルは大文字と小文字を区別しないで処理されます。この環境変数は、ライセンス・ファイルの大文字と小文字を区別するように、ソフトウェア・パブリッシャがライセンス・サーバを設定している場合のみ適用されます。
TCP_NODELAY	ライセンス要求の処理時にライセンス・サーバのパフォーマンスを向上させます。1 に設定すると、パフォーマンスを拡張できます。使用上の注意：この環境変数を有効にすると、ネットワーク・トラフィックが増大する可能性があります。
LM_LICENSE_FILE または VENDOR_LICENSE_FILE	パスをライセンス・ファイルにリセットします。“:” (UNIX) または “;” (Windows) で区切られたライセンスの検索パスにすることができます。VENDOR_LICENSE_FILE が使用されている場合、VENDOR は、このアプリケーションで使用されるベンダ・デーモンの名前になります。たとえば、Macrovision 製品の場合は MMSN_LICENSE_FILE になります。ファイル名または <i>port@host</i> にすることができます。「 環境変数を使用したライセンス検索パスの設定 」も参照してください (VENDOR_LICENSE_FILE にはバージョン 6.0 以降のクライアントが必要です)。

エラー・コード

この項では、一般的なフォーマットやエラーメッセージの説明など、FLEXnet Publisher ライセンス・ツールキット・エラー・メッセージについて説明します。

エラー・メッセージのフォーマット

アプリケーションから出力される FLEXnet Publisher ライセンス・ツールキット・エラー・メッセージの要素は次のとおりです。

- **エラー番号** – 正または負の番号です。
- **エラー・テキスト** – 問題を要約した短い説明です (< 80 文字)。
- **エラー説明 (オプション)** – 問題とその解決法または回避策についての簡単な説明です (3-5 行)。
- **マイナー・エラー番号** – 1 から始まる正の番号です。エラーを一意に識別します。ソフトウェア・パブリッシャが、さらに高度なサポートのために利用します。その内容はマニュアルには記載されていません。
- **システム・エラー番号 (オプション)** – オペレーティング・システムにより前回設定された UNIX または Windows OS エラー・コード。
- **システム・エラー説明 (オプション)** – システム・エラーに関する短い説明 (< 80 文字)。
- その他のサポート情報 (オプション)

エラー・メッセージはバージョン 6 になって改善されました。エラーの説明とサポート情報は、バージョン 6.0 以降を使用するアプリケーションでのみ使用できます。

FLEXnet Publisher ライセンス・ツールキットではこれらのエラー・メッセージに 2 つのフォーマットがあります。また、アプリケーションによってフォーマットがカスタマイズされることもあります。

フォーマット 1 (ショート)

FLEXnet error text (-lm_errno, minor_num[:sys_errno]) [sys_error_text]
エラー情報が見つからないこともあります。

例

```
Can't connect to license server machine (-15,12:61) Connection refused
```

フォーマット 2 (ロング – バージョン 6.0 以降)

FLEXnet error text
FLEXnet error explanation
[Optional Supporting information]
FLEXnet error: -lm_errno, minor_num. [System Error: sys_errno] ["system_error_text"]

例

```
Cannot connect to license server system
  The server (lmgrd) has not been started yet, or
  the wrong port@host or license file is being used, or the
  port or hostname in the license file has been changed.
Feature:      f1
Server name:  localhost
License path: @localhost:license.dat:./*.lic
FLEXlm error: -15,12. System Error: 61 "Connection refused"
```

エラー・コードの説明

次の表に、FLEXenabled アプリケーションの一般的なエラーを示します。

表 14-1: エラー・コード

エラー・コード	説明
21	FLEXnet ライセンス・サービスを起動するための十分な権限がないため、lc_flexinit が失敗しました。この問題を解決するには、サービスが自動的に起動するように設定します。
20	FLEXnet ライセンス・サービスがインストールされていません。
13	Mac OS X オペレーティング・システムで、必要なファイルへの計算パスが長すぎます。
12	Mac OS X オペレーティング・システムで、バンドル ID が無効です。
11	バンドル ID で指定されたフレームワークがロードされませんでした。
10	URL からパスを作成中にエラーが発生しました。
9	URL を作成中にエラーが発生しました。
8	パス文字列が UTF-8 フォーマットで指定されていません。

表 14-1: エラー・コード (続き)

エラー・コード	説明
7	lc_flexinit_cleanup への呼び出しの後に、lc_flexinit への呼び出しを行うことはできません。
6	実行プログラムが preptool を使用して準備されていません。preptool の詳細については、「Trusted Storage-based Programming Reference」を参照してください。
5	リソースを割り当てることができません。
4	初期化できません。
3	サポートされていないバージョンのオペレーティング・システムです。
2	アクティブ化ライブラリをロードできません。
1	アクティブ化ライブラリが見つかりません。
-1	ライセンス・ファイルが見つかりません。
-2	ライセンス・ファイルの構文が正しくありません。
-3	この機能に対応するライセンス・サーバ・システムが存在しません。
-4	ライセンスされたユーザ数にすでに達しています。
-5	そのような機能は存在しません。
-6	ライセンス・ファイルに TCP/IP ポート番号がなく、かつ FLEXnet ライセンス・サービスが存在しません (v6 より前のバージョンの場合)。
-7	ライセンス・サーバ・マネージャ・サービスに対するソケット接続がありません。
-8	ライセンス・キーまたはシグニチャが正しくありません (矛盾しています)。 ライセンス・キー/シグニチャとこの機能のデータが一致しません。このようなことが起こるのは通常、ライセンス・ファイルが変更されたときです。
-9	ホストが正しくありません。 このシステムの hostid とライセンス・ファイル内の hostid が一致しません。
-10	機能の有効期限が切れています。
-11	ライセンス・ファイルの日付フォーマットが正しくありません。
-12	ライセンス・サーバ・システムから返されるデータが正しくありません。
-13	ライセンス・ファイルに SERVER 行がありません。
-14	ネットワーク・データベース内で SERVER のホスト名が見つかりません。 ライセンス・ファイル内の SERVER 行でホスト名の検索に失敗しました。これは、NIS または DNS あるいはホスト・ファイルが正しくないときによく起こります。対処方法：ホスト名の代わりに IP アドレス (123.456.789.123 など) を使用してください。

表 14-1: エラー・コード (続き)

エラー・コード	説明
-15	ライセンス・サーバ・システムに接続できません。 サーバ (ladmin または lmgrd) がまだ開始されていないか、間違った <code>port@host</code> もしくはライセンス・ファイルが使われているか、ライセンス・ファイル内の TCP/IP ポートまたはホスト名が変更されています。 Windows XP SP3 プラットフォームでは、1 秒間に可能な TCP/IP 接続試行回数が制限されており、ご使用のアプリケーションはこの制限を上回った可能性があります。この制限を変更する方法については、製造元のマニュアルを参照してください。
-16	ライセンス・サーバ・システムからデータを読み込めません。
-17	ライセンス・サーバ・システムにデータを書き込むことができません。
-18	ライセンス・サーバ・システムはこの機能をサポートしていません。
-19	select システム・コールでエラーが発生しました。
-20	ライセンス・サーバ・システムがビジー (大部分ではない) です。
-21	ライセンス・ファイルがこのバージョンをサポートしていません。
-22	ライセンス・サーバ・システムで機能チェックイン障害が検出されました。
-23	ライセンス・サーバ・システムが一時的にビジー (新規サーバの接続中) です。
-24	この機能のキューにユーザが入っています。
-25	ライセンス・サーバ・システムがこのバージョンのこの機能をサポートしていません。
-26	この機能がサポートしているより多くのライセンスが要求されています。
-29	イーサネット・デバイスが見つかりません。
-30	ライセンス・ファイルを読み込めません。
-31	機能開始日にまだ至っていません。
-32	そのような属性はありません。
-33	ベンダ・デーモンとの暗号化ハンドシェイク方式が正しくありません。
-34	クライアントとライセンス・サーバ・システムの時刻差が大きすぎます。
-35	この機能のキュー内。
-36	ベンダ・デーモンの機能データベースが壊れています。
-37	この機能の重複選択が一致しません。バージョン 8.0 以降のベンダ・デーモンで廃止されています。
-38	機能の EXCLUDE リストのユーザ/ホストです。
-39	機能の INCLUDE リストのユーザ/ホストです。

表 14-1: エラー・コード (続き)

エラー・コード	説明
-40	動的メモリを割り付けることができません。
-41	機能がこれまで一度もチェックアウトされていません。
-42	パラメータが無効です。
-47	ベンダ・デーモンに時刻設定のチェック機能がありません。
-52	ベンダ・デーモンが所定のタイムアウト時間内に応答しませんでした。
-53	ベンダ定義のチェックアウト・フィルタによってチェックアウト要求が拒否されました。
-54	ライセンス・ファイルに FEATURESET 行がありません。
-55	ライセンス・ファイルの FEATURESET 行が正しくありません。
-56	ライセンス・ファイルから FEATURESET データを計算できません。
-57	socket コールが失敗しました。
-59	メッセージ・チェックサム・エラー。
-60	ライセンス・サーバ・システムのメッセージ・チェックサム・エラー。
-61	ライセンス・サーバ・システムからライセンス・ファイル・データを読み込むことができません。
-62	ネットワーク・ソフトウェア (TCP/IP) がありません。
-63	あなたはライセンス管理者ではありません。
-64	最小 Imremove 時間が経過する前に要求を Imremove してください。
-67	借用できるライセンスがありません。
-68	ライセンス借用サポートが有効ではありません。
-69	ライセンス・サーバ・システムで、FLOAT_OK はスタンドアロンでは動作できません。
-71	TZ 環境変数が正しくありません。
-73	ローカルなチェックアウト・フィルタによって要求が拒否されました。
-74	ライセンス・ファイル・パスの終わりを越えて読み込もうとしています。
-75	SYSSSETIMR コールが失敗しました (VMS の場合)。オペレーティング・システム障害のため、エラーが発生しました。
-76	内部 FLEXnet Publisher エラー。Acesso ソフトウェア 社にエラーを報告してください。
-77	バージョン番号が正しくありません。英字を含まない浮動小数点数でなければなりません。
-82	ライセンス・ファイルの PACKAGE 行が正しくありません。

表 14-1: エラー・コード (続き)

エラー・コード	説明
-83	クライアントの <i>FLEXnet Publisher</i> バージョンはサーバより新しいバージョンです。
-84	USER_BASED ライセンスに指定のユーザがありません。ライセンス・サーバのシステム・ログを参照してください。
-85	ライセンス・サーバ・システムがこの要求をサポートしていません。
-87	チェックアウトが、オプション・ファイルに指定された最大数を超過しています。
-88	システム・クロックの設定が戻されました。
-89	このプラットフォームはライセンスで承認されていません。
-90	ライセンス・ファイルのフォーマットがまだ有効でないか、ライセンス・ファイルにスペルミスがあります。 当該ファイルの発行対象となったバージョンが新しいため、このプログラムで <i>FLEXnet Publisher</i> が認識されません。
-91	暗号シードが一意ではありません。
-92	lmreread 中に機能が削除されたか、SERVER 行の hostid が正しくありません。
-93	この機能を使用できるのは別のライセンス・プールです。 これは警告条件です。サーバによって 1 つ以上の INCREMENT 行が単一のプールにプールされているときに、プール内の INCREMENT 行に対して要求が出されました。
-94	互換性のない属性を使用してライセンスを生成しようとしています。
-95	THIS_HOST へのネットワーク接続が失敗しました。 ライセンス・ファイルにある SERVER 行の this_host を実際のホスト名に変更してください。
-96	ライセンス・サーバ・マシンがダウンしているか応答しません。 サーバの起動方法をシステム管理者に問い合わせるか、参照先のホストが正しいかどうかを確認してください (LM_LICENSE_FILE 環境変数を参照)。
-97	必要なベンダ・デーモンがダウンしています。 1) lmadm または lmgrd ログ・ファイルを確認するか、2) lmreread を実行してください。
-98	この FEATURE 行は 10 進形式に変換できません。
-99	10 進形式のライセンスが正しく入力されていません。
-100	残存ライセンスを削除できません。
-101	すべてのライセンスがほかで予約されています。 すべてのライセンスがシステム管理者によってほかで予約されています。予約はオプション・ファイルに登録されています。サーバを再起動して、オプション・ファイル内の変更を有効にする必要があります。

表 14-1: エラー・コード (続き)

エラー・コード	説明
-102	FLEXid 借用エラーが発生しました。
-103	Terminal Server のリモート・クライアントは許可されません。
-104	それほど長く借用できません。
-105	機能がすでにライセンス・サーバに返却されています。
-106	ライセンス・サーバ・システムがネットワーク接続から外れています。 ベンダ・デーモンがそれ以上のユーザを処理できません。詳細については、「デバッグ・ログ」を参照してください。
-110	dongルを読み込めません：dongルまたはドライバをチェックしてください。 dongルが接続されていないか、このdongル・タイプに必要なソフトウェア・ドライバがインストールされていません。
-112	dongル・ドライバが見つかりません。 FLEXnet ID dongル hostid を読み込むためには、正しいドライバがインストールされている必要があります。これらのドライバは、ソフトウェア・パブリッシャから入手できます。
-114	必須の SIGN= キーワードがライセンス証明書内に見つかりません。 このライセンスの SIGN= バージョンをベンダから入手する必要があります。
-115	パブリック・キー・パッケージのエラー。
-116	このプラットフォームでは TRL がサポートされていません。
-117	借用に失敗しました。
-118	借用期間が満了しました。
-119	ライセンス・サーバで lmdown と lmreread を実行する必要があります。
-120	ライセンスを借用しているとき、サーバを lmdown することはできません。
-121	FLOAT_OK では、FLEXid hostid は 1 つだけ必要です。
-122	ローカル借用情報を削除できません。
-123	借用ライセンスの早期返却はサポートされていません。 詳細については、ベンダに問い合わせてください。
-124	借用ライセンスの返却エラー。
-125	PACKAGE コンポーネントを指定する必要があります。
-126	複合 hostid が初期化されていません。
-127	複合 hostid に必要な項目が見つからないか無効です。

表 14-1: エラー・コード (続き)

エラー・コード	説明
-128	エラーです。借用ライセンスが既知のどのサーバ・ライセンスとも一致しません。
-135	イベント・ログ有効化エラー。
-136	イベント・ログが無効です。
-137	イベント・ログ書き込みエラー。
-139	ネットワーク通信のタイムアウトです。
-140	メッセージ・コマンドが正しくありません。
-141	ソケット書き込みエラー。ピアがソケットをクローズしました。
-142	エラーです。単一の複合 hostid に結び付けられたバージョン固有ライセンスを生成できません。
-143	非カウント・ライセンスでは、バージョン固有のシグニチャはサポートされていません。
-144	ライセンス・テンプレートに冗長なシグニチャ指定子が含まれています。
-145	V71_LK シグニチャが正しくありません。
-146	V71_SIGN シグニチャが正しくありません。
-147	V80_LK シグニチャが正しくありません。
-148	V80_SIGN シグニチャが正しくありません。
-149	V81_LK シグニチャが正しくありません。
-150	V81_SIGN シグニチャが正しくありません。
-151	V81_SIGN2 シグニチャが正しくありません。
-152	V84_LK シグニチャが正しくありません。
-153	V84_SIGN シグニチャが正しくありません。
-154	V84_SIGN2 シグニチャが正しくありません。
-155	必要なライセンス・キーがライセンス証明書内に見つかりません。アプリケーションでは、ライセンス証明書にライセンス・キーが必要です。この証明書のライセンス・キー・バージョンをベンダから入手する必要があります。
-156	AUTH= keyword で指定されたシグニチャが正しくありません。
-157	信頼できる記憶域が侵害されました。修復が必要です。修復方法の詳細については、ベンダにお問い合わせください。
-158	信頼できる記憶域のオープンに失敗しました。詳細については、ベンダにお問い合わせください。
-159	充当レコードが無効です。詳細については、ベンダにお問い合わせください。

表 14-1: エラー・コード (続き)

エラー・コード	説明
-160	受け取ったアクティブ化要求が無効です。詳細については、ベンダにお問い合わせください。
-161	信頼できる記憶域で要求に一致する充当が存在しません。詳細については、ベンダにお問い合わせください。
-162	受け取ったアクティブ化応答が無効です。詳細については、ベンダにお問い合わせください。
-163	指定されたアクティブ化を返すことができません。詳細については、ベンダにお問い合わせください。
-164	戻りカウントが充当の最大値を超えます。詳細については、ベンダにお問い合わせください。
-165	修復カウントが残っていません。修復権限の詳細については、ベンダにお問い合わせください。
-166	指定された操作は許可されていません。詳細については、ベンダにお問い合わせください。
-167	オプション・ファイルの仕様によりユーザまたはホストがこの使用権利のアクティブ化から除外されているため、要求されたアクティブ化は拒否されました。
-168	オプション・ファイルには、使用権利の仕様が含まれていますが、このユーザまたはホストはこれらの仕様に含まれていません。

レポート・ログ・ファイル

ライセンス・サーバは、レポート・ログ・ファイルとデバッグ・ログ・ファイルの両方を生成します。この付録では、主にレポート・ログ・ファイルについて説明します。デバッグ・ログ・ファイルについては、「[デバッグ・ログ・ファイル](#)」を参照してください。

レポート・ログ・ファイルには機能の使用状況が記録されます。これはベンダ・デーモンによって生成されます。ただし、ベンダ・デーモンはデフォルトではレポートをログに出力しないため、この動作を有効にする必要があります。レポート・ログ内のデータは圧縮および認証され、レポジトリで体系的に格納されます。

Acesso ソフトウェア のソフトウェア管理ソリューションである **FLEXnet Manager** を使用することで、ライセンス使用データを高度な形式で可視化し、ライセンスの使用状況などの重要な情報について有益なレポートを作成できます。**FLEXnet Manager** は、これらのレポートをスケジュールに基づいて作成するように完全に自動化されており、Windows NT、Linux、UNIX などの異機種混合ネットワーク・サーバでライセンス・サーバとライセンスの使用状況を追跡できます。企業向け **FLEXnet Manager** の評価版の入手方法については、Acesso ソフトウェア (www.acesso.com) までお問い合わせください。

レポート・ログ出力の管理

ベンダ・デーモンがある程度の期間実行されると、レポート・ログの出力が増加します。ライセンスが活発に使用されている場合、これらのログ・ファイルのサイズは非常に大きなものとなります。これらのファイルをどこに出力し、どのようなサイクルで切り替えおよび保管を行うかを検討する必要があります。つまり、時間の経過と共にレポート・ログの出力ファイルを切り替える必要があります。各ファイルには、ある一定期間のライセンス使用状況が記録されます。

レポート・ログ・データは、ベンダ・デーモンが収集して内部データ・バッファ領域に保存し、そこから出力ファイルにまとめて書き出されます。デーモンの内部バッファは、毎分 1 回かあるいはバッファが満杯になったときの、どちらかが先に発生した時点で書き出されます。レポート・ログ・ファイルの内容を最新の状態にするためには、`lrmreread` コマンドを実行してバッファを強制的に書き出す必要があります。書き出しの終わったレポート・ログ・ファイルは、標準のファイル圧縮ツールでサイズを縮小してください。

不意の破壊やパフォーマンスの低下を避けるため、ベンダ・デーモンのレポート・ログを出力するファイルは、ベンダ・デーモンと同じシステムのローカル・ディスクに置くことをお勧めします。ベンダ・デーモンごとに固有のレポート・ログ・ファイルを出力先として設定する必要があります。

ベンダ・デーモンのレポート・ログ出力の有効化

ライセンス・サーバの開始前または開始後に特定のベンダ・デーモンのレポート・ログ出力を有効にする方法は 2 とおりあります。

- ベンダ・デーモンのオプション・ファイルに **REPORTLOG** 行を追加する。詳細については、「[REPORTLOG](#)」を参照してください。
- ベンダ・デーモン上で **lmswitchr** を起動する。詳細については、「[lmswitchr](#)」を参照してください。

ベンダ・デーモンのレポート・ログ出力のリダイレクト

特定のベンダ・デーモンのレポート・ログ出力を個別のファイルに移動することができます。各ファイルには、それぞれ異なる期間の活動が記録されます。ベンダ・デーモンが実行しているかどうかに応じて、次の 3 つの方法が使用できます。

- ベンダ・デーモンのオプション・ファイル内の **REPORTLOG** 行を変更し、**lmreread** (バージョン 8.0 以降のベンダ・デーモン) を起動または再起動してオプション・ファイルを再読み込みする。
- ベンダ・デーモン上で **lmswitchr** を起動する。詳細については、「[lmswitchr](#)」を参照してください。
- ベンダ・デーモン上で **lmnewlog** を起動する。バージョン 7.1 以降のベンダ・デーモンが必要です。詳細については、「[lmnewlog](#)」を参照してください。

デバッグ・ログ・ファイル

ライセンス・サーバは、デバッグ・ログ・ファイルとレポート・ログ・ファイルの両方を生成します。この項では、デバッグ・ログ・ファイルを中心に説明します。レポート・ログ・ファイルについては、「[レポート・ログ・ファイル](#)」を参照してください。

デバッグ・ログ・ファイルは、ライセンス・サーバのデバッグに役立つステータスとエラー・メッセージを記録するファイルです。ライセンス・サーバはデバッグ・ログ出力を常時生成します。デバッグ・ログの出力には、`ladmin` または `lmgd` に特有のイベントや各ベンダ・デーモンに特有のイベントなど、さまざまなものがあります。

デバッグ・ログ出力の管理

ライセンス・サーバ・マネージャとそのベンダ・デーモンが一定期間実行されると、そのログ出力はかなりの分量になります。時間の経過とともにデバッグ・ログ出力の重要度は減少するので、古いデバッグ・ログ出力と最近の出力を分離するために古い出力を保管または削除する必要があります。

パフォーマンスの低下を避けるため、各デバッグ・ログ・ファイルは、ライセンス・サーバ・マネージャおよびそのベンダ・デーモンを実行しているシステムのローカル・ディスクに置くことをおすすめします。しかし、デバッグ・ログ・ファイルをリモート・ディスクに置かざるを得ない場合に、ライセンス・サーバの動作が著しく遅いときは、`lmgd` を開始する際に `-nfs_log` オプションを指定すればパフォーマンスが改善されます。

デバッグ・ログ出力フォーマットについては、「[デバッグ・ログ・メッセージ](#)」を参照してください。

ライセンス・サーバのデバッグ・ログ出力の取得

ライセンス・サーバ・マネージャとして `lmadmin` を使用している場合、`lmadmin` およびそれが管理する各ベンダ・デーモンの個別のログ・ファイルが作成されます。ログ・ファイルは <ルート・ディレクトリ>/logs ディレクトリに書き出されます。

`lmgrd` とそれが管理するベンダ・デーモンは、デフォルトでは標準出力にデバッグ・ログ出力を書き出します。このデバッグ・ログ出力をファイルに書き出すためには、ライセンス・サーバの出力をファイルにリダイレクトするか、または `lmgrd` を開始する際に `-l debug_log_path` オプションを指定します。

特定のベンダ・デーモンのデバッグ・ログ出力の取得

同じライセンス・サーバの管理下にある各ベンダ・デーモンのデバッグ・ログ出力を、それぞれ固有のファイルに書き出すことができます（バージョン 8.0 以降のベンダ・デーモン）。これには 3 つの方法があります。

- ライセンス・サーバ・マネージャとして `lmadmin` を使用する場合は、[Vendor Daemon Configuration] 画面の GUI で場所とファイル名を直接設定する。ベンダ・デーモン・ログについては、オンライン・ヘルプを参照してください。
- 各ベンダ・デーモンのオプション・ファイルに `DEBUGLOG` 行を追加する。詳細については、「`DEBUGLOG`」を参照してください。
- ベンダ・デーモン上で `lmswitch` を起動する。詳細については、「`lmswitch`」を参照してください。

`lmgrd` は標準出力にデバッグ・ログ出力を書き出すことに注意してください。

実行中のベンダ・デーモンのデバッグ・ログ出力のリダイレクト

特定のベンダ・デーモンのデバッグ・ログ出力をファイルにリダイレクトすることができます。これには 2 つの方法があります。

- ベンダ・デーモンのオプション・ファイル内の `DEBUGLOG` 行を変更し、`lmreread` を起動してオプション・ファイルの再読み込みを実行する。詳細については、「`DEBUGLOG`」を参照してください。
- ベンダ・デーモン上で `lmswitch` を起動する。詳細については、「`lmswitch`」を参照してください。

ベンダ・デーモンのデバッグ・ログ出力の出力規制

デフォルトでは、デバッグ・ログ出力にすべてのイベントが記録されます。特定のベンダ・デーモンに関して記録されるイベントを制限するには、ベンダ・デーモンのオプション・ファイルに `NOLOG` 行を追加します。記録するイベントを制限する理由は、1 つにはデバッグ・ログ出力のサイズを小さくすることがあります。

参照
[NOLOG](#)

デバッグ・ログ・メッセージ

FLEXnet Publisher ライセンス・ツールキット・プロセスが生成するデバッグ・ログ・ファイルのフォーマットは次のとおりです。

hh:mm:ss (daemon) message

各パラメータの意味は、次のとおりです。

表 16-1: デバッグ・ログ・メッセージ

メッセージ	説明
<i>hh:mm:ss</i>	メッセージを記録した時刻。
daemon	lmadmin、lmgrd、またはベンダ・デーモン名のいずれか。要求されたすべてのライセンスが 1 つのデーモンで処理しきれないような状況では、名前の後ろに “_” と番号が付くことがあります。これは、メッセージが子デーモンから生成されたことを示します。
message	メッセージの本文

デバッグ・ログ・ファイルを使用して次のことができます。

- 設定上の問題を診断する。
- デーモン・ソフトウェアのエラーを診断する。



注意: デバッグ・ログ・ファイルを FLEXnet Manager のレポート生成に使うことはできません。

情報メッセージ

表 16-2 に、FLEXnet Publisher ライセンス・ツールキットで使用するさまざまな情報メッセージの一覧を示します。

表 16-2: 情報メッセージ

メッセージ	説明
Connected to host	このデーモンは、host 上のピアに接続されている。
CONNECTED, master is host	クォーラムが立ち上がっていて、全員がマスタを選択した状態のとき、ライセンス・デーモンはこのメッセージを記録する。
DENIED: num_lic feature to user	user が、feature の num_lic ライセンスへのアクセスを拒否された。
EXITING DUE TO SIGNAL nnn EXITING with code nnn	すべてのデーモンは、デーモンが終了した理由を出力する。
EXPIRED: feature	feature が期限を過ぎた。

表 16-2: 情報メッセージ (続き)

メッセージ	説明
IN: "feature" user (num_lic licenses)	user が、feature の num_lic ライセンスをチェックインした。
Lost connection to host	デーモンがノード host 上のそのピアともう通信できない。これで、クライアントが再接続を余儀なくされることがある。あるいはデーモンの数が最低限度を下回り、クライアントの終了プロセスが始まることある。マスタとの接続を失ったライセンス・デーモンはすべてのベンダ・デーモンを強制終了し、ベンダ・デーモンは自分自身を停止する。
Lost quorum	クォーラムを失ったデーモンは他のデーモンからの接続要求だけを処理する。
MULTIPLE vendor servers running. Kill and restart license daemon.	ライセンス・サーバ・マネージャが、同じ vendor という名前のベンダ・デーモンが複数実行されていることを検出した。ladmin または lmgrd およびすべての vendor デーモンを停止してから ladmin または lmgrd を再起動する必要がある。
OUT: feature user (num_lic licenses)	user が、feature の num_lic ライセンスをチェックアウトした。
RESERVE feature for USER user RESERVE feature for HOST host	feature のライセンスが user か host のために予約されている。
REStarted vendor (internet port nnn)	ベンダ・デーモン vendor が TCP/IP ポート nnn で再開された。
Retrying socket bind (address in use)	ライセンス・サーバは、“address in use” エラーを検出した場合、約 6 分間にわたって自分のソケットをバインドすることを試みる。
Selected (EXISTING) master host.	このライセンス・デーモンは、既存のマスタ host をマスタとして選択した。
SERVER shutdown requested.	デーモンが、ユーザの実行した kill コマンドにより停止を要求された。
Server started on host for: feature_list	リストされた機能について (おそらく新しい) サーバが開始された。
Shutting down vendor	ライセンス・サーバ・マネージャがベンダ・デーモン vendor を停止しようとしている。
SIGCHLD received. Killing child servers.	ベンダ・デーモンは、ライセンス・デーモンから停止を要求されたとき、このメッセージを記録する。
Started vendor	ライセンス・サーバ・マネージャは、新しいベンダ・デーモンを開始するとき、このメッセージを記録する。
Trying to connect to host	デーモンが、host への接続を確立しようとしている。

設定問題のメッセージ

表 16-3 に、FLEXnet Publisher ライセンス・ツールキットに見られる設定問題のメッセージの一覧を示します。

表 16-3: 設定問題のメッセージ

メッセージ	説明
host: Not a valid server host, exiting	このデーモンは無効なホスト名で実行された。
host: Wrong hostid, exiting	host の hostid が正しくない。
BAD CODE for feature	指定された機能名のライセンス・キーまたはシグニチャが正しくない。入力ミスか、エンド・ユーザにより変更された可能性がある。
CANNOT OPEN options file	ライセンス・ファイル内で指定されたオプション・ファイルを開くことができなかった。
Couldn't find a master	デーモンが合意できるマスタが見つからなかった。
License daemon: lost all connections	このメッセージは、サーバへのすべての接続が失われたとき記録される。たいていネットワークに問題がある。
Lost lock, exiting Error closing lock file Unable to re-open lock file	ベンダ・デーモンのロック・ファイルに問題がある。1 つのノードで複数のデーモンが実行されようとしている。ps コマンドで実行中の他方のデーモンを見つけ、kill -9 で強制終了する必要がある。
No DAEMON line for vendor	ライセンス・ファイルに vendor の DAEMON 行または VENDOR 行がない。
No DAEMON lines, exiting	ライセンス・デーモンは、ライセンス・ファイルに DAEMON 行または VENDOR 行がない場合、このメッセージを記録する。開始すべきベンダ・デーモンが存在しないので、ライセンス・デーモンは何もすることがない。
No features to serve!	ベンダ・デーモンは、サービスすべき機能を見つけられなかった。ライセンス・ファイルが壊れているか正しく入力されていない可能性がある。
UNSUPPORTED FEATURE request: feature by user	ユーザの要求した機能を、このベンダ・デーモンはサポートしていない。いくつかの理由が考えられる。たとえば、ライセンス・ファイルが正しくない、機能が使用期限を過ぎている、デーモンが間違ったライセンス・ファイルにアクセスしているなど。
Unknown host: host	ライセンス・ファイル内の SERVER 行で指定されたホスト名がネットワーク・データベース (おそらく /etc/hosts) 内に存在しない。

デーモンのソフトウェア・エラーに関するメッセージ

表 16-4 に、さまざまなデーモンのソフトウェア・エラーに関するメッセージの一覧を示します。

表 16-4: デーモンのソフトウェア・エラーに関するメッセージ

メッセージ	説明
accept: message	accept システム・コールでエラーが検出された。
Can't allocate server table space	malloc エラー。スワップ領域のサイズを確認する必要がある。
Connection to <i>host</i> TIMED OUT	デーモンが <i>host</i> に接続できなかった。
Illegal connection request to <i>vendor</i>	<i>vendor</i> への接続要求が出されたが、このベンダ・デーモンは <i>vendor</i> でない。
read: error message	“read” システム・コールでエラーが検出された。
select: message	“select” システム・コールでエラーが検出された。通常、システム・ネットワークに障害があることを示している。
Server exiting	サーバが終了中である。通常、これはエラーによるものでない。

FLEXnet のライセンス・バージョンの識別

コンポーネント間のバージョンの互換性

一般的には、常に最新バージョンの `ladmin`、`lgrd`、および `lmutil/lmtools` を使用してください。これらはすべて www.acresso.com から入手できます。このようにすれば、FLEXnet ライセンスの最新バージョンで導入されているさまざまな新機能の多くをすぐに利用できます。ただし、機能強化の中には新しいバージョンの FLEXnet Publisher ライセンス・ツールキットを使用してベンダ・デーモンがビルドされていることを要求するものもあれば、新しいバージョンの FLEXnet Publisher ライセンス・ツールキットを使用して FLEXenabled アプリケーションがビルドされていることを要求するものもあります。ベンダ・デーモンの最新バージョンについては、ソフトウェア・パブリッシャにお問い合わせください。

FLEXnet ライセンス コンポーネントのバージョンの互換性に関する規則は、次のとおりです。

- `lmutil/lmtools` のバージョン \geq
- `ladmin` (注意事項を参照) または `lgrd` のバージョン \geq
- ベンダ・デーモンのバージョン \geq
- FLEXenabled アプリケーションにリンクされるクライアント・ライブラリのバージョン \geq
- ライセンス・ファイル・フォーマットのバージョン

ライセンス・ファイルを除いて、これらのコンポーネントのバージョンは `lmver` を使用して検出できません。ベンダ・デーモン、`lgrd`、`lmutil` については、`-v` 引数を使用してバージョンを表示することもできます。



注意: `ladmin` はバージョン 9.2 以降のコンポーネントでのみ使用できます。

ライセンス・ファイルのバージョンの確認

以下の規則は、個々の FEATURE 行、INCREMENT 行、UPGRADE 行に適用されます。1 つのファイルに複数のバージョンを混在させることも可能です。特定のアプリケーションがチェックアウトする機能のみによって、その機能のライセンスのバージョンを決定します。

表 17-1: ライセンス・ファイルのバージョンの確認

バージョン	説明
バージョン 2	FEATURE 行の末尾に、空の引用符または引用符で囲まれた文字列がある。
バージョン 3 以上	INCREMENT 行または UPGRADE 行がある。
バージョン 4 以上	OVERDRAFT、DUP_GROUP、INTERNET、または PACKAGE がある。
バージョン 5 以上	SUPERSEDE、ISSUED、USER_BASED、HOST_BASED、または SN がある。
バージョン 6 以上	START がある。
バージョン 7.1 以上	SIGN= keyword がある。
>= バージョン 8.0	BORROW、FLOAT_OK、TS_OK がある。
>= バージョン 8.1	SUITE_RESERVED がある。
>= バージョン 8.4	COMPOSITE がある。
>= バージョン 11.5	ONE_TS_OK と SUPERSEDE_SIGN がある。

バージョンの概要

バージョン 1.0 – 1988

FLEXlm の最初のリリース。FLEXlm の基本的機能がすべて含まれています。

バージョン 1.5 – 1990 年 2 月

広く使用されるようになった初めてのバージョン (DEMO が含まれる)

バージョン 2.1 – 1991 年 3 月

- TIMEOUT のサポートを強化
- Ethernet ホスト ID のサポートを強化

バージョン 2.21 – 1991 年 11 月

- 多数のプラットフォームのサポートの追加、およびプラットフォーム固有の機能強化 (ホスト ID など)
- ホスト ID ANY を追加

バージョン 2.26 – 1992 年 3 月 (Sun 専用)

- ライセンス残留の追加

バージョン 2.4 – 1992 年 12 月

- インクリメンタル・ライセンス配布のための「全 Feature 行の使用」機能を追加
- ベンダ・カスタマイズ・ルーチンの機能拡張
- オプション・ファイルの機能拡張
- 新しいホスト ID タイプの追加： USER、 HOSTNAME、 DISPLAY
- ライセンス・ファイルの場所の形式として *port@host* を追加 (ライセンス・ファイルをサーバからダウンロードする)

バージョン 2.61 – 1993 年 3 月 (Sun 専用)

- ライセンス・ファイルに INCREMENT 行と UPGRADE 行を追加

バージョン 3.0 – 1994 年 5 月

- INCREMENT と UPGRADE の動作の変更と改善
- UDP プロトコルのサポートを追加
- `uname -i` ホスト ID を追加 (HP 向け)
- ライセンス検索パスとしての `LM_LICENSE_FILE` 環境変数のサポートを強化するため複数のジョブを追加
- 新しい、オプションのライセンス・ファイル形式、 `keyword=value` 構文でオプションの新機能 (`asset_info`、 `ISSUER`、 `NOTICE` などを指定可能、 “¥” ライセンス・ファイル継続文字、 1 機能につき 2,048 の文字制限)

バージョン 4.0 – 1994 年 12 月

- 浮動小数点の使用を排除 (信頼性向上のため)
- FEATURE 行への追加 (`ck`、 `OVERDRAFT`、 `DUP_GROUP`、 `INTERNET` ホスト ID)
- PACKAGE 行
- License Finder
- `lmdiag` と `FLEXLM_DIAGNOSTICS` (診断用)

バージョン 4.1 – 1995 年 5 月

- パフォーマンスの向上と新しいプラットフォームのサポート

バージョン 4.1 – パッチ・リリース 6、1995 年 10 月

- Windows 95 対応の Windows パッチ・リリース、パフォーマンス関数の多数の改善

バージョン 5.0 – 1996 年 3 月

- `port@host` の動作 (FLEXenabled アプリケーションがライセンス・ファイルを読み込まない) の改善
- ライセンス・ファイルの `USE_SERVER` 行による `port@host` の自動化
- ホスト ID のリスト – 1 つの機能を複数のホスト ID にロックする
- `FEATURE` の新しい属性：SN (シリアル番号)、`USER_BASED`、`HOST_BASED`、`MINIMUM`、`SUPERSEDE`、`ISSUED` (発行日)、`CAPACITY` (システムの容量に基づいて課金)
- ホスト名の代わりに IP アドレスを使用することで NIS と DNS の回避が可能
- レポート・ログ・ファイルのフォーマットの改善
- サーバの起動時に、2 週間以内に期限切れになるライセンスを通知
- オプション・ファイルの機能性の向上

バージョン 5.11 – 1997 年 2 月

- `SUPERSEDE` リスト、`PLATFORMS=` ライセンス属性
- 新しいオプション：`MAX`、`TIMEOUTALL`
- Windows コントロールパネルの追加
- Windows ライセンス・ジェネレータ `GENLIC` を追加

バージョン 5.12 – 1997 年 4 月

- パフォーマンスの向上と新しいプラットフォームのサポート

バージョン 6.0 – 1997 年 9 月

- `lmgrd` で複数のライセンス・ファイルの読み込みが可能
- `FLEXlm` ライセンス・ディレクトリのサポート：`*.lic` を自動的に使用
- ライセンス・ファイルを編集せずに顧客側で使用可能
- `DAEMON/VENDOR` 行のパスを省略可能 (`$PATH` 環境変数が使用される)
- 10 進形式のライセンス・フォーマットと、ライセンスをキーボードから入力するための `lminstall` ユーティリティ
- `FEATURE` 行が短くなり、読みやすさと入力やすさが向上

- 出荷時に、ユーザによる編集不要な別ファイルに PACKAGE 行を記述することが可能
- デフォルトの TCP/IP ポート番号を使用することで SERVER 行のポート番号を省略可能
- デフォルトのオプション・ファイルのパス
- SERVER 行でのホスト名として this_host をサポート
- VENDOR_LICENSE_FILE をサポート (例： DEMO_LICENSE_FILE)
- @host をサポート (デフォルトのポート番号が使用される)
- Windows のみ：ユーザにライセンス・ファイルまたはライセンス・サーバの名前を入力させる
- ライセンス・ファイルで大文字と小文字を区別しないようにすることが可能
- lmdown と lmreread で -vendor vendor 引数の指定が可能
- START=dd-mmm-yyyy ライセンス属性 (オプション)

バージョン 6.1 – 1998 年 6 月

- パフォーマンスの向上

バージョン 7.0 – 1999 年 8 月

- License Certificate Manager が自動ライセンス・フルフィルメントをサポート
- 「試用後に購入」ライセンシングのサポート
- 電子メール・プログラムから挿入された改行をライセンス・ファイルが処理
- ライセンス行の自動ソートが可能
- Windows 用 lmtools のインタフェースを改善
- lmgrd が Windows のコマンド・ラインで実行されたときにデフォルトではバックグラウンドで動作させる
- 3 サーバ冗長化の信頼性が向上 (バージョン 7.0 のベンダ・デーモンと lmgrd)
- lmreread と lmdown に -all 引数を指定することですべての lmgrd の停止または再読み込みが可能
- FLEXlm の環境変数の代わりとなるレジストリ (Windows) と \$HOME/.flexlmrc (UNIX) をサポート
- チェックアウト成功後にレジストリまたは \$HOME/.flexlmrc のライセンス・パスを自動的にインストール
- LM_PROJECT のオプションのサポート (PROJECT)
- パフォーマンスの向上、特に Windows NT
- Intel Pentium III CPU-ID (バージョン 7.0d 以降、1999 年 11 月)

バージョン 7.1 – 2000 年 8 月

- セキュリティの機能強化
- ライセンスでの SIGN= keyword
- 1mnewlog ユーティリティ (バージョン 7.0d 以降のベンダ・デーモン)

バージョン 7.2 – 2000 年 12 月

- パフォーマンスの強化

バージョン 8.0 – 2001 年 10 月

- 1mborrow (バージョン 8.0 以降のコンポーネント)、1mpath (バージョン 8.0 以降のベンダ・デーモン)、1mswitch (バージョン 8.0 以降のベンダ・デーモン) の各ユーティリティ
- 1mreread によるオプション・ファイルと SERVER ホスト名の再読み込み
- BORROW キーワードによるライセンスの借用

バージョン 8.1 – 2002 年 1 月

- CRO Security の機能強化

バージョン 8.2 – 2002 年 8 月

- Windows XP 準拠のためのサポートを追加

バージョン 8.3 – 2002 年 10 月

- 借用ライセンスの早期返却のサポートを追加

バージョン 8.4 – 2003 年 1 月

- 予約済みパッケージ・スイートのサポート

バージョン 9.0 – 2003 年 3 月

- COMPOSITE= ホスト ID タイプのサポート

バージョン 9.2 – 2003 年 7 月

- オプション・ファイルのキーワード GROUPCASEINSENSITIVE と MAX_BORROW_HOURS を追加

バージョン 9.5 – 2004 年 11 月

- 新しい環境変数：LM_UTIL_CASE_SENSITIVE

バージョン 10.0 – 2004 年 4 月

- FLEXnet ライセンスとしてのリリース

- 完全修飾ドメイン名のサポート

バージョン 10.1 – 2004 年 11 月

- USB ドングルに対する追加 FLEXid ドライバのサポート

バージョン 10.8 – 2005 年 4 月

- hostid に対する IPv6 アドレスのサポート
- 3 サーバ冗長構成サポートの機能拡張
- 共通のベンダ・デーモンのサポート

バージョン 11.1 – 2005 年 11 月

- 信頼できる記憶領域のライセンス権利のサポート
- このリリースで IPv6 による hostids のサポートが復帰

バージョン 11.5 – 2007 年 11 月

- NOLOG オプション・キーワードの新しい属性のサポート
- IPv6 のサポート
- 新しいエラー・コード
- 機能定義行の新しいキーワード – ONE_TS_OK と SUPERSEDE_SIGN

バージョン 11.6 – 2008 年 5 月

- 新しいライセンス・サーバ・マネージャ `lmadmin` にはバージョン 9.2 以降のコンポーネントが必要。
- Linux プラットフォームで複数のイーサネット・ホスト ID をサポート。

第 17 章：FLEXnet のライセンス・バージョンの識別
バージョンの概要

索引

数字

- 10 進形式のライセンス 39
- 3 サーバ冗長化
 - SERVER 行 66
 - 独立したライセンス・ファイル 23

A

- Acresso Software への連絡方法 2
- ANY hostid 101
- asset_info 74
- AUTH 71,76

B

- BORROW_LOWWATER 112

C

- COMPOSITE
 - hostid 101

D

- DEBUGLOG 112
- DEMO hostid 101

- DISPLAY
 - hostid 101
 - type 111
- dist_info 74
- DUP_GROUP 72

E

- EXCLUDE 113
- EXCLUDE_BORROW 113
- EXCLUDEALL 115

F

- FEATURE 行 70
 - asset_info 74
 - AUTH 71
 - dist_info 74
 - DUP_GROUP 72
 - FLOAT_OK 72
 - HOST_BASED 72
 - HOSTID 72
 - ISSUED 72
 - ISSUER 72
 - NOTICE 72
 - ONE_TS_OK 72
 - OVERDRAFT 73
 - PLATFORMS 73
 - SIGN 71
 - SN 73

sort 74
 START 73
 SUITE_DUP_GROUP 73
 SUPERSEDE 73
 TS_OK 73
 USER_BASED 73
 user_info 74
 vendor_info 74
 VENDOR_STRING 73
 機能のバージョン 71
 構文 75
 シグニチャ 71
 シリアル番号 73
 ソート順 74
 ベンダ・デーモン名 71
 有効期限 71
 優先度 74
 ライセンス・カウント 71
 FLEXLM_BATCH 132
 FLEXLM_DIAGNOSTICS 104
 レベル 1 104
 レベル 2 105
 レベル 3 106
 FLEXLM_TIMEOUT 132
 FLEXnet Manager 124
 FLOAT_OK 72
 FLOAT_OK 付きの FLEXnetID 82

G

GROUP 型 117
 GROUPEINSENSITIVE 117

H

HOST 型 111
 HOST_BASED 72
 HOST_GROUP 型 118
 HOSTID 72
 hostid
 ANY 101
 COMPOSITE 101
 DEMO 101
 DISPLAY 101
 HOSTNAME 101
 ID 102

INTERNET 102
 SERVER 行 68
 USER 102
 特殊 101
 HOSTNAME hostid 101

I

ID hostid 102
 INCLUDE 118
 INCLUDE_BORROW 119
 INCLUDEALL 121
 INCREMENT 行 70
 INTERNET
 hostid 102
 type 111
 IPv6
 サポートの概要 51
 ISSUED 72
 ISSUER 72

L

LINGER 121
 LM_BORROW 132
 LM_LICENSE_FILE 133
 LM_PROJECT 132
 オプション・ファイルでの使用 111
 プロジェクトについてのレポート 125
 LM_SERVER_HIGHEST_FD 132
 lmadm 15
 lmdiag
 構文 35
 トラブルシューティング 35
 lmdown
 lmadm での使用を有効にする 16
 アクセス制限 22
 構文 36
 無効化 22
 lmgrd
 起動 22, 23
 構文 22
 最新を使用 153
 冗長サーバ 23
 停止 36
 デバッグ・ログの開始 22

- デバッグ・ログ・ファイル 149
- バージョン間の互換性 21
- lmgrd の起動 23
- lmhostid
 - 構文 37
- lmhostid、構文 37
- lminstall
 - 構文 39
 - ライセンス・ファイルの形式 39
- lmnewlog、構文 40
- lmremove
 - アクセス制限 22
 - 構文 42
 - 無効化 22
 - 有効化 17
- lmreread
 - アクセス制限 22
 - 構文 43
- lmstat
 - lmreread 用の出力 44
 - 構文 44
- lmswitchr、構文 47
- lmswitch、構文 46
- lmtools 48
- lmutil
 - lmdiag 35
 - lmdown 36
 - lmhostid 37
 - lminstall 39
 - lmnewlog 40
 - lmremove 42
 - lmreread 43
 - lmstat 44
 - lmswitch 46
 - lmswitchr 47
 - lmver 48
- lmver、構文 48

M

- MAX 122
- MAX_BORROW_HOURS 123
- MAX_OVERDRAFT 123

N

- NOLOG 124
- NOTICE 72

O

- ONE_TS_OK 72
- OPTIONS=SUITE 76
- OPTIONS=SUITE_RESERVED 76
- OVERDRAFT 73

P

- PACKAGE 行 75
 - AUTH 76
 - OPTIONS=SUITE 76
 - OPTIONS=SUITE_RESERVED 76
 - SIGN 76
 - 構文 76
 - シグニチャ 76
- PLATFORMS 73
- PROJECT 型 111

R

- REPORTLOG 124
- RESERVE 125

S

- SERVER 行 66
 - 3 サーバ冗長化 66
 - hostid 68
 - 構文 68
 - デフォルトのポート番号 68
 - ポート番号 68
 - ホスト 68
 - ライセンス・ファイルの結合 98
- SIGN 71,76
- SN 73
- sort 74
- START 73
- SUITE_DUP_GROUP 73
- SUPERSEDE 73

T

TIMEOUT 126
 TIMEOUTALL 127
 TS_OK 73

U

UPGRADE 行、構文 77
 USE_SERVER 行 70
 USER hostid 102
 USER 型 111
 USER_BASED 73
 user_info 74

V

VENDOR 行 69
 オプション・ファイル・パス 70
 ベンダ・デーモンのパス 69
 ベンダ・デーモン名 69
 ポート番号 70
 vendor.opt 70, 108
 vendor_info 74
 VENDOR_LICENSE_FILE 65, 133
 VENDOR_STRING 73
 Vista 48, 131

あ

アクセス制限
 lmdown 22
 lmremove 22
 lmrread 22

え

エラー・コード
 説明 136
 フォーマット 135

お

オプション・ファイル
 BORROW_LOWWATER 112
 DEBUGLOG 112
 DISPLAY 型 111
 EXCLUDE 113
 EXCLUDE_BORROW 113
 EXCLUDEALL 115
 GROUP 型 117
 GROUPCASEINSENSITIVE 117
 HOST 型 111
 HOST_BASED で必須 72
 HOST_GROUP 型 118
 INCLUDE 118
 INCLUDE_BORROW 119
 INCLUDEALL 121
 INTERNET 型 111
 LINGER 121
 MAX 122
 MAX_BORROW_HOURS 123
 MAX_OVERDRAFT 123
 NOLOG 124
 PROJECT 型 111
 REPORTLOG 124
 RESERVE 125
 TIMEOUT 126
 TIMEOUTALL 127
 type 引数 110
 USER 型 111
 USER_BASED で必須 73
 作成 108
 ベンダ・デーモンによる読み込み 127
 優先度の規則 127
 例 128
 オプション・ファイルの作成 108
 オプション・ファイル・パス 70

か

環境変数
 FLEXLM_BATCH 132
 FLEXLM_DIAGNOSTICS 132
 FLEXLM_TIMEOUT 132
 LM_BORROW 132
 LM_LICENSE_FILE 133

LM_PROJECT 132
 LM_SERVER_HIGHEST_FD 132
 VENDOR_LICENSE_FILE 133
 設定 131
 環境変数の設定 131

き

機能
 version 71

こ

混在、ライセンス 80

さ

再ホスト、ライセンス 81

し

シングルチャ 71,76
 指定、ライセンス・ファイルの場所 63
 借用 84

そ

ソケット
 ライセンス・サーバが使用する数 11

ち

チェックアウトの問題の診断
 チェックアウトのトラブルシューティング 35

て

デバッグ・ログの切り替え
 lmswitch 46

と

同時ライセンス 79
 トラブルシューティング
 FLEXLM_DIAGNOSTICS の使用 104
 lmdiag を使用 35

ね

ネットワーク帯域幅と FLEXnet パブリッシャ 12
 ネットワーク・ライセンス 79

の

ノードロック・ライセンス 80

は

パッケージ・スイート 76

ふ

フローティング・ライセンス 79
 プロジェクトについてのレポート 125

へ

ベンダ・デーモン
 lmnewlog 40
 lmreread 43
 lmswitchr 47
 VENDOR_LICENSE_FILE 133
 オプション・ファイル 108
 冗長サーバ 23
 デバッグ・ログ・ファイル 149
 バージョンの互換性 21
 非カウント・ライセンス 81
 メモリ使用率 12
 レポート・ログ 16
 ベンダ・デーモンのパス 69
 ベンダ・デーモン名
 FEATURE 行 71
 VENDOR 行 69

ほ

ポート番号

SERVER 行 68

VENDOR 行 70

サーバのデフォルトの範囲 68

ホスト、SERVER 行 68

む

無効化

lmdown 22

lmremove 22

め

メモリの使用量、デーモン 12

も

モバイル・ライセンス

FLEXid へのノードロック 82

FLOAT_OK 付きの FLEXnetID 82

借用 84

プリペイド・ライセンス・プールによる充当 88

ユーザ名へのノードロック 87

ラップトップへのノードロック 82

ゆ

有効期限 71

優先度または FEATURE 行 74

ら

ライセンス

混在 80

借用 84

同時 79

ネットワーク・ライセンス 79

ノードロック 80

フローティング 79

ライセンス・カウント 71

ライセンス検索パス 94

ライセンス・サーバ 15

Windows サービスとしてインストール 49

アラート 15

使用されるソケット 11

ディスク領域の使用 12

デバッグ 103

ライセンス権利 16

ライセンス・サーバのステータス 44

ライセンス・サーバのデバッグ 103

ライセンス・サーバ・マネージャ 15

ライセンス・ディレクトリ 23,25

ライセンスの再ホスト 81

ライセンスのフォーマットの変換 39

ライセンス・ファイル

10 進形式 78

FEATURE 行 70

INCREMENT 行 70

lminstall 39

PACKAGE 行 75

SERVER 行 98

UPGRADE 行 77

USE_SERVER 行 70

VENDOR 行 69

行の順序 78,80

結合方法 97

更新後の再読み込み 43

異なるバージョン間の結合可能性 98

タイプ 79

場所の指定 63

フォーマット 65

複数サーバの使用 23

ライセンス・ファイルの行の順序 78,80

ライセンス・プール 71,110

り

リモート・ディスク使用のガイドライン 13

れ

レポート・ログの切り替え

lmnewlog 40

lmswitchr 47

レポート・ログの有効化 124

レポート・ログ・ファイル 12