

User-based Licensing Administration Guide

◆著作権

本書に記載されている情報の全部または一部、ならびに本書で紹介する製品は、著作権所有者の文書による事前の許可を得ない限り、転用・複製することを禁じます。

本書に記載されている製品は、Arm 社が提供する Arm 社製ツールを対象としており、製品の市販性または利用の適切性を含め、暗示的・明示的に関係なく一切の責任を負いません。また、Arm 社製ツールのバージョンアップに伴い、今後予告なしに本書内容を変更する場合があります。

本書は、対象製品の利用者をサポートすることだけを目的としています。

序章

この度は、弊社より Arm 製ソフトウェアツールをご購入いただきありがとうございます。

このガイドブックは User-based ライセンスモデルでライセンスされた、Arm ソフトウェア製品の導入に際し、ライセンスに関する速やかな立ち上げを支援します。

本ドキュメントは Arm の提供する **"User-based licensing Version 1.2024500 Administration Guide (Document ID: 107573_1.2024500_05_en)"** の内容に基づき翻訳、作成されたものです。内容につきましては全て上記ドキュメントをマスターといたしておりますので、ご使用の際には必ず上記ドキュメントを参照の上、本ドキュメントは参考資料として用いる形をお取りくださいますようお願い申し上げます。

目次

1. User-based ライセンシング Getting started	4
1.1 管理者のための User-based ライセンシングの用語	6
2. 製品の管理.....	9
2.1 製品とライセンスの追加	9
2.2 製品とライセンスシートの一覧表示	10
2.3 オーダーの一覧表示	12
2.4 オーダーの有効期限	12
2.5 割り当てたライセンス数を減らす	13
3. アクティベーションコードの管理	14
3.1 アクティベーションコードの生成.....	14
3.2 アクティベーションコードの参照.....	15
3.3 アクティベーションコードの無効化	16
4. ライセンスサーバのインストールと設定.....	18
4.1 ハードウェアおよびソフトウェア要件	18
4.2 ライセンスサーバのインストール	19
4.3 ライセンスサーバの設定.....	23
4.4 ライセンスサーバの登録.....	25
4.5 ライセンスサーバへの製品の割り当て	26
4.6 ユーザライセンスのアクティベーション	28
4.7 ライセンスサーバのリリース履歴.....	30
5. ライセンスサーバの管理	31
5.1 ライセンスサーバの停止.....	31
5.2 ライセンスサーバの起動.....	31
5.3 ライセンスサーバの再起動	32
5.4 ライセンスサーバの状態確認	32
5.5 ライセンスと使用状況の一覧	33
5.6 ライセンス数の変更	33
5.7 新しい製品バージョンのためのライセンス更新	35
5.8 新しいライセンスサーバへのライセンスの移行	36
5.9 ユーザの一覧.....	37
5.10 ライセンスのアクティベートとエラーの追跡.....	38
5.11 アクセスコントロールリストの作成.....	39
5.12 ユーザアクセスコントロールリストの表示	41
5.13 ユーザアクセスコントロールリストの削除	42
5.14 ライセンスサーババージョンの確認	43
5.15 管理者パスワードの変更	43
5.16 管理者パスワードのリセット	44
5.17 ライセンスサーバのモニタリング	45
5.18 ライセンスサーバの廃止	45

5.19	ライセンスサーバの復旧	46
5.20	ライセンスサーバのアップグレード	46
5.21	ライセンスサーバのアンインストール	48
5.22	ライセンスサーバディレクトリの変更	48
5.23	ライセンスサーバ要求数の変更	49
	ドキュメントに関する情報	51
	凡例	51
	役だつ情報	52

1. User-based ライセンシング Getting started

User-based License (UBL) は、以前のライセンスソリューションに置き換わる、Arm からの新しいライセンスモデルです。

UBL は、組織内のライセンスの管理と、UBL 管理対象ツールのエンドユーザのユーザビリティの改善の両方に対してメリットを提供します。UBL はそのライセンスによって複数のデバイス上で有効化された Arm 開発ツールへの無制限のアクセスをユーザに提供します。



Note

User-based のライセンスサーバソフトウェアは、Arm ソフトウェア製品で使用される以下を含むライセンスモデルとは異なり、互換性がありません；

- FlexNet Publisher node-locked and floating
- Keil® node-locked and floating
- Allinea node-locked and floating

Arm は、[User-based Licensing Migration Guide](#) で、様々なユースケースにおける User-based ライセンスモデルの使用に関するガイダンスを提供しています。

ライセンスの取得

ユーザにライセンスを割り当てる前に、Arm user-based licensing portal のアカウントが必要です。アカウントでは、1 つ以上の製品のライセンスが関連付けられている必要があります。製品が関連付けられていない場合は、[製品を追加](#)する必要があります。

User-based ライセンス導入のタイプ

UBL では、次の方法でライセンスを導入できます：

- Arm Cloud License Server (CLS)

管理者は、Arm portal から各ユーザのライセンスアクティベーションコードを取得します。管理者は電子メールなどでアクティベーションコードをユーザに通知します。

ユーザは、ライセンスソースとしてアクティベーションコードを使用して製品をアクティベートすることでライセンスを取得できます。このライセンスにより、ユーザはあらゆるデバイス上で製品に含まれる使用可能なすべてのツールにアクセスできます。ユーザは、管理者が Arm portal でアクティベーションコードを無効化するまでライセンスを保有します。

CLS 管理の詳細については、[アクティベーションコードの管理](#) の章を参照してください。

- Arm Local License Server (LLS)

ライセンスサーバをインストールし、組織内で管理します。管理者は Arm portal のアカウントでライセンスサーバとサーバに製品を登録します。管理者はライセンスサーバで利用できる各製品のライセンス数を選択します。

ユーザは、ライセンスサーバ URL をライセンスソースとして使用し、製品をアクティベートすることでライセ

ンスを取得できます。このライセンスにより、ユーザはあらゆるデバイス上で製品に含まれる使用可能なすべてのツールにアクセスできます。24 時間に 1 回、ユーザが Arm 開発ツールを起動すると、ライセンスが更新されます。ユーザが 7 日以内にライセンスを更新しない場合、そのライセンスはライセンスサーバにおいて他のユーザが使用できるようになります。

LLS の管理の詳細については、[ライセンスサーバのインストールと設定](#) および [ライセンスサーバの管理](#) の章を参照してください。

どちらのタイプが適していますか？

以下の LLS および CLS の特徴をレビューして、どちらが適しているかを決定してください：

- CLS の特徴：
 - 内部インフラ要件はありません。
 - 管理者は各アクティベーションコードを管理する必要があります。
 - ライセンスのアクティベートと更新のためにインターネットアクセスが必要です。
- LLS の特徴：
 - 1 台のサーバで多数のライセンスを簡単に管理できます。
 - 開発者の人数が大規模な場合のライセンス展開が容易になります。
 - ローカルに使用状況の診断情報を生成できます。

Arm は開発者が多数の場合にライセンス供与する場合は LLS を、開発者が少数の場合には CLS を推奨しています。ユーザが複数の拠点またはネットワークに分散している場合は、両方のタイプを組み合わせで使用できます。

クラウド環境での製品のアクティベート

クラウド環境で Arm 開発ツールを実行している場合：

- 社内イントラネットにアクセスできるクラウドインスタンス内の少数のツールについては、LLS を使用してライセンスをアクティベートできます。詳細については User-based Licensing User Guide 内「**ライセンスサーバを使用した製品のアクティベーション**」の章を参照してください。
- インターネットアクセスが可能なクラウドインスタンスの少数のツールについては、CLS を使用してライセンスをアクティベートできます。詳細については User-based Licensing User Guide 内「**アクティベーションコードを使用した製品のアクティベーション**」の章を参照してください。
- Arm 開発ツールを使用する多数のクラウドインスタンスが存在する場合や、インターネットや社内イントラネットにアクセスできないクラウドインスタンスの場合は次のいずれかの方法を使用できます：
 - LLS または CLS を使用してデバイスにライセンスの transfer file を作成します。続いて transfer file を使用してクラウドインスタンスで Arm 開発ツールのライセンスをアクティベートします。詳細については User-based Licensing User Guide 内「**代理アクティベーション**」の章を参照してください。
 - LLS または CLS を使用してデバイスのライセンスをアクティブにします。続いて次のいずれかを実行します：
 - ◇ 各クラウドインスタンスのライセンスキャッシュをアクティベート済みのライセンスキャッシュに置き換えます。

- ◇ ライセンスキャッシュを例えばネットワークドライブなどで使用できるようにします。各クラウドインスタンスで、ARMLM_CACHED_LICENSES_LOCATION 環境変数をライセンスキャッシュの場所に設定します。詳細については、User-based Licensing User Guide 内”ローカルライセンスキャッシュディレクトリの変更”の章を参照してください。

いずれの方法もライセンスの有効期限が切れる前にこのプロセスが繰り返されない場合、ライセンスは無効化されます。通常、ライセンスキャッシュは 7 日後に有効期限が切れますが、代理アクティベーションを使用して作成されたライセンスの場合は、この有効期限が長くなることがあります。

ユーザデータの転送

ユーザが製品をアクティベートしたりライセンスを更新したりすると、アクティベーションコードまたはライセンスサーバ(ローカルまたはクラウドベース)を使用して、アクティベーションサービスにリクエストが送信されます。このリクエストに含まれる個人データは、ユーザ名のみです。

1.1 管理者のための User-based ライセンシングの用語

Arm User-based ライセンスのドキュメントで使用される用語について、管理者特有の用語も含めて解説します。

以下の用語が、User-based ライセンスのドキュメントで使用されます：

Activation (アクティベーション)

特定のユーザの資格を登録する行為です。Arm 開発ツールで Arm 製品をアクティブにするとツールを使用するライセンスが有効になります。

Activation code (アクティベーションコード)

文字と数字の羅列です。Universally Unique Identifier (UUID) のフォーマットを持つ、特定のエンドユーザまたはサービスアカウントの製品登録シートを示します。



ローカルライセンスサーバを使用して製品をアクティベートすることもできます。

Arm Cloud License Server (Arm クラウドライセンスサーバ)

ユーザにアクティベーションコードを提供するために使用されます。

Arm license server (Arm ライセンスサーバ)

製品のアクティベーションコードがアクティベートされるか、ライセンスが更新されるときに接続する Arm に存在するサーバ。

Arm user-based licensing portal (Arm user-based ライセンシングポータル)

Arm 製品とライセンスを管理するためのポータルです。生成されたアクティベーションコードや、ライセンスはポータル内のライセンスサーバに保存されます。

Cached license (キャッシュされたライセンス)

ローカルに保存されたライセンス情報です。このライセンスは 7 日間有効ですが、Arm 製品がローカルライセンスサーバまたは Arm ライセンスサーバに正常に接続すると延長されます。

Development Tools (開発ツール)

ユーザライセンスを使用する Arm ソフトウェアツール。

Device (デバイス)

Arm 開発ソフトウェアツールを実行できるすべてのコンピューティングデバイスの総称です。ワークステーション、仮想マシン、サーバ、または携帯電話やタブレット等のモバイルデバイスなどが該当します。

End-user (エンドユーザ)

Arm 開発ソフトウェアツールを使用している方を示します。

Floating (フローティング)

ライセンスは一元的に保持され、クライアントからのチェックアウトに対してオンデマンドで引き渡されます。ライセンス管理機能が使用されている間ライセンスはチェックアウトされ、チェックインされるとライセンスの使用が終了します。フローティングライセンスは、クライアントにライセンスを提供するライセンスサーバに固定されます。このモデルは、最大同時使用によって制限されるので、同時ライセンシングモデル(concurrent licensing model) と呼ばれます。ライセンスが borrow またはキャッシュされない限り、クライアントはライセンス管理機能が使用されている間、ライセンスサーバとのネットワーク接続を維持する必要があります。

License (ライセンス)

アクティベーションによって出力されるデータ。ユーザの資格を示します。ライセンスは証明書(certificate) と呼ばれます。ライセンスは通常、Arm 開発ツールのアクティベーション時に使用されるデバイスのローカルファイルシステムに保存されます。

License server (ライセンスサーバ)

ライセンスをアクティベートおよび更新するために使用されるローカルサーバです。



アクティベーションコードを使用して製品をアクティベートすることもできます。

Node-locked (ノードロック)

ライセンス管理ソフトウェアが実行されているデバイスに固定されているライセンスです。このモデルは、デバイススペースライセンシングモデル(device-based licensing model) と呼ばれます。

Order (オーダー)

オーダーにより、Arm 製品とライセンスの権利が付与されます。オーダーには一意の番号、開始日と終了日が含まれます。オーダーの有効期限が切れた場合についての詳細は、[オーダーの有効期限](#) の章を参照してください。

Overdraft (オーバードラフト)

一部の製品では、所有するライセンス数を一時的に超えることができるオーバードラフトが存在します。

Product (製品)

ユーザが使用する資格のある Arm 開発ツールを示す Arm 製品のタイトルです。この製品を使用するとユーザは 1 つまたはそれ以上の Arm 開発ツールを使用できます。

Proxy activation (代理アクティベーション)

ライセンスファイルを作成し、それをコピーして別のデバイスにインポートする製品のアクティベートの方法です。これは、次の場合に役立ちます: *デバイスがネットワーク要件を満たしていない場合 *Continuous Integration(CI) などの自動化された並列ワークフローで複数のデバイスを使用する場合

Seat (シート)

使用可能または購入されたライセンスの数をカウントするための抽象的な用語です。User-based のライセンシングモデル下では、Arm 開発ソフトウェアツールがインストールまたは実行されているデバイスの数に関係なく、ユーザごとに 1 つのシートのみ存在します。

Serial number (シリアル番号)

Arm user-based licensing portal で、ライセンスを伴う新しい製品を追加したり、既存の製品のライセンス数を増やしたりするために使用される番号です。シリアル番号はオーダーに含まれており、次のフォーマットをとります:

XXXXXX-XXXXXX-XXXXXX-XXXXXX-XXXXXX

Service account (サービスアカウント)

ソフトウェアの自動使用を調整するために使用されるオペレーティングシステムのアカウントです。

User (ユーザ)

Arm 製品ソフトウェアを使用するエンティティを識別するために使用される一般的な用語です。エンドユーザまたはサービスアカウントにすることもできます。

Username (ユーザ名)

オペレーティングシステム上のアカウントを識別する名前です。ユーザ名は Arm 開発ツールを実行するオペレーティングシステムの各プロセスに関連付けられています。

2. 製品の管理

ユーザにライセンスを割り当てる前に、Arm user-based licensing portal のアカウントに 1 つ以上の製品のライセンスが関連付けられている必要があります。アカウントにさらに製品を追加することもできます。

2.1 製品とライセンスの追加

シリアル番号を使用して、Arm user-based licensing portal でアカウントに製品を追加できます。同様に既存の製品にライセンスを追加することもできます。

この操作について

Arm または Arm 販売代理店から製品をご購入いただくと、以下のいずれかの方法で製品オーダーを受け取ります。

- シリアルコードを受け取る。このトピックでは、シリアルコードを使用して製品オーダーをアカウントに追加する方法について詳しく説明します。

製品を追加する方法の概要については、[Accessing the Arm License Portal video tutorial](#) (1:02min.)の動画をご覧ください。

- Arm が製品オーダーを直接アカウントに追加します。この場合、このトピックの手順に従う必要はありません。

手順

1. Arm user-based licensing portal <https://developer.arm.com/support/licensing/user-based> にアクセスします。
2. Arm user-based licensing portal に複数のアカウントがある場合は、使用したいアカウント ID をクリックします。



あらゆるアカウントに製品を追加できるため、正しいアカウント ID を選択していることを確認してください

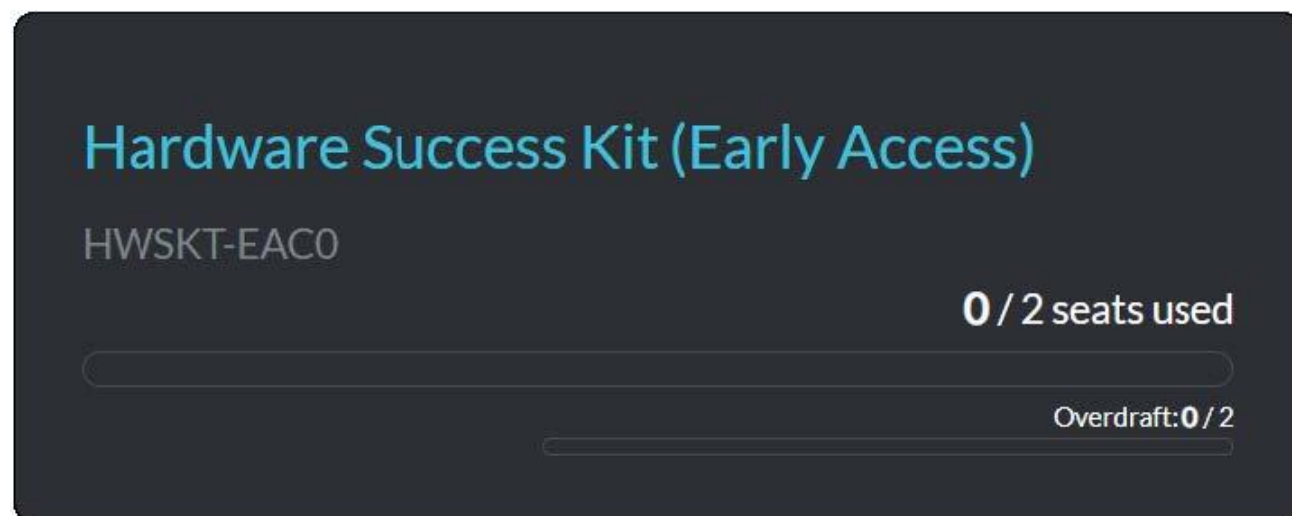
3. **Products** タブで、**Add a product** をクリックします。
4. 表示されたダイアログボックスで、シリアル番号を入力し、**Redeem** をクリックします。

結果

バナーに"Successfully redeemed serial number" と表示されます。追加された製品は、Products タブ内にタイル状に表示されます：

- product code に対するライセンスを過去に生成していない場合は、新しく製品のタイルが作成されます。
- product code に対する既存のライセンスがある場合は、製品タイル内に使用可能なシート数に関するライセンス情報が追加で表示されます。

図 2-1: 製品タイルの例



関連情報

[製品とライセンスシートの一覧表示](#)

[オーダーの一覧表示](#)

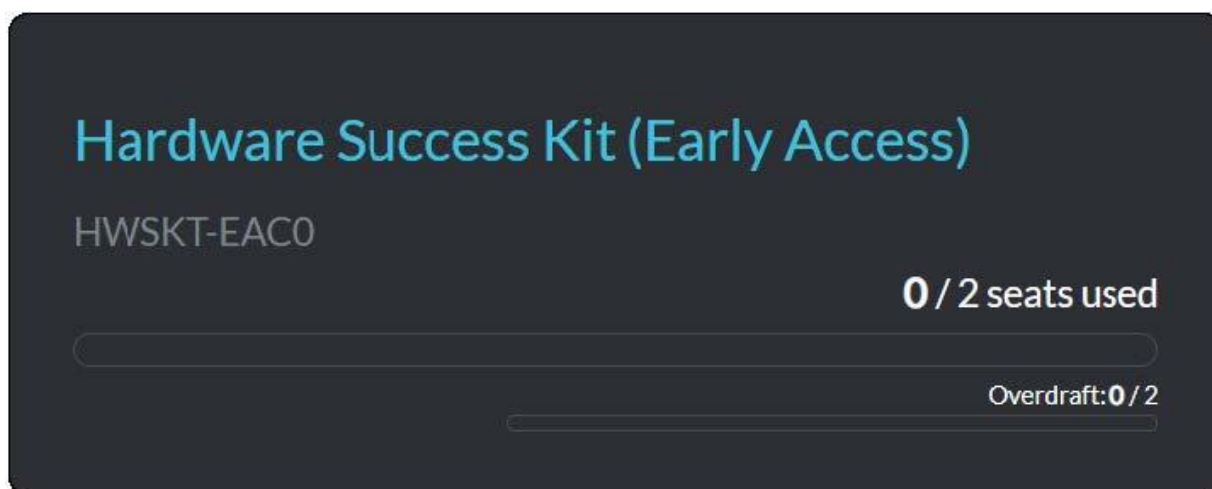
2.2 製品とライセンスシートの一覧表示

アクティブおよび利用可能なライセンスシートとともに製品の情報を表示できます。

手順

1. Arm user-based licensing portal <https://developer.arm.com/support/licensing/user-based> にアクセスします。
2. **Products** タブを選択します。
3. 保有している製品ごとにタイルが表示されます。例:

図 2-2: 製品タイルの例



タイルには以下が表示されます:

- ・ 製品名と product code
 - ・ 使用中および使用可能なライセンスシート数
 - ・ 可能な場合使用中および有効なオーバードラフトシート数
4. 製品に関連するライセンスの使用状況およびオーダーの詳細については、製品のタイルをクリックしてください。

図 2-3: 製品の詳細情報の例

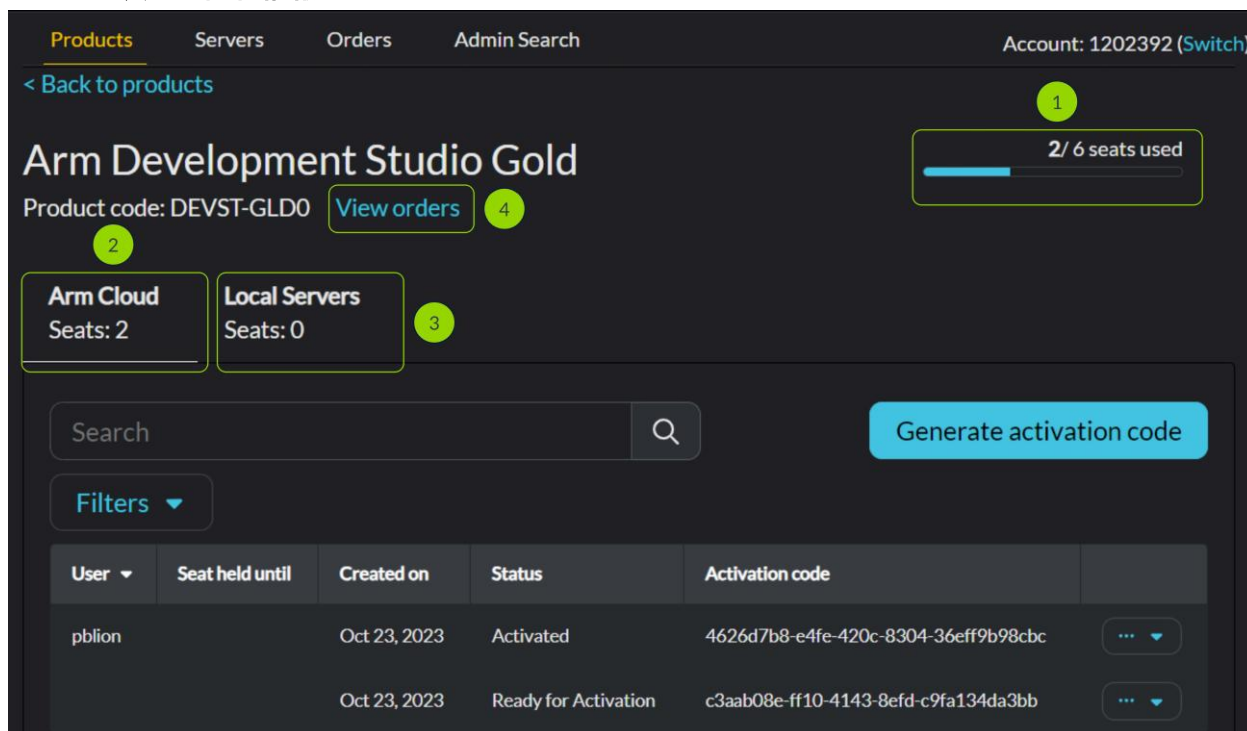


表 3-1: 製品の詳細情報

1	該当製品に対する使用中のライセンスシート数と最大ライセンスシート数
2	使用中のアクティベーションコードライセンスのシート数。 Arm Cloud タブをクリックすると、使用中のアクティベーションコードライセンスシートの詳細が表示されます。
3	ライセンスサーバで使用中のライセンスシート数。 Local servers タブをクリックすると、当該製品のライセンスを使用しているライセンスサーバの詳細が表示されます。各ライセンスサーバに割り当てられたライセンスは Seats allocated と表示されます。
4	View orders をクリックすると、製品に関連付けられたオーダーが Orders タブに表示されます。一覧表示されるオーダーは当該製品についてフィルタリングされています。

2.3 オーダーの一覧表示

製品ライセンスに関するオーダーの一覧を表示します。

手順

1. Arm user-based licensing portal <https://developer.arm.com/support/licensing/user-based> にアクセスします。
2. **Orders** タブを選択します。
3. 以下の詳細情報とともにオーダーが一覧表示されます:

- ・ **Order ID**
- ・ **Product name**
- ・ **Product code**
- ・ **Start date**
- ・ **Expiry date**
- ・ **Status** (Valid, Expired, または Not started)
- ・ **Seats**
- ・ **Overdraft**
- ・ **Used Overdraft**

※これら用語の詳細については [User-based ライセンシングの用語](#) の章を参照してください。

4. **Filters** をクリックして、フィルタ対象の **Product** または **Status** を選択することでオーダーをフィルタリングできます。フィルタされた項目は **Filters** ボタンの横にタイルとして表示されます。
たとえば、Status が **Expired** 且つ **Not Started** のオーダーを選択して表示するなど、複数のフィルタを追加できます。

フィルタタイルの × の箇所をクリックして個別のフィルタを削除する、または **Clear filters** をクリックしてすべてのフィルタを削除できます

2.4 オーダーの有効期限

すべてのオーダーには有効期限があります。有効期限を過ぎると、オーダーした製品とライセンスは利用できなくなります。これにより以下のような影響があります:

- ・ Cloud License Server(CLS) の場合、期限切れのオーダーに関連するアクティベーションコードライセンスは、使用可能なライセンスシートのある注文に移動されます。ライセンスは、有効期限が最も新しいオーダーから移動されます。

オーダーの有効期限が切れた後、その製品において有効なライセンスシートが十分でない場合、ユーザは製品の新しいライセンスをアクティベートしたり、キャッシュされたライセンスを更新したりすることはできません。

- Local License Server(LLS) の場合、ライセンスサーバ上の製品で使用可能なライセンスの数が減ります。ライセンス数の減少により、一部のユーザが製品の新しいライセンスをアクティベートしたり、キャッシュされたライセンスを更新したりできなくなる可能性があります。

有効なライセンスの数と期限切れのライセンスの数を確認するには、以下のライセンスサーバコマンドを使用します：

```
armlm_list_products --show-expired
```

製品に対して有効なライセンス数が、ライセンスサーバに割り当てているアクティベーションコードまたはライセンス数を超えた場合、ポータルでは以下ようになります：

- 製品タイルには、有効なシート数よりも多くのアクティベーションコードまたは割り当てられたシートがあるという警告が表示されます。
- 製品の詳細ページには、その製品において超過して割り当てられているライセンス数を含む警告が表示されます。アクティベーションコードの場合、製品に使用できる十分なオーバードラフトライセンスのシートがある場合は、**Use <number> overdraft seats** をクリックしてこれらを使用できます。ここでの<number> は超過して割り当てられたライセンス数を示します。
- 新しいアクティベーションコードを生成したり、ライセンスサーバにライセンスを割り当てたりすることができなくなります。

ライセンスの猶予期間

Arm 開発ツールライセンスでは、ライセンスに対するオーダーの有効期限に基づいた有効期限が設定されています。ライセンスには猶予期間があり、ライセンスの有効期限後も短期間、製品を使用することができます。この猶予期間を利用して、追加のライセンスを利用できるようにしてください。

より多くのライセンスを利用可能にする

製品に十分なライセンスがない場合は、以下の方法でより多くのライセンスを使用できるようになります：

- 必要な製品とライセンスのオーダーを購入する
- 以下を行って割り当てられたライセンス数を減らす
 - [アクティベーションコードの無効化](#)
 - [ライセンスサーバに割り当てられたライセンス数を減らす](#)

2.5 割り当てたライセンス数を減らす

より多くのライセンスを使用する必要がある場合は、割り当てられたライセンスの数を減らすことができます。

- アクティブでなくなった Cloud License Server ユーザの場合、そのユーザの[アクティベーションコードを無効化](#)します。
- Local License Server でライセンスサーバ上に未使用のライセンスがある場合は、その[ライセンスサーバに割り当てるライセンス数を減らす](#)ことができます。

3. アクティベーションコードの管理

Arm user-based licensing アクティベーションコードの管理について解説します。

3.1 アクティベーションコードの生成

Arm user-based licensing モデルで Arm 開発ツールのユーザに対してアクティベーションコードを生成します。

始める前に

- Arm user-based licensing portal <https://developer.arm.com/support/licensing/user-based> にアカウントを持っていないければなりません。
- user-based licensing portal のアカウントは有効なライセンスを伴う 1 つ以上の製品を保有していなければなりません。より詳細については[製品とライセンスシートの一覧表示](#) の章を参照してください。
- アカウントが要求される製品を保有していない場合、製品を追加しなければなりません。より詳細については[製品とライセンスの追加](#) の章を参照してください。
- アクティベーションコードの生成、アクティベーションおよび無効化についての概要は [Cloud-based Licenses and Activation Codes video tutorial](#) を参照してください。

手順

1. Arm user-based licensing portal <https://developer.arm.com/support/licensing/user-based> にアクセスします。
2. **Products** タブを選択してから **Arm Cloud** タブを選択します。
3. **Generate Activation Code** をクリックします:
 - 該当する製品に使用可能なシートがある場合、ページ上部に **Activation code <activation code> generated successfully** と示されたバナーが表示されます。**Ready for activation** のステータスが示されたアクティベーションコードがアクティベーションコードのリストに表示されます。

オーバードラフトシートではなく、最初に期限切れとなるオーダーからライセンスシートが取得されます。

- 使用可能なシートはないものの、オーバードラフトシートを保有している場合、オーバードラフト状態で使用されていることを警告するダイアログが表示されます。アクティベーションコードを生成するには **Confirm** をクリックします。ページ上部に **Activation code <activation code> generated successfully** と示されたバナーが表示されます。**Ready for activation** のステータスが示されたアクティベーションコードがアクティベーションコードのリストに表示されます。
- オーバードラフトシートが有効で、最初に期限切れとなるオーダーからライセンスシートが取得されます。
- 該当する製品にシートおよびオーバードラフトシートがない場合、**Generate Activation Code** のボタンは無効化されます。
4. 該当のアクティベーションコード上で  をクリックし、**Copy activation code** を選択してアクティベーションコードをクリップボードにコピーします。

5. ユーザがアクティベーションコードを使用できるようにします。たとえば、アクティベーションコードを記載したメールをユーザに送信します。

結果

ユーザがアクティベーションコードをアクティベートする方法の詳細については、User-based Licensing User Guide 内“アクティベーションコードを使用した製品のアクティベーション”の章を参照してください。状況によっては、管理者またはユーザが別の方法でアクティベーションコードをアクティベートする必要があります：

- 以下の場合、代理アクティベーションが使用できます：
 - インターネットにアクセスできないユーザ
 - 7 日間を超えるライセンスアクティベーション期間を必要とするユーザ
 - 同一ユーザに対して複数のデバイスでライセンスのアクティベートが必要なワークフロー。Continuous integration(CI) は、このタイプのワークフローの一例にあたります。
- ユーザが以下のデバイスで Arm 開発ツールを実行している場合は、ローカルライセンスキャッシュのディレクトリを変更する必要があります：
 - ローカルドライブがない
 - ユーザのホームディレクトリがない
 - ユーザのローカルドライブに十分なディスク容量がない

アクティベートを行った後、製品ライセンスは少なくとも 7 日間ユーザにバインドされます。その後、同じデバイスで User-base のライセンスをサポートする Arm 開発ツールを使用すると、Arm ライセンスサーバに接続してライセンスが続く 7 日間更新されます。Arm ライセンスサーバにアクセスするためのネットワーク要件の詳細については、User-based Licensing User Guide 内“User-based ライセンシングの要件”の章を参照してください。

関連情報

[アクティベーションコードの参照](#)

[アクティベーションコードの無効化](#)

3.2 アクティベーションコードの参照

製品に対して生成されたアクティベーションコードを参照します。

手順

1. Arm user-based licensing portal <https://developer.arm.com/support/licensing/user-based> にアクセスします。
2. **Products** タブを選択してから **Arm Cloud** タブを選択します。
3. 生成済みで削除されていないすべてのアクティベーションコードが一覧表示されます。アクティベーションコードのステータスは以下のとおりです：
 - **Ready for activation**: ユーザによるアクティベーションが行われていません。
 - **Activated**: ユーザによってアクティベーション済みです。
 - **Revoked**: アクティベートされたコードは失効していますが、キャッシュされたライセンスの有効期限はまだ切れていません。キャッシュされたライセンスがすべて期限切れになると、コードは削除されます。通常、キャッシュされたライセンスの有効期限は最大 7 日間ですが、代理アクティベーションを使用して作成されたライセンスの場合は、これより長くなる場合があります。



リストに表示されたアクティベーションコードに関連付けられているライセンスシートは、別のアクティベーションコードを生成するために使用できません。ライセンスシートはアクティベーションコードが削除され、一覧に表示されなくなると使用可能になります。詳細については、[アクティベーションコードの無効化](#) の章を参照してください。

4. **Filters** をクリックして、フィルタ対象の **Status** を選択することでアクティベーションコードをフィルタリングできます。フィルタされた項目は **Filters** ボタンの横にタイルとして表示されます。
たとえば、Status が **Ready for activation** 且つ **Activated** のオーダーを選択して表示するなど、複数のフィルタを追加できます。

フィルタタイルの × の箇所をクリックして個別のフィルタを削除する、または **Clear filters** をクリックしてすべてのフィルタを削除できます

3.3 アクティベーションコードの無効化

Arm user-based licensing モデルで Arm 開発ツールのユーザに対するアクティベーションコードを無効化します。

この操作について

無効化されたアクティベーションコードのライセンスシートは、利用可能になるまでに最大 7 日かかります。代理アクティベーションを使用して作成されたライセンスの場合は、これより長くなる場合があります。

手順

1. Arm user-based licensing portal <https://developer.arm.com/support/licensing/user-based> にアクセスします。
2. **Products** タブを選択してから **Arm Cloud** タブを選択します。
3. 該当のアクティベーションコード上で  をクリックし、**Revoke activation code** を選択します。

結果

以下の場合、即座にアクティベーションコードが削除されます：

- アクティベーションコードのステータスが **Ready for Activation** の場合
- アクティベーションコードのステータスが **Activated** 且つユーザが過去 7 日間の間に Arm 開発ツールを使用していない場合

アクティベーションコードのステータスが **Activated** であり、ユーザが過去 7 日の間に Arm 開発ツールを使用していた場合、アクティベーションコードのステータスは **Revoked** に変わります。ユーザは新しいデバイスでライセンスをアクティベートできなくなるとともに、過去にキャッシュされたライセンスは更新されません。アクティベーションコードは、**Arm Cloud** タブの **Seat held until** 列に表示されている日付に削除されます。



ライセンスシートは Inactive となったアクティベーションコードが削除された場合のみ使用可能になります。

必要に応じて、新たに利用可能になったライセンスシートをライセンスサーバに追加できます：

- 新しいライセンスサーバを作成する場合は、[ライセンスサーバのインストールと設定](#) の章を参照してください。
- 既存のライセンスサーバのライセンス数を更新する場合は、[ライセンス数の変更](#) の章を参照してください。

関連情報

[アクティベーションコードの生成](#)

4. ライセンスサーバのインストールと設定

Arm user-based licensing モデルにもとづきライセンスを供与される Arm 製品のライセンスサーバのインストール、設定、および登録方法について説明します。この操作によって、ライセンスサーバに製品ライセンスが登録されます。

4.1 ハードウェアおよびソフトウェア要件

この章で解説するハードウェアおよびソフトウェアの要件は、Arm user-based ライセンス用のライセンスサーバソフトウェアのための要件です。一般的な導入セットアップでは、ライセンスサーバソフトウェアは、クライアントデバイスで実行される Arm 開発ツールとは別のライセンス管理専用のデバイスで実行されます。

ハードウェア要件

ライセンスサーバでは、次の最小ハードウェア要件があります：

- プロセッサ： dual core 64-bit x86 2GHz を実装するプロセッサ(または同等の性能を持つもの)
- メモリ： 4GB
- ストレージ： 500MB

ライセンスサーバは以下の仮想マシンをサポートしています：

- Amazon EC2
- Citrix XenServer 7.2
- Google Compute Cloud
- Microsoft Hyper-V 6.3 on Windows Server 2016
- Oracle VirtualBox 6.1.18
- Parallels 15.1.2
- QEMU-KVM 2.7
- VMware ESXi 6.5
- VMware Workstation 14.1.5

ライセンスサーバ用に生成されたライセンスは、ライセンスサーバの hostid にロックされます。

したがってハードウェアでは固定且つ安定した hostid が少なくとも 1 つ必要です。

たとえばライセンスサーバホストとなるデバイスを再起動するとき、または有線ネットワークと無線ネットワークを切り替えるときに、安定した hostid が変わってはいけません。



Caution

現在のライセンスサーバの実装では、hostid はホストマシン上で有効となっているネットワークアダプタの 1 つの MAC アドレスです。

仮想マシン上で安定した hostid を確保するには、それぞれが静的な MAC アドレスを持つ、決まった数のイーサネットアダプタを保持しているよう仮想マシンを設定しておく必要があります。これらの MAC アドレスは仮想マシンの電源が入れなおされても固定されていなければなりません。

サポートされるオペレーティングシステム

ライセンスサーバは、次のオペレーティングシステムでサポートされています。

- Red Hat Enterprise Linux or CentOS 7
- Red Hat Enterprise Linux or CentOS 8
- Ubuntu 20.04 LTS
- ライセンスサーバソフトウェアがバージョン 1.2024010.0 以降の場合、Ubuntu 22.04 LTS

システムソフトウェア要件

ライセンスサーバとユーティリティを使用するには次のソフトウェアがインストールされ、必要に応じて実行されなければなりません：

- 一般的な Linux ユーティリティ: `bash`, `tar`, `sed`, `getopt`, `uname`, `sleep` および `grep`
- `systemd` Linux service manager
- Python 3.6 以上。ライセンスサーバソフトウェアがバージョン 1.2023060.0 以前の場合 PyYAML モジュールをインストールする必要があります。
- 以下のうちいずれかの Java Virtual Machine (JVM) implementation :
 - Oracle Java SE 8
 - OpenJDK 8
 - OpenJDK 11

4.2 ライセンスサーバのインストール

ライセンスサーバをインストールするには、次の手順を行います。

始める前に

- Arm user-based licensing portal <https://developer.arm.com/support/licensing/user-based> にアカウントを持っていない必要があります。
 - user-based licensing portal のアカウントは有効なライセンスを伴う 1 つ以上の製品を保有していない必要があります。より詳細については [製品とライセンスシートの一覧表示](#) の章を参照してください。
 - アカウントが要求される製品を保有していない場合、製品を追加しなければなりません。より詳細については [製品とライセンスの追加](#) の章を参照してください。
- User-based ライセンスモデルでライセンスされた Arm 製品のライセンスサーバを過去にインストールしている場合は、最初にそのサーバをアンインストールする必要があります。
 - ライセンスサーバをアップグレードするとともに既存のライセンスを保持したい場合、既存のすべてのライセンスサーバデータを保存する必要があります。詳しくは、[ライセンスサーバのアップグレード](#) を参照してください。
 - デバイスに新しいライセンスサーバをインストールする場合、ライセンスを削除した後に既存のライセンスサーバデータをすべて削除する必要があります。詳しくは [ライセンスサーバの廃止](#) を参照してください。デバイスのすべてのユーザのライセンスデータを削除する必要があります。



セキュリティ上の理由から、Arm はライセンスサーバを `flexnetls` ユーザおよびグループとして実行することをお勧めします。また、このユーザを `root` に変更しないことをお勧めします。別のユーザとグループを使用する必要がある場合は、インストール開始前に作成してください。

- Arm はサーバのパフォーマンスに影響を与えるためライセンスサーバと同じマシンで Arm 開発ツールを実行することを推奨していません。

Arm 開発ツールを導入するマシンの数が限られている場合は、[アクティベーションコード](#)を使用してください。ライセンス取得にアクティベーションコードのみを使用する場合は、ライセンスサーバは必要ありません。

- ローカルライセンスサーバのインストールと設定についての概要は [Local License Server Setup video tutorial](#) (2:23min.) を参照してください。

この操作について

インストールには、次のような操作が行える権限が必要です：

- ライセンスサーバサービスのための新しいグループとユーザを作成

作成されるデフォルトのユーザは `flexnetls` です。このユーザはホストへのアクセスが制限されており、ホームディレクトリを持たないため、ユーザのセキュリティリスクが軽減されます。

- `root` ユーザによるインストールディレクトリの作成(デフォルトは `/opt/flexnetls-armlmd`)

これにより、インストールディレクトリ内のファイルを他のユーザやプロセスが変更できなくなります。`root` ユーザを使用してインストールすると、ライセンスサーバを `systemd` サービスとして登録することもできます。また、ホストの起動およびシャットダウン時にライセンスサーバを自動的に起動やシャットダウンができます。

- データ保存ディレクトリの作成(デフォルトは `/var/opt/flexnetls-armlmd`)
- `systemd` によるライセンスサーバサービスの登録とサービスの開始



- ライセンスサーバを起動または停止するすべてのアクションで `root` 権限が要求されます。
- セキュリティ上最適な方法として、他のすべてのライセンスサーバコマンドは `root` 以外のユーザが実行する必要があります。ライセンスサーバの設定を変更する、またはホストに関する機密情報を開示するすべてのコマンドはライセンスサーバの管理者パスワードが要求されます。

手順

1. ライセンスサーバの Linux インストレーションパッケージを <https://lm.arm.com/downloads> からダウンロードします。
2. コマンドラインから、ダウンロードしたソフトウェアバンドルを含むディレクトリに移動し、次のコマンドを使用して解凍します：

3.

```
tar -xf flexnetls-armlmd-<version>.tar.gz
```

解凍により、インストーラディレクトリ `flexnetls-armlmd-<version>` が作成されます。



`flexnetls-armlmd-<version>` インストールディレクトリは、任意のディスクの場所に配置でき、インストールプロセスの完了後に削除できます。

4. `root` 権限で次のコマンドを実行して、ライセンスサーバをインストールします:

```
sudo [-E] flexnetls-armlmd-<version>/install_license_server
  [--port <port>] [--install-dir <installation_directory>]
  [--data-dir <data_directory>]
```

パラメータ:

- `-E` パラメータは、`root` としてコマンドを実行するときに現在のユーザの環境を保持します。
Java Virtual Machine (JVM) の場所が、現在のユーザの環境で設定されている `JAVA_HOME` 環境変数に依存している場合、こちらが必要になることがあります。
- `<port>` は、ライセンスサーバが `listen` する TCP ネットワークポートを指定します。
`--port` パラメータが指定されていない場合、ポートのデフォルトは `7070` です。



1024 以下のポート番号は `root` として実行するプロセス用に予約されているため、ポート番号としてここでは指定できません。

- `<installation_directory>` は、ライセンスサーバソフトウェアのインストールディレクトリです。
`install-dir` パラメータが指定されていない場合、インストールディレクトリはデフォルトで `/opt/flexnetls-armlmd` になります。
- `<data_directory>` は、ライセンスサーバの状態ファイルとログを保存するために使用されるディレクトリです。
`data-dir` パラメータが指定されていない場合、データディレクトリはデフォルトで `/var/opt/flexnetlsarmlmd` になります。
- `<user>` は、ライセンスサーバを実行するユーザ名です。`user` パラメータが指定されていない場合、ライセンスサーバは `flexnetls` ユーザとして実行されます。
- `<group>` は、ライセンスサーバを実行するグループ名です。`group` パラメータが指定されていない場合、ライセンスサーバは `flexnetls` グループとして実行されます。

ライセンスサーバは、インストールプロセスの完了後に自動的に起動されます。ライセンスサーバは、サーバデバイスの再起動時にも自動的に起動します。

5. ライセンスサーバが実行されていることを確認します。

ライセンスサーバソフトウェアがインストールされると自動的に起動します。サーバが起動しない場合は、インストールに問題があることを示している可能性があります。次のコマンドを使用してサーバのステータスを確認します。

```
armlm_check_server_status
```

サーバが起動している場合は、以下が表示されます：

```
License server running and ready to accept requests at http://<external  
server name or IP address>:7070
```

サーバが起動していない場合は、以下のコマンドを使用して起動します：

```
sudo systemctl start flexnetls-armlmd
```

ライセンスサーバの起動に失敗した場合は、systemctl ログでエラーを確認できます。

たとえば、次の journalctl コマンドを使用して、オペレーティングシステムの前回の起動以降に記録されたすべてのライセンスサーバサービスイベントを出力します：

```
journalctl -u flexnetls-armlmd.service -b
```

結果

ライセンスサーバはインストールされると、自動的に起動します。ライセンスサーバサービスは、オペレーティングシステムで自動的に起動および停止するように設定されています。

インストールディレクトリ内には以下が存在します：

- **bin** には管理ユーティリティが含まれます。
- **etc** にはデータおよびサーバディレクトリへのパスの詳細が記載された `paths` テキストファイルが含まれます。
- **server** にはライセンスサーバデーモンと関連する設定ファイルが含まれます。
- **license_terms** には software license agreement が含まれます。
- **VERSION** テキストファイルは、ライセンスサーバのバージョンの詳細を示します。

次のステップ

[ライセンスサーバの設定](#)

4.3 ライセンスサーバの設定

ライセンスサーバのインストールが完了したら、サーバを設定する必要があります。

始める前に

- [ライセンスサーバのインストール](#)

手順

1. ライセンスサーバデバイスの PATH 環境変数を設定して、ライセンスサーバのインストールディレクトリに bin ディレクトリを含めます。次に例を示します。

```
export PATH=/opt/flexnetls-armlmd/bin:$PATH
```

2. デフォルトのインストール済み管理者ユーザ(admin) パスワードを使用して管理タスクを実行することはできないため、パスワードを変更する必要があります。
パスワードを変更するには `armlm_change_admin_password` コマンドを実行します。



管理者パスワードはリセットすることが難しいため、失くさないように注意してください。
詳細については、[管理者パスワードのリセット](#) の章を参照してください。

新しいパスワードは次の条件を満たす必要があります：

- 8～64 文字
- 1 桁以上
- 少なくとも 1 つの大文字
- 少なくとも 1 つの特殊文字 (^*\$-+?_&=!%{ }/#@ など)
- 空白文字なし

パスワード変更ツールは、新しいパスワードをもう一度入力して確認するように求めます。管理パスワードが正常に変更されると、以下が出力されます：

```
Administrative account password changed successfully
```

3. ライセンスサーバをインストールすると、インストールプロセスによって `hostid` が選択されます。ホストのハードウェア構成によっては複数の `hostid` が存在することがあります。選択した `hostid` が適切であることを確認する必要があります。たとえば、最も安定した `hostid` が使用されるようにしなければなりません。



- ライセンスサーバ用に生成されたライセンスは、ライセンスサーバの `hostid` にロックされます。ライセンスサーバの登録後にライセンスサーバ `hostid` を変更することはできません。
- ライセンスサーバの `hostid` は、固定されており安定している必要があります。たとえば、ライセンスサーバホストの再起動時や有線ネットワークとワイヤレスネットワークの切り替えの時に `hostid` を変更してはなりません。

次のコマンドを使用して、選択した hostid を確認します：

```
armlm_show_hostid
```

admin のパスワードが要求されます。出力データには、選択した hostid と使用可能な hostid が表示されます。

以下に例を示します。

```
{
  "selected" : {
    "hostidType" : "ETHERNET",
    "hostidValue" : "0800270AA6FF"
  },
  "hostids" : [ {
    "hostidType" : "ETHERNET",
    "hostidValue" : "0800270AA6FF"
  }, {
    "hostidType" : "ETHERNET",
    "hostidValue" : "080027503FFF"
  } ]
}
```

ライセンスサーバで使用される hostid を変更したい場合：

- a) <installation_directory>/server/local-configuration.yaml ファイルを編集します。
- b) 次の行をコメントアウトします：

```
#active-hostid:
```

- c) armlm_show_hostid コマンドで識別される hostid の 1 つを<hostid>/ETHERNET の書式を使用して active-hostid 値として追加します。例：

```
active-hostid: 080027503FFF/ETHERNET
```

- d) 次のコマンドを使用してライセンスサーバを再起動します：

```
sudo systemctl restart flexnetls-armlmd
```

4. インストール時に、ライセンスサーバのポート番号を設定できます。インストール時に設定しない場合、ポート番号はデフォルトの 7070 になります。ポート番号は次の手順で変更できます：

- a) <installation_directory>/server/local-configuration.yaml ファイルを編集します。
- b) port の値を変更します。例：

```
port: 7071
```

- c) 次のコマンドを使用してライセンスサーバを再起動します：

```
sudo systemctl restart flexnetls-armlmd
```

次のステップ

[ライセンスサーバの登録](#)

4.4 ライセンスサーバの登録

Arm licensing portal にライセンスサーバを登録する必要があります。

始める前に

- [ライセンスサーバの設定](#)

手順

1. 以下のコマンドを使用してライセンスサーバが実行されていることを確認します:

```
armlm_check_server_status
```

サーバが起動していない場合は、以下のコマンドを使用して起動します:

```
sudo systemctl start flexnetls-armlmd
```



ライセンスサーバの起動に失敗した場合は、systemctl ログでエラーを確認できます。
たとえば、次の `journalctl` コマンドを使用して、オペレーティングシステムの前回の起動以降に記録されたすべての `flexnetls-armlmd` サービスイベントを出力します:

```
journalctl -u flexnetls-armlmd.service -b
```

2. 次のコマンドを使用して、現在のディレクトリにライセンスサーバの `identity` ファイルを作成します:

```
armlm_generate_server_identity --identity-file identity.bin
```

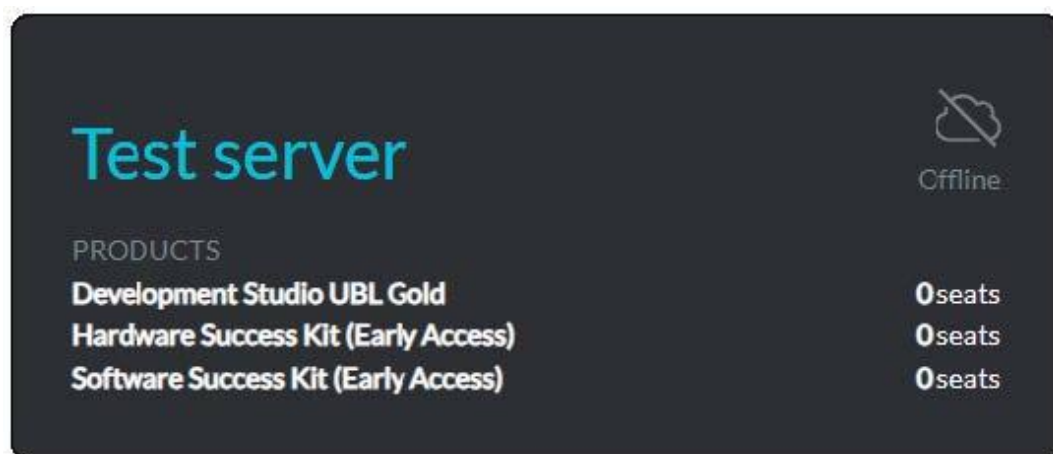
`admin` のパスワードが要求されます。

3. ライセンスサーバを Arm に登録します:
 - a) Arm user-based licensing portal <https://developer.arm.com/support/licensing/user-based> にアクセスします。
 - b) **Servers** タブをクリックします。
 - c) **Register a server** をクリックして、新しいダイアログボックスで以下を行います:
 1. **Server name** のフィールドにサーバ名を入力します。
 2. **Choose or drag & drop your identity file here** と書かれた箇所にファイルをドラッグするか領域をクリックしてファイルを選択することで、生成した `identity.bin` を追加します。
 3. **Upload file & register server** をクリックします。

結果

しばらくすると **Server** タブに戻ります。新しいライセンスサーバのタイルが作成されます。例:

図 4-1: ライセンスサーバタイトルの例



次のステップ

[ライセンスサーバへの製品の割り当て](#)

4.5 ライセンスサーバへの製品の割り当て

ライセンスサーバを登録したら、製品ライセンスをサーバに割り当てます。

始める前に

- [ライセンスサーバの登録](#)を行っている必要があります。
- ライセンスサーバに管理者アクセスが出来なければなりません。
- 既存の有効なアクティベーションコードに関連付けられたライセンスを利用可能にするには、アクティベーションコードを無効化する必要があります。アクティベーションコードライセンスが最近使用されている場合、そのライセンスはすぐにはライセンスサーバに追加できません。詳細については、[アクティベーションコードの無効化](#) の章を参照してください。

手順

1. Arm user-based licensing portal <https://developer.arm.com/support/licensing/user-based> にアクセスします。
2. **Servers** タブをクリックします。
3. ライセンスサーバタイトル上でクリックします。所有している製品について **Allocatable seats** (割り当て可能なシート数)を含むサーバの詳細がページに表示されます。例:

図 4-2: ライセンスサーバに表示される製品情報の例

Hardware Success Kit (Early Access)

ORDER ID	START DATE	EXPIRY DATE	ALLOCATABLE SEATS
UAT_ORDER_048	Jan 17, 2024	Feb 29, 2024	2 (incl. 1 overdraft)

Allocated seats 0/2 - 0 +

- 製品情報の + アイコンをクリックして、ライセンスを製品に追加します。
- 必要な数のライセンスを追加したら **Confirm changes** をクリックします。**Confirm changes** ダイアログボックスに変更されたライセンス数が表示されます。



Warning

ライセンスサーバの管理者権限がない場合は、変更を確定しないでください。ライセンスサーバにアクセスできない状態で変更を確定すると、ライセンスがロックされます。ライセンスをサーバに追加した後、ライセンスサーバから削除して解放することはできないため、ライセンスをアカウントに戻すことができません。

ライセンスがロックされた場合、ロックを解除するには、<https://services.arm.com/support/s/contactsupport> で Support Case を作成して送信する必要があります。

変更が正しい場合は **Confirm changes** をクリックします。

変更が正常に適用されたことを示すバナーが表示されます。

- Manage Server** をクリックし、**Download licenses** を選択することでライセンスをダウンロードします。以下のファイル名でデフォルトのダウンロードフォルダにダウンロードが行われます：

`licenses-<server_ID>-<timestamp>.bin`

ここでの<server_ID> はサーバの識別子であり、<timestamp>はライセンスファイルが作成された日時です。

- 必要に応じて、ライセンスサーバデバイスに製品ライセンスファイルを転送します。
- ライセンスサーバデバイスで次のコマンドを使用して製品ライセンスをロードします：

```
armlm_update_licenses --data-file <license_file>
```

ここでの<license_file> は製品ライセンスファイルの名前です。admin のパスワードが要求されます。

以下の出力が行われれば、それ以上の操作が不要であることを示します：

```
Licenses have been successfully updated. No confirmation is required.
```

次のステップ

[ユーザライセンスのアクティベーション](#)

関連情報

[アクセスコントロールリストの作成](#)

[ライセンスと使用状況の一覧](#)

[ライセンス数の変更](#)

4.6 ユーザライセンスのアクティベーション

ライセンスサーバの登録が完了したら、Arm 製品のライセンスをアクティベートする方法をユーザに通知する必要があります。この場合、ユーザは Arm 製品を使用する人または自動プロセスでも構いません。ユーザは User-based Licensing User Guide 内“**ライセンスサーバを使用した製品のアクティベーション**”の章の手順に従って開発ツールのライセンスを取得することができます。

状況によっては、管理者またはユーザが別の方法でアクティベーションコードをアクティベートする必要があります：

- 以下の場合、代理アクティベーションが使用できます：
 - インターネットにアクセスできないユーザ
 - 7 日間を超えるライセンスアクティベーション期間を必要とするユーザ
 - 同一ユーザに対して複数のデバイスでライセンスのアクティベートが必要なワークフロー。Continuous integration(CI) は、このタイプのワークフローの一例にあたります。
- ユーザが以下のデバイスで Arm 開発ツールを実行している場合は、ローカルライセンスキャッシュのディレクトリを変更する必要があります：
 - ローカルドライブがない
 - ユーザのホームディレクトリがない
 - ユーザのローカルドライブに十分なディスク容量がない

アクティベートを行った後、製品ライセンスは少なくとも 7 日間ユーザにバインドされます。その後、同じデバイスで User-base のライセンスをサポートする Arm 開発ツールを使用すると、Arm ライセンスサーバに接続してライセンスが続く 7 日間更新されます。製品ライセンスを延長できない場合でも 7 日間の期限までの間はそのままライセンスを製品に使用できます。この状況はデバイスがライセンスサーバに接続できない場合に発生する可能性があります。

ライセンスサーバ URL

製品ライセンスをアクティベートするためにユーザに通知するサーバ URL は以下に従っている必要があります：

- `http://` または `https://` のプロトコルプレフィックスを含みます。
- ポート番号がデフォルトのポート番号でない限り、ポート番号を含めます。デフォルトのポート番号は、`http://` の場合は 80、`https://` の場合は 443 です。



これは、ユーザがアクセスする必要がある唯一のライセンスサーバポートです。

- アドレスまたはポート番号の後にパスを含みません。たとえば、`http://myserver:5999` は有効なサーバ URL ですが、`http://myserver:5999/api/1.0/instance/~` は無効です。

環境変数を使用したライセンスング



短期間に多数のデバイスからライセンスアクティベーションがリクエストされる場合、環境変数を使用したライセンス管理は適していません。例えば、Continuous Integration (CI) ワークフローではこのような状況が発生する可能性があります。このようなワークフローでは、ライセンスサーバに大量のリクエストが殺到し、サーバがライセンスリクエストを拒否する可能性があります。このようなワークフローでは代理アクティベーションを使用してライセンスを一度アクティベートし、生成されたライセンスファイルをすべてのデバイスにインポートする必要があります。

ユーザは `ARMLM_ONDEMAND_ACTIVATION` 環境変数に `<product_code>@<server_URL>` を設定することで、ライセンスサーバから製品ライセンスを取得できます。

この環境変数を設定する前に、ユーザは以下のコマンドを使用して、ライセンスサーバ URL が有効であり、必要な製品ライセンスが利用可能であることを確認する必要があります。

```
armlm inspect --server <license server URL>
```

サーバに接続できた場合、このコマンドはライセンスサーバで使用可能な各製品のライセンスと、デバイスでユーザが使用中のライセンスの情報を返します。

複数デバイスでのライセンスのアクティベーション

ユーザに割り当てられたライセンスは、複数のデバイスで使用できます。以下のいずれかの方法でユーザに割り当てられたライセンスを別のデバイスで使えるようにできます：

- 通常どおり、他のデバイスで Arm 開発ツールのライセンスをアクティベートできます。
- 代理アクティベーションでは、1 つのデバイス上でライセンスアクティベーションファイルを作成し、そのファイルを使用して 1 つまたはそれ以上の他のデバイス上で同じユーザにライセンスを提供できます。詳細については、User-based Licensing User Guide 内“代理アクティベーション”の章を参照してください。
- キャッシュされた Arm 製品ライセンスの詳細は、`.armlm` ディレクトリに保存されます。このライセンスキャッシュディレクトリを使用すると同じユーザが他のデバイスでライセンスを利用できるようになります。詳細については User-based Licensing User Guide 内“複数デバイスでのライセンスのアクティベーション”の章を参照してください。

4.7 ライセンスサーバのリリース履歴

次の表は、機能変更が行われた User-based ライセンスのライセンスサーバのバージョンを示しています。ライセンスサーバのバージョンを確認するには[ライセンスサーババージョンの確認](#) の章を参照してください。ライセンスサーバを最新バージョンに更新する方法の詳細については[ライセンスサーバのアップグレード](#) の章を参照してください。

表 4-1: User-based ライセンシングユーティリティの変更

ライセンスサーババージョン	用法
1.2022001.0 (document version 1.2022010)	初回プロダクトリリース
1.2023060.0 (document version 1.2023600)	ライセンスサーバが一定時間内に処理するライセンスリクエストの数を変更するための設定を追加。詳細については、 ライセンスサーバリクエストの最大数の変更 の章を参照してください。
1.2024010.0 (document version 1.2024100)	• Ubuntu 22.04 LTS をサポート • ユーザ名の大文字と小文字の区別を無効化 • PyYAML の要求事項を解除
1.2024050.0 (document version 1.2024500)	アクセスコントロールリストを使用すると、特定のユーザに対して User-based ライセンスのライセンスサーバへのアクセスを許可または拒否できます。詳細については アクセスコントロールリストの作成 の章を参照してください。

5. ライセンスサーバの管理

提供されているライセンスサーバコマンドなど、Arm User-based ライセンシングのライセンスサーバを管理する方法について説明します。

5.1 ライセンスサーバの停止

ライセンスサーバはインストール後およびサーバデバイスの再起動時に自動的に起動されます。

手順

以下のコマンドでライセンスサーバを停止します：

```
sudo systemctl stop flexnetls-armlmd
```

関連情報

[ライセンスサーバの状態確認](#)

[ライセンスサーバの起動](#)

5.2 ライセンスサーバの起動

ライセンスサーバはインストール後およびサーバデバイスの再起動時に自動的に起動されます。

始める前に

ライセンスサーバは管理者によってすでに[停止状態](#)にされています。

手順

以下のコマンドでライセンスサーバを起動します：

```
sudo systemctl start flexnetls-armlmd
```



ライセンスサーバの起動に失敗した場合は、systemctl ログでエラーを確認できます。

たとえば、次の journalctl コマンドを使用して、オペレーティングシステムの前回の起動以降に記録されたすべての flexnetls-armlmd サービスイベントを出力します：

```
journalctl -u flexnetls-armlmd.service -b
```

関連情報

[ライセンスサーバの状態確認](#)

5.3 ライセンスサーバの再起動

ライセンスサーバの再起動には以下の手順を行います。

始める前に

ライセンスサーバは実行中でなければなりません。

手順

以下のコマンドでライセンスサーバを再起動します：

```
sudo systemctl restart flexnetls-armlmd
```



ライセンスサーバの起動に失敗した場合は、systemctl ログでエラーを確認できます。

たとえば、次の journalctl コマンドを使用して、オペレーティングシステムの前回の起動以降に記録されたすべての flexnetls-armlmd サービスイベントを出力します：

```
journalctl -u flexnetls-armlmd.service -b
```

関連情報

[ライセンスサーバの状態確認](#)

5.4 ライセンスサーバの状態確認

ライセンスサーバの状態を確認するには以下の手順を行います。

手順

以下のコマンドでライセンスサーバの状態を確認します：

```
armlm_check_server_status
```

関連情報

[ライセンスサーバの再起動](#)

[ライセンスサーバの停止](#)

[ライセンスサーバの起動](#)

5.5 ライセンスと使用状況の一覧

ライセンスサーバの上の製品ライセンスとその使用状況を一覧表示するには以下の手順を行います。

始める前に

ライセンスサーバは実行中でなければなりません。

手順

以下のコマンドでライセンスサーバのライセンスを一覧表示します：

```
armlm_list_products [--show-expired | -e] [--utc]
```

ここでの：

- **--show-expired** および **-e** はオプションのパラメータです。デフォルトのこのコマンド出力は以下となります：
 - アクティブな製品
 - 直近 30 日以内に期限が切れた製品**--show-expired** または **-e** パラメータを指定した場合、出力には期限切れの製品もすべて表示されます。
- **--utc** はオプションのパラメータです。デフォルトでは出力にはローカルタイムゾーンの時間が表示されます。このパラメータを指定すると出力は UTC の時間が表示されます。

結果

以下はこのコマンドの出力の例です：

```
3 product(s) found on license server
Hardware Success Kit Standard, HWSKT-STD0, 60 seat(s), 33 seat(s) used
Order 273591004, valid until: 2023-02-24, 50 seat(s), 28 seat(s) used
Order 273591034, valid until: 2023-12-31, 10 seat(s), 5 seat(s) used
Keil MDK Professional, KEMDK-PRO0, 20 seat(s), 12 seat(s) used
Order 273591003, valid until: 2023-02-15, 20 seat(s), 12 seat(s) used
```

関連情報

[ユーザの一覧](#)

5.6 ライセンス数の変更

ライセンスサーバで使用可能なライセンス数を変更するには以下の手順を行います。

始める前に

ライセンスサーバは実行中でなければなりません。

アクティベーションコードライセンスをライセンスサーバでできるようにするには、アクティベーションコードを無効化する必要があります。アクティベーションコードライセンスが直近で使用されている場合、そのライセンス

はすぐにライセンスサーバに追加することはできません。詳細については、[アクティベーションコードの無効化](#) の章を参照してください。

手順

1. ライセンスサーバに割り当てられているライセンス数を Arm user-based licensing portal で以下のように変更します：
 - a) Arm user-based licensing portal <https://developer.arm.com/support/licensing/user-based> にアクセスします。
 - b) **Servers** タブをクリックします。
 - c) 変更を行うライセンスサーバに対するサーバタイトルをクリックします。
 - d) 変更を行う製品に対して+ または- をクリックして、ライセンスの追加または削除を行います。
 - e) **Confirm Changes** をクリックします。**Confirm changes** ダイアログボックスに変更されたライセンス数が表示されます。変更が正しい場合は **Confirm changes** をクリックします。
変更が正常に適用されたことを示すバナーが表示されます。



製品のライセンス数を減らした場合、サーバステータスは **Awaiting synchronization** に変わります。confirmation file のアップロードが完了するまでその製品のライセンス数を変更できません。

- f) **Manage Server** をクリックしたあと、**Download licenses** を選択してライセンスファイルをダウンロードします。以下のファイル名でデフォルトのダウンロードフォルダにファイルがダウンロードされます：
`licenses-<server_ID>-<timestamp>.bin`

ここで<server_ID> はサーバの識別子であり、<timestamp>はライセンスファイルが作成された日時です。

2. ファイルをライセンスサーバデバイスに転送し、以下のライセンスコマンドを使用して、変更されたライセンスデータをロードします：

```
armlm_update_licenses --data-file licenses-<server_ID>-<timestamp>.bin
```

admin のパスワードが要求されます。以下の出力データにより追加のアクションが必要かどうかを判断します：

- ・ライセンスが削除された場合、以下の出力によって追加のアクションが必要であることが示されます：

```
Licenses have been successfully updated. A confirmation file needs to be
generated to complete this process.
Please run the "armlm_generate_server_confirmation" command and upload the
generated file to the licensing portal.
```

- ・ライセンスが追加された場合、以下の出力によって追加のアクションが必要ないことが示されます：

```
Licenses have been successfully updated. No confirmation is required.
```

3. ライセンスが削除されると、確認の操作が必要になります：

- a) 以下のライセンスサーバコマンドを使用して confirmation file を生成します：

```
armlm_generate_server_confirmation --confirmation-file confirmation.bin
```

admin のパスワードが要求されます。コマンドの出力によって confirmation file の名前と場所が確認できます。例：

```
Activation response written to /home/user/confirmation.bin
```

- b) Arm user-based licensing portal <https://developer.arm.com/support/licensing/user-based> にアクセスします。
- c) **Servers** タブを選択し、変更を行うライセンスサーバに対するサーバタイルをクリックします。
- d) **Manage Server** をクリックし、**Upload Confirmation** を選択します。新しいダイアログボックスで以下を行います：
1. **Choose or drag & drop your identity file here** と書かれた箇所にファイルをドラッグするか領域をクリックしてファイルを選択することで、ファイルを追加します。
 2. **Upload confirmation file** をクリックします。
- アップロードが正常に行われたことを示すバナーが表示されます。

関連情報

[ライセンスと使用状況の一覧](#)

5.7 新しい製品バージョンのためのライセンス更新

ライセンスに対する開発ツールの新しいバージョンを Arm がリリースすると、Arm 製品のライセンスは新しい開発ツールの機能に合わせて変更されます。ユーザが開発ツールの新しいバージョンをインストールする場合、ライセンスサーバを最新バージョンのライセンスで更新する必要があります。

始める前に

ライセンスサーバは実行中でなければなりません。

この操作について

ライセンスが更新されていない場合、現在のライセンスで製品がサポートされていないことを示すエラーが表示されることがあります。例：

```
armclang: error: FuSa versions of the Product is not supported with this license
```



ライセンスの更新時にライセンスを追加または削除する必要がある場合は、この手順を使用する代わりに、[ライセンス数の変更](#) を行う必要があります。

手順

1. Arm user-based licensing portal でライセンスをダウンロードします：
 - a) Arm user-based licensing portal <https://developer.arm.com/support/licensing/user-based> にアクセスします。
 - b) **Servers** タブを選択します。
 - c) 変更を行うライセンスサーバに対するサーバタイルをクリックします。
 - d) **Manage Server** をクリックし、**Download licenses** を選択してライセンスファイルをダウンロードします。以下のファイル名でデフォルトのダウンロードフォルダにダウンロードが行われます：
`licenses-<server_ID>-<timestamp>.bin`
ここでの<server_ID> はサーバの識別子であり、<timestamp>はライセンスファイルが作成された日時です。
2. ライセンスサーバデバイスに製品ライセンスファイルを転送します。
3. 次のコマンドを使用してライセンスデータをロードします：

```
armlm_update_licenses --data-file licenses-<server_ID>-<timestamp>.bin
```

admin のパスワードが要求されます。以下の出力によって追加のアクションが必要ないことが示されます：

```
Licenses have been successfully updated. No confirmation is required.
```

次のステップ

ユーザは使用中のライセンスを更新する必要があります：

- 代理アクティベーションライセンスのユーザは代理アクティベーションによってライセンスを認証しなければなりません。
- ライセンスサーバの他のユーザライセンスはライセンスの更新時に自動的にアップデートされます。ライセンスは 24 時間ごとに更新されますが、それよりも早くライセンスを必要とする場合、ユーザはライセンスを手作業で延長することで更新できます。

5.8 新しいライセンスサーバへのライセンスの移行

既存のライセンスサーバから新しいライセンスサーバへのライセンスの移行は以下の手順を行います。

始める前に

ライセンスサーバは実行中でなければなりません。

手順

1. [ライセンスサーバの廃止](#)の手順に従って、ライセンスサーバを廃止します。
2. 新しいライセンスサーバを以下の手順に従って作成します：
 - [ライセンスサーバのインストール](#)
 - [ライセンスサーバの設定](#)
 - [ライセンスサーバの登録](#)

- ・ [ライセンスサーバへの製品の割り当て](#)

次のステップ

- ・ ライセンスサーバの URL を変更した場合は、Arm 開発ツールに新しい URL を入力する必要があることをユーザに通知する必要があります。
 1. ユーザが ARMLM_ONDEMAND_ACTIVATION 環境変数を使用して製品をアクティベートする場合は、新しい URL<existing_product_code>@<new_server_URL> を利用して、ARMLM_ONDEMAND_ACTIVATION の値を更新します。
 2. ユーザは新しい URL を使用して製品をアクティベートしなければなりません。
- ・ ライセンスサーバの URL を変更していない場合、ユーザに影響はありません。ユーザが Arm 開発ツールを起動すると、不正な hostid が原因で、既存のライセンスが破損したと認識されます。開発ツールは自動的にライセンスサーバに接続して、正しい hostid を持つ新しいライセンスを取得します。

5.9 ユーザの一覧

ライセンスサーバ上のユーザおよびユーザに関連づけられた製品を一覧表示するには以下の手順を行います。出力には、ライセンスが更新されなかった場合に他のユーザがライセンスを利用できるようになる日付も含まれます。

始める前に

ライセンスサーバは実行中でなければなりません。

手順

以下のコマンドでライセンスサーバのユーザを一覧表示します：

```
armlm_list_users [--csv] [--utc]
```

ここでの：

- ・ `--csv` はオプションのパラメータです。デフォルトではこのコマンドはテーブルフォーマットで出力します。このパラメータを指定すると、コマンドは出力をカンマ区切り(CSV) フォーマットで出力します
- ・ `--utc` はオプションのパラメータです。デフォルトでは出力にはローカルタイムゾーンの時間が表示されます。このパラメータを指定すると出力は UTC の時間が表示されます。

結果

以下がテーブルフォーマットでのコマンド出力の例です：

```
User Product Code Product Name Last Access Held Until
-----
adlxho HWSKT-STD0 Hardware Success Kit 2022-Apr-19 10:13:19 UTC 2022-May-19
hxyiso HWSKT-STD0 Hardware Success Kit 2022-Apr-14 09:08:38 UTC 2022-May-14
jxycot HWSKT-STD0 Hardware Success Kit 2022-Apr-12 09:58:42 UTC 2022-May-12
jxyche HWSKT-STD0 Hardware Success Kit 2022-Apr-03 00:25:09 UTC 2022-May-03
```

関連情報

[ライセンスと使用状況の一覧](#)

5.10 ライセンスのアクティベートとエラーの追跡

ライセンスのアクティベート、ライセンスのエラー、利用可能なライセンスの数をライセンスサーバのログで追跡できます。

この操作について

製品のアクティベートのリクエストは<data_directory>/flexnetls.log ファイルに記録されます。

<data_directory> は、ライセンスサーバのインストール時に設定されたデータディレクトリです。

インストール時にデータディレクトリが指定されていない場合、ログは、
/var/opt/flexnetls-armlmd/logs/flexnetls.log に保存されます。

手順

ライセンスを追跡するには、<data_directory>/flexnetls.log ログファイルで **request preview** および **count=** 文字列を含む行を検索します。例えば、次のログは、1 人のユーザがライセンスのアクティベートに成功した一方で、もう 1 人のユーザは利用可能なライセンスがないためライセンスのアクティベートに失敗したことを示しています。

```
2024-11-13T12:00:00,000Z INFO - [<user> request preview] Successfully handled
acquire request for feature xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx-x 2025.09 count=1
correlationID=xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxxxxxx
2024-11-13T12:00:01,000Z INFO - [<user> request preview] Successfully handled
acquire request for feature xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx-x 2025.09 count=0
correlationID=xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxxxxxx
```

ここでの<user> はアクティベートを行おうとしたユーザ名です。

これらの行は、以下の情報を提供します：

失敗したライセンスアクティベート

失敗したライセンスアクティベートは、**request preview** と **count=0** を含む行を検索すると確認できます。

成功したライセンスアクティベート

成功したライセンスアクティベートは、**request preview** と **count=<*>** を含む行を検索すると確認できます。ここでの<*> は 1 以上の整数を表すワイルドカードを意味します。

利用可能なライセンスの数

利用可能なライセンスの数を追跡するには、**request preview** と **count=** を含むアクティベートのリクエストに関する行を検索します。**count=** の値は、アクティベートリクエスト前の利用可能なライセンスの数です。

例:

- `count=` の値が 9 の場合、1 つのリクエストが正常に処理された後は 8 つのライセンスが利用可能です。
- `count=` の値が 0 の場合、このリクエストで利用可能なライセンスはありません。

5.11 アクセスコントロールリストの作成

アクセスコントロールリストは、特定のユーザに対して、User-based ライセンスのライセンスサーバに対するアクセスを許可または拒否するために使用されます。ライセンスサーバへのアクセス権を持たないユーザは、ライセンスサーバを使用して製品のアクティベートやライセンスの更新を行うことはできません。

始める前に

- ライセンスサーバのバージョンは 1.2024050.0 以降でなければなりません。それより前のバージョンを使用している場合は、ライセンスサーバをアップグレードする必要があります。
- アクセスコントロールリストの作成方法の概要については、[Access control lists video tutorial](#) (2:11min.) を参照してください。

この操作について

アクセスコントロールリストを使用すると、User-based ライセンシングライセンスへのアクセスを制限できます。

例:

- ライセンスサーバで利用可能なライセンスを、特定のユーザのみに使用させる
- ライセンスサーバ上のライセンスを、特定のユーザのみ使用できなくする。例えば、一部のユーザにアクティベーションコードライセンスを与え、それらのユーザはサーバーライセンスから別のライセンスを取得できないようにしたい場合に有効

手順

1. アクセスコントロールリストファイルを作成します。このファイルの内容は以下のフォーマットを用いる必要があります:

```
model "userAccessList" {
  on hostid(<users>) {
    use "default"
    <list_type>
  }
  on any() {
    <reverse_list_type>
  }
}
```

ここでの:

- `<users>` はカンマ区切りのユーザー一覧です。各ユーザは`<username>/USER` の形式で指定します。
`<username>` は大文字と小文字を区別しません。

- ・ `<list_type>` は次のいずれかでなければなりません:
 - `accept` は、指定された`<users>` によるライセンスサーバへのアクセスを許可する許可リストを作成します。それ以外のすべてのユーザは、ライセンスサーバへのアクセスを拒否されます。
 - `deny` は、指定された`<users>` によるライセンスサーバへのアクセスを拒否する拒否リストを作成します。それ以外のすべてのユーザは、ライセンスサーバにアクセスできます。
- ・ `<reverse_list_type>` は次のいずれかでなければなりません:
 - `<list_type>` が `accept` の場合は、`deny`
 - `<list_type>` が `deny` の場合は、`accept`

アクセスコントロールリストファイルの例は、`<installation_directory>/templates` ディレクトリにあります。`<installation_directory>` はライセンスサーバのインストールディレクトリです。インストール時にインストールディレクトリを変更しなかった場合、このディレクトリはデフォルトで `/opt/flexnetls-armlmd` になります。

次のアクセスコントロールファイルの例では、`jasmine`、`ciuser`、および `anyuser` というユーザのみがライセンスサーバにアクセスして製品のアクティベートや、ライセンス更新が行えます。

```
model "userAccessList" {
  on hostid("jasmine/USER", "ciuser/USER", "anyuser/USER") {
    use "default"
    accept
  }
  on any() {
    deny
  }
}
```

2. 次のコマンドを使用して、アクセスコントロールリストファイルをライセンスサーバにロードします:

```
armlm_set_acl --acl-file <ACLfile>
```

ここでの`<ACLfilename>` はアクセスコントロールリストファイルへのフルパスです。`admin` のパスワードが要求されます。アクティベートを行おうとしたユーザ名です。

結果

アクセスコントロールリストが正常にインポートされると、ライセンスサーバから以下のような応答が返されます:

```
ACL set successfully
```

アクセスコントロールリストファイル内のアクセスコントロール設定は、このライセンスサーバ上のそれまでのアクセスコントロールリスト設定を置き換えます。

アクセスコントロールリストを更新してユーザのアクセスを拒否すると、ユーザはライセンスを更新したり、新しい製品をアクティベートしたりできなくなります。ただし、次の点に注意してください:

- ・ ユーザは、アクティブなライセンスを持つデバイスで引き続き Arm 開発ツールを使用できます。ライセンスはデバイスでライセンスが更新されてから 7 日が経過すると、そのデバイス上で非アクティベートされま

す。代理アクティベーションを使用してアクティベートされた製品の場合、ライセンスは最大 365 日間アクティブにできます。

- ライセンスがアクティブな間は、他のライセンスサーバユーザはそのライセンスを使用できません。[ユーザの一覧](#)を表示し、各ユーザの **Held Until** のレコードを確認することで、ユーザライセンスが解放される日を確認できます。

関連情報

[ユーザアクセスコントロールリストの表示](#)

[ユーザアクセスコントロールリストの削除](#)

5.12 ユーザアクセスコントロールリストの表示

ライセンスサーバで現在使用されているユーザアクセスコントロールリストを表示します。

始める前に

ライセンスサーバは実行中でなければなりません。

この操作について

アクセスコントロールリストは、特定のユーザに対して、User-based ライセンスのライセンスサーバに対するアクセスを許可または拒否するために使用されます。ライセンスサーバへのアクセス権を持たないユーザは、ライセンスサーバを使用して製品のアクティベートやライセンスの更新を行うことはできません。

手順

次のコマンドを使用して、ライセンスサーバのアクセスコントロールリストを表示します：

```
armlm_show_acl [--password <admin_password>]
```

`--password <admin_password>` が指定されていない場合は、`admin` のパスワードが要求されます。

結果

以下はこのコマンドの出力の例です：

```
model "userAccessList" {
  on hostid("jasmine/USER", "ciuser/USER", "anyuser/USER") {
    use "default"
    accept
  }
  on any() {
    deny
  }
}
```

この例では、jasmine、ciuser、および anyuser というユーザがライセンスサーバにアクセスして製品のアクティベートや、ライセンス更新が行えることを示しています。その他のユーザはライセンスサーバにアクセスできません。

関連情報

[ユーザアクセスコントロールリストの作成](#)

[ユーザアクセスコントロールリストの削除](#)

5.13 ユーザアクセスコントロールリストの削除

ライセンスサーバのユーザアクセスコントロールリストを削除して、すべてのユーザがライセンスサーバにアクセスできるようにします。

始める前に

ライセンスサーバは実行中でなければなりません。

この操作について

アクセスコントロールリストは、特定のユーザに対して、User-based ライセンスのライセンスサーバに対するアクセスを許可または拒否するために使用されます。ライセンスサーバへのアクセス権を持たないユーザは、ライセンスサーバを使用して製品のアクティベートやライセンスの更新を行うことはできません。

手順

アクセスコントロールリストのデフォルトテンプレートファイルをロードします：

```
armlm_set_acl --acl-file <installation_directory>/templates/default.dsl
```

ここでの<installation_directory> はライセンスサーバのインストールディレクトリです。インストール時にインストールディレクトリを変更していない場合、このディレクトリはデフォルトで/opt/flexnetls-armlmdになります。

admin のパスワードが要求されます。

結果

アクセスコントロールリストが正常にインポートされると、ライセンスサーバから以下のような応答が返されます：

```
ACL set successfully
```

これですべてのユーザがライセンスサーバにアクセスして、Arm 製品をアクティベートしたり、ライセンスを更新したりできるようになります。

関連情報

[ユーザアクセスコントロールリストの表示](#)

[ユーザアクセスコントロールリストの削除](#)

5.14 ライセンスサーババージョンの確認

ライセンスサーバのバージョン番号は、<installation_directory>/VERSION のファイルに記載されています。<installation_directory> はライセンスサーバのインストールディレクトリです。インストール時にインストールディレクトリを変更しなかった場合、このディレクトリはデフォルトで/opt/flexnetls-armlmdになります。

5.15 管理者パスワードの変更

コマンドラインから管理者パスワードの変更ができます。

始める前に

ライセンスサーバは実行中でなければなりません。

手順

1. コマンドラインから、armlm_change_admin_password コマンドを実行します。
2. 既存のパスワードを入力します。



管理者のパスワードを忘れた場合は、[管理者パスワードのリセット](#)の手順を使用してパスワードをリセットできます

3. 新しいパスワードを入力します。新しいパスワードは次の要件を満たす必要があります：
 - ・ 8 ～64 文字
 - ・ 少なくとも1つの数字
 - ・ 少なくとも1つの大文字
 - ・ 少なくとも1つの特殊文字 (例: ^*\$-+?_&=!%{}/#@)
 - ・ スペース文字は使用不可
4. 新しいパスワードを再入力して確認を行います。

結果

管理パスワードが正常に変更されると、以下のような出力が行われます：

```
Administrative account password changed successfully
```

5.16 管理者パスワードのリセット

元のパスワードがわからなくなった場合は、管理者パスワードをリセットします。元のパスワードがわかっている場合は、[管理者パスワードの変更](#)の手順を行います。

始める前に

パスワードのリセットの操作中は、ライセンスサーバを使用できません。サーバをアンインストールしたあとサーバを再インストールし、ライセンスをサーバに再度ロードする必要があるため、時間がかかることがあります。この手順を適切にスケジュールし、ライセンスサーバが使用不可になる期間をユーザに知らせておかなければなりません。

手順

1. 最新のライセンスデータを含むファイルを作成します。
 - a) Arm user-based licensing portal <https://developer.arm.com/support/licensing/user-based> にアクセスします。
 - b) **Servers** タブを選択し、変更を行うライセンスサーバに対するサーバタイルをクリックします。
 - c) **Manage Server** をクリックし、**Download licenses** を選択してライセンスファイルをダウンロードします。以下のファイル名でデフォルトのダウンロードフォルダにダウンロードが行われます：


```
licenses-<server_ID>-<timestamp>.bin
```

 ここで<server_ID> はサーバの識別子であり、<timestamp>はライセンスファイルが作成された日時です。
 - d) ライセンスファイルがライセンスサーバデバイスにダウンロードされなかった場合は、そのデバイスにファイルを転送します。



ライセンスデータファイルは、ライセンスサーバデバイスの以下の両ディレクトリに保存しないようにしてください。これらのディレクトリは、ライセンスサーバのアンインストール時に削除されます：

- ・ ライセンスサーバのインストールディレクトリ(デフォルトは/opt/flexnetls-armlmd)
- ・ データストレージディレクトリ(デフォルトは/var/opt/flexnetls-armlmd)

2. <installation_directory>/server/local-configuration.yaml ファイルをバックアップします。ここで<installation_directory> はライセンスサーバのインストールディレクトリです。インストール時にインストールディレクトリを変更していない場合、このディレクトリはデフォルトで /opt/flexnetls-armlmd になります。
3. [ライセンスサーバのアンインストール](#)の手順に従ってライセンスサーバをアンインストールします。すべてのライセンスデータを削除するには、--delete-storage パラメータを使用する必要があります。
4. [ライセンスサーバのインストール](#)の手順に従ってライセンスサーバを再インストールします。この手順において新しい管理者パスワードを作成します。
5. <installation_directory>/server/local-configuration.yaml ファイルをバックアップコピーで上書きし、[ライセンスサーバを再起動](#)します。

6. 次のコマンドを使用してライセンスデータをロードします：

```
armlm_update_licenses --data-file <license_file>
```

ここでの<license_file> はライセンスファイル名です。admin のパスワードが要求されます。

以下の出力が行われれば、それ以上の操作が不要であることを示します：

```
Licenses have been successfully updated. No confirmation is required.
```

5.17 ライセンスサーバのモニタリング

Arm user-based ライセンシングではライセンスサーバのログファイルは、<data_directory>/logs ディレクトリに保存されます。ここでの<data_directory> は、ライセンスサーバのインストール時に設定されたデータディレクトリです。

インストール時にデータディレクトリが指定されなかった場合、ログは/var/opt/flexnetls-armlmd/logs に保存されます。

5.18 ライセンスサーバの廃止

ライセンスサーバが不要になったら、ライセンスサーバデバイスと Arm user-based licensing portal で製品ライセンスを削除する必要があります。この手順により、Arm 製品のライセンスが他のサーバで使用できるようになります。

始める前に

- ライセンスサーバは実行中でなければなりません。
- 削除するライセンスサーバにアクセスする Arm ユーザが引き続き Arm 製品を必要とする場合は、代替のライセンスサーバを設定する必要があります。詳しくは、[ライセンスサーバのインストール](#)を参照してください。
- ライセンスサーバを廃止する手順の概要については、[User-based Licensing: Removal of Licenses and Decommissioning Server video tutorial](#) を参照してください。

手順

1. ライセンスサーバ上の製品ライセンスを確認します。詳しくは、[ライセンスと使用状況の一覧](#)を参照してください。
2. ライセンスサーバからすべての Arm 製品ライセンスを削除します。これには、サーバ上のすべての製品に関する Arm user-based licensing portal へアップロードされた confirmation file も含まれます。詳細については、[ライセンス数の変更](#) の章を参照してください。
3. Arm user-based licensing portal で以下の手順でライセンスサーバを削除します：
 - a) Arm user-based licensing portal <https://developer.arm.com/support/licensing/user-based> にアクセスします。
 - b) **Servers** をクリックします。
 - c) **Manage Server** をクリックし、**Obsolate Server** を選択します。

- d) 表示されたダイアログボックスで Arm user-based licensing portal からサーバを削除するには **Confirm** をクリックします。

4. [ライセンスサーバのアンインストール](#)の手順に従って、ライセンスサーバをアンインストールします。

5.19 ライセンスサーバの復旧

ライセンスサーバのサーバ管理者が退職したなどの理由でライセンスサーバにアクセスできなくなった場合、ライセンスサーバが利用できなくなる可能性があります。このような場合は Arm User-based ライセンスのライセンスを Arm user-based licensing portal 上の割り当てにユーザ自身で戻すことができないため、Arm サポートに依頼する必要があります。

始める前に

Arm user-based licensing portal 上で現在アクセスできないライセンスサーバにライセンスを登録する際に使ったアカウントにアクセスできなければなりません。このアカウントにアクセスできない場合は、<https://services.arm.com/support/s/contactsupport> で Arm support case を起こしてアカウントへのアクセスができるようリクエストしてください。

手順

1. ライセンスサーバの名前が正しいことを確認します。
 - a) Arm user-based licensing portal <https://developer.arm.com/support/licensing/user-based> にアクセスします。
 - b) **Products** タブを選択し、情報を確認したい製品のタイルをクリックします。
 - c) **Local Servers** タブを選択すると、**Server name** の列にライセンスサーバ名が一覧表示されます。
2. <https://services.arm.com/support/s/contactsupport> で Arm support case を起こし、サポートでライセンスサーバの廃止を行ってもらうよう要求します。
3. サーバが廃止されると、廃止されたサーバ上にあったライセンスが使用可能になります。
4. 新しいサーバをインストールし、製品にライセンスを割り当てます。詳細については[ライセンスサーバのインストール](#)の章を参照してください。

5.20 ライセンスサーバのアップグレード

User-based のライセンスモデルでライセンスされている Arm 製品の既存のライセンスサーバを新しいバージョンのライセンスサーバにアップグレードするには、次の手順を行います。

始める前に

User-based のライセンスモデルライセンスを提供されている Arm 製品のライセンスサーバを事前にインストールしておかなければなりません。

この操作について

ライセンスマネージャの各バージョンで追加された機能の詳細については、[ライセンスサーバのリリース履歴](#)の章を参照してください。



ライセンスサーバをアップグレードするときは、`delete-storage` オプションを使用しないでください。

手順

1. <https://lm.arm.com/downloads> からライセンスサーバ用 Linux installation package の最新バージョンをダウンロードします。
2. `<installation_directory>/server/local-configuration.yaml` ファイルをバックアップします。ここでの`<installation_directory>` はライセンスサーバのインストールディレクトリです。インストール時にインストールディレクトリを変更していない場合、このディレクトリはデフォルトで `/opt/flexnetls-armlmd` になります。
3. コマンドラインから、以下のいずれにも該当しないディレクトリにディレクトリを変更します：
 - ・ ライセンスサーバのインストールディレクトリ(デフォルトは`/opt/flexnetls-armlmd`)
 - ・ データストレージディレクトリ(デフォルトは`/var/opt/flexnetls-armlmd`)
4. 次のコマンドを使用してライセンスサーバをアンインストールします：

```
sudo <installation_directory>/bin/armlm_uninstall_license_server
```

ここでの`<installation_directory>` はライセンスサーバのインストールディレクトリです。インストール時にインストールディレクトリを変更しなかった場合、このディレクトリはデフォルトで `/opt/flexnetlsarmlmd` になります。



ライセンスサーバをアップグレードするときは、`delete-storage` オプションを使用しないでください。

5. [ライセンスサーバのインストール](#)の手順に従って、アップグレードするライセンスサーバをインストールします。



- ・ 以前のインストールで使用したデータストレージディレクトリをそのまま使用する必要があります。(デフォルトでは`/var/opt/flexnetls-armlmd`)
- ・ 以前のインストールで使用した `hostid` をそのまま使用する必要があります。

6. `<installation_directory>/server/local-configuration.yaml` ファイルをバックアップコピーで上書きし、[ライセンスサーバを再起動](#)します。
7. 既存のライセンスが利用可能であることを確認します。[ライセンスと使用状況の一覧](#) の章を参照してください。

関連情報

[ライセンスサーバのインストール](#)

5.21 ライセンスサーバのアンインストール

ライセンスデータ(オプション)を含むライセンスサーバをアンインストールします。

始める前に



アンインストール後にライセンスサーバを再インストールしない場合は、他のライセンスサーバで使用できるように、Arm 製品ライセンスを復旧する必要があります。この方法について詳しくは、[ライセンスサーバの廃止](#) の章を参照してください。

手順

1. コマンドラインから、以下のいずれにも該当しないディレクトリにディレクトリを変更します：
 - ・ ライセンスサーバのインストールディレクトリ(デフォルトは/opt/flexnetls-armlmd)
 - ・ データストレージディレクトリ(デフォルトは/var/opt/flexnetls-armlmd)
2. 次のコマンドを使用してライセンスサーバをアンインストールします：

```
sudo <installation_directory>/bin/armlm_uninstall_license_server [--deletestorage]
```

ここでは：

- ・ <installation_directory> はライセンスサーバのインストールディレクトリです。インストール時にインストールディレクトリを変更しなかった場合、このディレクトリはデフォルトで /opt/flexnetlsarmlmd になります。
- ・ --delete-storage はオプションのパラメータです。設定するとアンインストールプロセスによって割り当てられたライセンスや管理者パスワード、ライセンスサーバの設定などすべてのライセンスデータも削除されます。



- ・ ライセンスサーバをアップグレードするときは、--delete-storage オプションを使用しないでください。
- ・ [ライセンスサーバを廃止](#) する場合は、--delete-storage を使用してください。

5.22 ライセンスサーバディレクトリの変更

ライセンスサーバのインストールディレクトリとデータディレクトリを移動します。

始める前に

ライセンスサーバは実行中でなければなりません。

この操作について

インストールディレクトリとデータディレクトリを移動する必要がある場合、mv や cp コマンドでは移動できません。ライセンスサーバをアンインストールしてから再インストールしなければなりません。

手順

1. 後で必要になる全てのファイルをバックアップします:

- ・ ライセンスサーバのコンフィギュレーションファイルは
`<installation_directory>/server/localconfiguration.yaml` です。
`<installation_directory>` はライセンスサーバのインストールディレクトリです。インストール時にインストールディレクトリを変更していない場合、このディレクトリはデフォルトで
`/opt/flexnetls-armlmd` になります。
- ・ アクセスコントロールリストを作成した場合は、アクセスコントロールリストの詳細を保存してください。
`armlm_show_acl` コマンドを実行し、出力をファイルにセーブすることで保存できます。例:

```
armlm_show_acl --password myadminpassword > /safe/acl_savedeletestorage]
```

- ・ `<data_storage_directory>/logs` ディレクトリ内の必要なログファイル。
`<data_storage_directory>` はデータストレージディレクトリです。インストール時にデータストレージディレクトリを変更していない場合、このディレクトリはデフォルトで
`/var/opt/flexnetls-armlmd` になります。
`armlm_show_acl` コマンドを実行し、出力をファイルにセーブすることで保存できます。例:

2. [ライセンスサーバを廃止します](#)。
3. [ライセンスサーバのインストールと設定を行います](#)。インストールの操作中にインストールおよびデータのために使用したいディレクトリをデフォルトのインストールディレクトリおよびデフォルトのデータディレクトリとして指定できます。
4. 過去に保存したアクセスコントロールファイルを使用して、[アクセスコントロールリストを作成](#)します。
5. `<installation_directory>/server/local-configuration.yaml` ファイルをバックアップコピーで上書きします。
6. [ライセンスサーバを再起動します](#)。

5.23 ライセンスサーバ要求数の変更

ライセンスサーバのレート制限は、ライセンスサーバが一定時間内に処理するライセンス要求の数です。サーバがレート制限を超える要求を受信した場合、指定された時間分待機してからリトライするようにクライアントに指示するエラーコードを伴って追加のライセンス要求が拒否されます。レート制限を常に超過している場合、ライセンスサーバの応答時間が低下し、クライアントのタイムアウトを超過するとライセンス要求が失敗します。

始める前に

ライセンスサーバは実行中でなければなりません。

この操作について

デフォルトでは、ライセンスサーバのレート制限は 1 秒あたり 10 ライセンスの要求となります。ユーザがライセンス要求を処理する時間に問題がある、またはライセンス要求が拒否されていることが分かっている場合は、以下を行えます:

- ・ ライセンシングの方法を修正します。たとえば、多数のデバイスが短時間でライセンスサーバにアクセスしようとする場合は、1 つのライセンスサーバリクエストのみを要求する transfer file を使用します。詳細については、User-based Licensing User Guide 内”[代理アクティベーション](#)”の章を参照してください。

- ライセンスサーババージョン 1.2022120.2 以降の場合、次の手順で説明する方法でレート制限を変更できます。



- このデフォルト値は変更しないことをお勧めします。1 秒あたり 10 回を超える要求が持続する状況は、予期しない動作を引き起こします。レートリミットを変更する前に、この動作の原因を調査する必要があります。
- 要求の量を増やすと、ライセンスサーバのパフォーマンスの問題が発生する可能性があります。

手順

1. `<installation_directory>/server/local-configuration.yaml` のファイルを開きます。
`<installation_directory>` はライセンスサーバのインストールディレクトリです。インストール時にインストールディレクトリを変更していない場合、このディレクトリはデフォルトで `/opt/flexnetls-armlmd` になります。
2. レートリミットを変更するには、`rate-limit` パラメータを変更します。デフォルトのレートリミットは、1 秒あたり 10 リクエストです。
3. ファイルを保存して終了します。
4. [ライセンスサーバを再起動](#)します。

関連情報

[ライセンスサーバの再起動](#)

ドキュメントに関する情報

凡例

以下のサブセクションでは、Arm のドキュメントで使用される表記規則について説明します。

Glossary

"Arm Glossary"は、Arm のドキュメントで使用されている用語と、それらの用語の定義のリストです。Arm Glossary には、Arm において用いる意味が一般に受け入れられている意味と異なる場合を除き、業界標準の用語は含まれていません。

詳細については、Arm® Glossary を参照してください: developer.arm.com/glossary

表記規則

Arm のドキュメントでは、特定の意味を伝えるために、表記規約を使用します。

表記	用法
斜体	引用
強調文字	メニュー名などのインターフェイス要素 信号名 記述リストの用語(適切な場合)
等幅表記 (monospace)	コマンド、ファイル名、プログラム名、ソースコードなど、キーボードから入力できるテキスト
強調等幅表記 (monospace bold)	サンプルコード外で使用される言語キーワード
下線付き等幅表記 (monospace underline)	コマンドまたはオプションで使用可能な略語。完全なコマンド名またはオプション名ではなく、下線付きのテキストの形式で入力可能
<and>	コードまたはコードフラグメント内に現れるアセンブラ構文内で置き換え可能な用語を囲っています。 例: <code>MRC p15, 0, <Rd>, <CRn>, <CRm>, <Opcode_2></code>
小型英大文字	Arm® Glossary で定義されている特定の技術的意味を持つ用語。 例: IMPLEMENTATION DEFINED, IMPLEMENTATION SPECIFIC, UNKNOWN, および UNPREDICTABLE など
 Caution	推奨事項。これらの推奨事項に従わないと、システムの故障や損傷につながるおそれがあります。
 Warning	システム要件。これらの要件を守らないと、システムの故障や破損につながる可能性があります。
 Danger	システムの要件。これらの要件に従わないと、システムが故障したり破損したりすることが予測されます。
 Note	注意が必要な重要な情報

表記	用法
 Tip	作業をより簡単に、より良好に、またはより迅速に行うことができる有用な tip
 Remember	お読みいただいている情報に関する重要なポイントのリマインダ

役だつ情報

本書には、本製品固有の情報が記載されています。その他の役立つ情報については、次のリソースを参照してください。

Arm のドキュメントへのアクセスは、その機密性によって異なります：

- Non-Confidential なドキュメントは developer.arm.com/documentation で入手可能です。以下の表内のドキュメントリンクは、オンラインドキュメントのページに移動します。
- Confidential なドキュメントは、製品パッケージを通じてライセンスのみ入手可能です。

Arm 製品リソース	Document ID	機密性
User-based Licensing at Arm Developer	—	Non-Confidential
User-based Licensing Migration Guide	109727	Non-Confidential
User-based Licensing User Guide	102516	Non-Confidential
User-based Licensing: Accessing contrl lists video tutorial	—	Non-Confidential
User-based Licensing: Accessing the Arm License Portal video tutorial	—	Non-Confidential
User-based Licensing: Cloud-based Licenses and Activation Codes video tutorial	—	Non-Confidential
User-based Licensing: Local License Server Setup video tutorial	—	Non-Confidential
User-based Licensing: Removal of Licenses and Decommissioning Server video tutorial	—	Non-Confidential