

正規化ルール の 操作

正規化ルールを使用するには、次の役職が必要です。

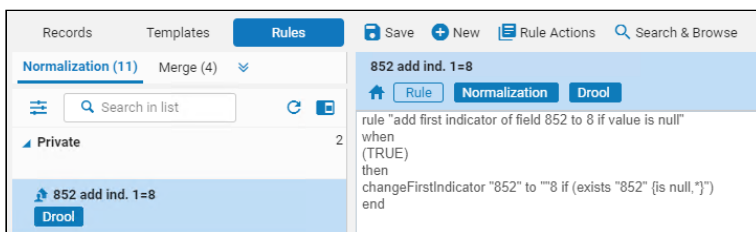
- 目録管理者
- 目録マネージャー
- 目録編集者

正規化ルールは、たとえば、レコードがメタデータエディターに保存されたとき、インポートプロファイルからインポートされたとき、外部検索リソースからインポートされたとき、またはメタデータエディターの [レコードの拡張] メニューで編集されたときなど、さまざまな段階で書誌メタデータを **変更** または **更新** するために使用されます。正規化ルールは特定の構文で記述されているため、レコードへの特定の **変更** に影響を及ぼすルールを作成できます。たとえば、フィールドを編集または削除する際、書誌レコードに含まれる他の条件または含まれない他の条件に基づいて、条件付きで行うことができます。

正規化ルールは、MDエディタの個々のレコードに適用することも、レコードのグループに適用することもできます。

- 正規化ルールを個々のレコードに適用するには、[レコードの強化] オプション ([\[MDエディタメニューとツールバーオプション\]](#) セクションを参照) を使用するか、正規化ルールをプレビューするときに個々のレコードに **変更** を適用します ([ルールの結果をプレビュー](#) を参照)。
- 正規化ルールを一連のレコードに適用するには、rcDroolNormalization、MarcXSLNormalization または DcDroolNormalization タスクを使用してプロセスを作成し ([「正規化プロセスの操作」](#) を参照)、MDエディタで作成する正規化ルールを指定する必要があります (手順 [「新しい正規化ルールファイルを作成する」](#) を参照)。プロセスを作成したら、そのプロセスを使用してジョブを実行できます ([定義済みセットでの手動ジョブの実行](#) を参照)。また、MDエディタで保存をクリックした際に、正規化ルールの適用を可能にするプロセスを作成することもできます ([「正規化プロセスの操作」](#) を参照)。

正規化ルールは、特定のプログラミング構文に従って、MDエディタの [ルール] タブにある編集ウィンドウを使用して作成します。



[ルール] タブの編集ウィンドウ

独自の正規化ルールプログラムを作成するだけでなく、既存のルールを編集ウィンドウにコピーして貼り付けたり、すぐに使用できるサンプル ([ルールタブ]) を使用したりすることで、正規化ルールを作成することもできます。編集ウィンドウにコピーできる正規化ルールと構文サンプルの詳細については、[正規化ルールの構文](#) を参照してください。

MDエディタのプレビュー機能を使用することで次のことができます。

- 正規化ルールとメタデータレコードを並べて表示

- メタデータレコードで実行した場合のルールの結果のプレビュー
- ルールとプレビューの**変更**を切り替え
- ルールを編集してすぐにテストする
- 正規化ルールの概要については、[正規化ルール](#)（6分）をご覧ください。
- 例を含む正規化ルールの詳細については、「[正規化ルール](#)」の動画（1時間）をご覧ください。

正規化ルールの作成

このセクションでは、正規化ルールを作成する方法またはコミュニティゾーンからルールをコピーする方法、以前作成したルールを操作する方法、および単一のメタデータレコードで実行されたルールの結果をプレビューする方法について説明します。詳細については、[こちらのページ](#)の正規化ルール領域を参照してください。

ネットワークゾーンレコードにリンクされた機関ゾーンレコードの正規化ルールは、ローカルフィールドにのみ適用できることに注意してください。正規化ルールをそのようなレコードに適用するには、正規化ルールと正規化プロセスをネットワーク機関でのみ実行します。

ネットワークゾーンコンソーシアムメンバーの場合、ユーザーは新しいルールをローカルに保存するか、ネットワーク機関に保存するかを制御できます。この選択を行うには、[レコード]または[テンプレート]領域を開き、[新規]>[配置オプション]に移動してください。この選択はユーザーごとに行われることに注意してください。

新しい正規化ルールを作成する方法

1. MDエディタページ（[リソース > 目録 > メタデータエディタを開く]）で、[ルール]領域を開きます。
2. MARC または Dublin Core レコードの場合、[新規] > [正規化] を選択して正規化ルールを作成します。[正規化ルールのプロパティ (新しいルール)] というタイトルのダイアログボックスが表示されます。

正規化ルールのプロパティダイアログボックス

3. 正規化ルールファイルの [名前] と [説明] を入力します。

Note

- 。ルール名に「ルール」という単語を使用する場合は、大文字にしないでください。大文字を使用すると、ルールは動作しません。
- 。ルール名にバックスラッシュ (\) を使用しないでください！バックスラッシュを使用した場合、そのルールはフィルターセット機能には使用できません。
- 。ルール名/説明の文字数制限は255文字です。

-
4. MARC レコードの正規化ルールを作成するには、**[Drool]**を選択します。DC (Dublin Core) レコードの正規化ルールを作成するには、**[XSL]**を選択します。

Note

MARCレコード正規化ルールも XSL をサポートしています。

-
5. アクセスオプションを、**[プライベート]** または **[共有]** から選択します。**[プライベート]** を選択した場合、自分だけがルールを操作でき、ルールを正規化プロセスに含めることはできません。**[共有]** を選択すると、ルールは目録者間で共有されます。この場合、複数のユーザーが同時にルールを表示できます。2人以上のユーザーが編集のためにルールを開いている場合、**変更**を保存しようとするすると警告メッセージが表示されます。(変更を保存するか、他のユーザーが変更を行って保存できるようにするかを選択できます。)
6. **[保存]** を選択します。MD エディタの編集ペインが表示されます。既存のルール構文を含めるか (**[編集] > [ルールの追加] > {type of rule}**)、ルールを定義できます (詳細については、[「正規化ルールの構文」](#)を参照)。
7. **保存**を選択します。ルールは、**[正規化ルール]** タブのルールファイルのリストに追加されます。

コミュニティゾーンで正規化ルールをコピー/共有する方法：

1. **[リソース] > [目録] > [メタデータ エディタを開く]** メニューから **[ルール]** タブを選択して、MD エディタ ページの **[正規化]** フォルダを展開します。
2. コピーするには、**[コミュニティ]** フォルダで、コピーしたいルールを右クリックし、**[重複]** を選択します。その **[重複ルール]** ダイアログが開きます。名前と説明を指定し、プライベート (自分だけが利用可能) として、または共有ルール (組織内のすべてのユーザーが利用可能) として保存するかどうかを指定します。そのダイアログボックスには、コミュニティゾーンにルールを投稿したユーザーの Alma ユーザー名と連絡先メールアドレスが表示されます。ご質問がある場合は、当該ユーザーに問い合わせてください。
3. コミュニティゾーンに独自のルールを投稿するには、ルールを右クリックして **[CZに投稿する]** を選択します。ルールの共有ダイアログが開きます。摘要名と説明を入力し、**保存**を選択してコミュニティゾーンにルールを保存します。

既存の正規化ルールを使用する方法：

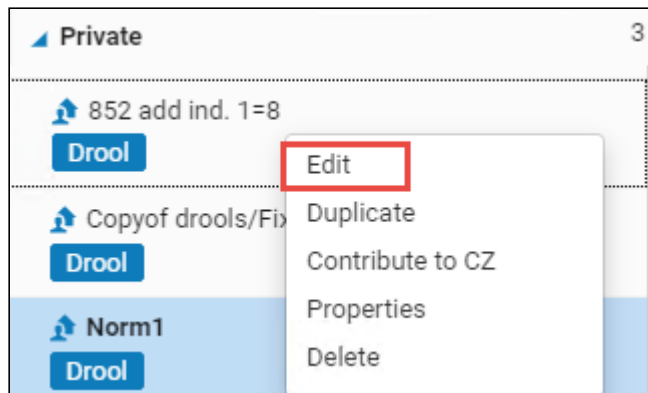
1. **[MD エディタ]** ページ (**[リソース] > [目録] > [メタデータ エディタを開く]**) で**[ルール]** タブを選択し、**[正規化]** フォルダを展開して、保存されているルールを表示します。
2. 使用するルールを選択し、次のオプションのいずれかから選択します。
 - 。 **[編集]** - ルール構文が含まれるテキストボックスを開き、この構文を**変更**できるようにします (詳細については、[「正規化ルールの構文」](#)を参照してください)。
 - 。 **[削除]** - はいを選択して、ルールファイルの削除を確認します。
 - 。 **[重複]** - 選択したルールファイルを複製し、元のファイルに影響を与えずに新しいルールとして**変更**して保存できるようにします。

- 。 [プロパティ] - 正規化ルールのプロパティダイアログボックスを開き、ルールファイルのプロパティを**変更**できます。

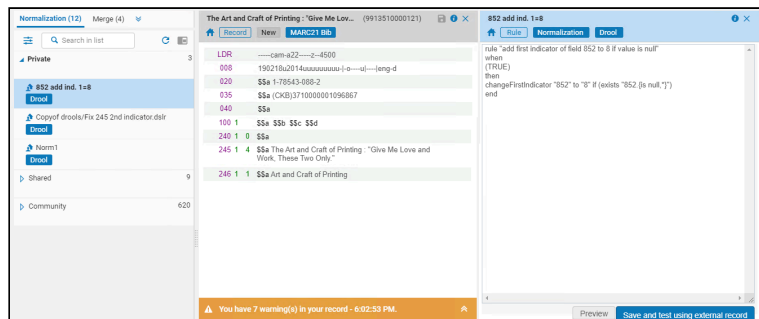
3. ルールの名前を**変更**するには、ルールを複製し、その重複に目的の名前を付けてから、古いルールを削除します。

ルールの結果をプレビューして保存およびテストするには：

1. 作業したい書誌レコードを見つけ（[リポジトリ検索]を使用するか、MDエディタの[レコード]タブ内に検索する）、MDエディタで開きます。
2. **F6** を押して、[スプリット エディタ] アイコンを選択します。
3. 左側のペインで [ルール] タブを選択し、[正規化ルール]フォルダーを展開します。
4. プレビューまたはテストするルールを（[コミュニティ]フォルダーではなく）[プライベート]フォルダまたは [共有] フォルダで右選択し、[編集]を選択します。



ルールはメタデータ エディタの右側のペインに表示されます。



5. [プレビュー] を選択します。ルール がレコードに適用され、結果が表示されます。

Note

ネットワークレコードを更新するには、正規化は（メンバーではなく）ネットワーク機関によって実行される必要があります。機関はネットワークレコードを更新できないため、正規化はローカルフィールド（メンバーのローカル拡張）にのみ適用されます。ネットワークレコードのルールをプレビューすると、次のメッセージが表示されます：正規化プロセスの実行時は、ルールはローカルフィールドにのみ適用されることに注意してください。

- [変更の適用] を選択して変更をレコードに保存するか、[正規化ルールに戻る] を選択します。
- 正規化ルールに最終的な変更を加えたら、[保存して外部レコードを使用してテストする] を選択し、正規化ルールの最終バージョンを保存します。詳細については、[外部データソースの正規化ルールのテスト](#)を参照してください。

Dublin CoreとMODSレコードの正規化ルール

メタデータエディタには、Dublin CoreとMODSに関連する次のタイプのルールがあります。

- XSL表示ルール
- XSL 正規化ルール

Note

XSL 正規化 ルールはDiscovery:localに対応していません。

Dublin Core正規化ルールは、通常の正規化ルール形式での記述はできず、XSL形式でのみの記述が可能です。つまり、メタデータエディタで直接XSLを作成するか（上記で説明）、メモ帳（または任意の外部アプリケーション）で作成し、それをメタデータエディタにコピーすることができます。

メタデータエディタのコミュニティゾーンでは、Dublin CoreのXSL正規化ルールの以下の例を表示できます。

- EXL – dc:language値「en」を「English」に変更
- EXL – Dublin Coreレコードの文字列を置き換える一般的なルール
- EXL_Add_field_accessRights

Dublin CoreとMODSのXSLを記述するには：

1. 選択した外部アプリケーションでXSLを記述します。
2. [メタデータ エディタ]>[ルール] セクションを開きます。
3. メタデータエディターでXSLをコピーして貼り付け、ルールを保存します。

Primo VEの正規化ルールプロセス構成

Almaの正規化ルールに加えて、Primo VEの正規化ルールと Esploroの正規化ルールを作成できます。これらのルールを作成するには、**[新規]**から、次の項目を選択してください。

- **正規化（検出）** – このオプションは、Primo VEがシステムで定義されている場合にのみ表示されます。Primo VEにロードされ、Almaで管理されていない 書誌レコードのDCまたはXML正規化ルールを作成する場合は、このオプションを選択します。これらのルールの構文については、[「DCおよびXML形式の正規化ルールの構文」](#)を参照してください。
- **正規化（リサーチ）** – このオプションは、Esploroがシステムで定義されている場合にのみ表示されます。詳細については、[「アセット正規化ルールの管理」](#)（Esploro）を参照してください。

正規化ルールの構文

ルールには、1つ以上の他のルールを含めることができます。各ルールには、条件とレコードに適用される1つ以上のアクションが含まれています。レコードが条件を満たしている場合、アクションはレコードに適用されます。各アクションは、レコード内の一つのフィールドで実行することができます。アクションはルール内に表示されている順序で実行されます。

複数のルールを含むルールを定義する場合、優先度（または重要度）係数を使用する必要があります。優先度の高いルールで規定されているアクションが最初に実行されます。たとえば、優先度2のルールのアクションは、優先度1のルールのアクションの前に実行されます。正規化ルールでの優先順位（または重要度）の使用法の例については、[Alma正規化ルールの例](#)を参照してください。優先度（または重要度）は、ルールのフィールドが同じである場合にのみ必要であることに注意してください。それ以外の場合、各ルールが異なるフィールドを処理するとき、複数のルールが上から下に行われます。

正規化ルールの作成の詳細については、[正規化ルールの動画](#)（5:42分）をご覧ください。

レコードから指定されたフィールドを削除する正規化ルールを作成する方法やフィールドの内容を変更する方法については、[コンテンツを変更または削除する正規化ルーチン構文](#)の動画で学ぶことができます（9分57秒）。

```
When
(<conditions on MARC record>) then
Action
End
```

<conditions on MARC record>には、レコードに適用される1つ以上のブール句が含まれています。<conditions on MARC record>がTRUEを返す場合、ルールはレコードに適用されます。そうでない場合、ルールは適用されず、レコードは処理されません。

Note

- “When” が最初の行の唯一の単語である必要があります。条件は別の行に配置する必要があります。
- “When”という条件のみで、レコード全体に複数の条件を使うことができます。
- ルールに複数のブール演算子を含めることはできますが、多数のブール演算子を選択すると、パフォーマンスが低下する可能性があります。したがって、ブール演算子は各ルールあたり200個以下に収める必要があります。
- 指定しない場合、条件はレコードレベルで機能します。各MARC21フィールドに個別に条件を適用したい場合は、フィールドごとに条件を指定する必要があります。たとえば、同じタグを持つMARC21フィールドが複数ある場合などです。

正規化ルールにおけるエスケープ文字の理解

正規化ルールを作成する際、引用符、タブ、バックスラッシュなどの特殊文字を含める必要がある場合があります。これを正しく行うには、エスケープ文字を使用します。エスケープ文字とは、システムにこれらの記号をどのように解釈するかを伝える特別な組み合わせです。

有効なエスケープシーケンス

Almaは以下のエスケープシーケンスを認識します。

- `\b` - バックスペース
- `\t` - タブ
- `\n` - 改行
- `\f` - フォームフィード
- `\r` - キャリッジリターン

- \'- ダブルクォート
- \'- シングルクォート
- \- バックスラッシュ

これらのシーケンスは、有効な文字の前にバックスラッシュ1つで始まらなければなりません。

避けるべきよくある間違い

バックスラッシュの数が間違っているとエラーの原因になります。例：

- \zや\|zは無効であり、エラーが発生します。
- 有効なエスケープ文字を伴わない単独のバックスラッシュ (\) も無効です。

エラーの修正方法

次のようなエラーが表示された場合：

「無効なエスケープシーケンスです（有効なものは\b\t\n\f\r\"）」

バックスラッシュを調整することで修正できます。例：

- \\ を \ に置換
- \. を . に置換
- 「\(') を 「(' に、「\()」を 「()」に置き換える
- \\| を \| に置換

これらの変更により、構文エラーを回避しつつルールの意図した動作を維持できます。

レコード要素

条件とアクションは、MARCレコード、フィールド（1つ以上）、インジケータ、サブフィールド（1つ以上）、フィールド/サブフィールドの内容などのレコード要素に適用されます。

条件をテストしたり、レコード要素にアクションを適用したりするには、要素が次の構文に一致する必要があります。

構文

表現	意味
"<tag>"、"<new tag>"	001、245などのフィールドタグを表しています。
"<oldCode>"、 "<newCode>"	a、b、cなどのサブフィールドコードを表しています。
"<element>" データフィールドの場合	データフィールドに指定できる値は次のとおりです <ul style="list-style-type: none"> • FIELD – 例：245 • FIELD_VALUE – 例：245.value* • FIELD_INDICATOR – 例：245.{1,2} • FIELD_SUBFIELD_CODE – 例：245.a

表現	意味
	<ul style="list-style-type: none"> • FIELD_INDICATOR_SUBFIELD_CODE – 例：245.{1,2}.a • FIELD_SUBFIELD_CODE_VALUE – 例：245.a.value* • FIELD_INDICATOR_SUBFIELD_CODE_VALUE – 例：245.{1,2}.a.value*
"<element>" (コントロールフィールド)	<p>以下は、コントロールフィールドに使用できる値です</p> <ul style="list-style-type: none"> • FIELD_POSITION_LENGTH – 例：LDR.{17,3} • FIELD_POSITION_LENGTH_VALUE – 例：LDR.{17,3}.eng
"<element>"固定位置フィールドの場合	<p>以下は、固定位置に使用できる値です UNIMARC/CNMARC 1XXフィールド：</p> <ul style="list-style-type: none"> • FIELD – 例：100 • FIELD_VALUE – 例：100.{*,*}.{*,*}.value* • FIELD_INDICATOR – 例：100.{1,2} • FIELD_SUBFIELD_CODE – 例：100.a • FIELD_INDICATOR_SUBFIELD_CODE – 例：100.{1,2}.a • FIELD_SUBFIELD_CODE_VALUE – 例：100.a.value* • FIELD_INDICATOR_SUBFIELD_CODE_VALUE – 例：100.{1,2}.a.{*,*}.value* • FIELD_POSITION_LENGTH – 例：100.{*,*}.a.{17,3} • FIELD_POSITION_LENGTH_VALUE – 例：100.{*,*}.a.{17,3}.* <p>UNIMARC/CNMARC 1XXフィールドにのみ関連します。</p>
CONDITION (レコードレベル)	<p>可能な条件オプションは次のとおりです。重要な情報については、次のセクション (条件) を参照してください。</p> <ul style="list-style-type: none"> • TRUE – 常にtrue • not exists "{element}" – 要素が存在しない場合はtrue (データフィールドの場合) • not existsControl "{element}" – 要素が存在しない場合はtrue (コントロールフィールドの場合) • exists "{element}" – 要素が少なくとも1つ存在する場合はtrue (データフィールドの場合) • existsControl "{element}" – 要素が少なくとも1つ存在する場合はtrue (コントロールフィールドの場合) • existsMoreThanOnce "{element}" – 要素が複数存在する場合はtrue。この条件の使用例については、MARCレコードの表示規則 - 構文例を参照してください。 • contains – 要素に値が含まれる場合はtrue。統合ルールに使用されます。 <p>以下は、UNIMARC/CNMARC 1XXフィールドにのみ関連する条件オプションです。</p> <ul style="list-style-type: none"> • existsFixedField "{element}" – 要素が存在しない場合は「true」(1XX固定位置フィールドのデータの場合) • not existsFixedField "{element}" – 要素が存在しない場合は「true」(1XX固定位置フィールドのデータの場合)

条件

条件は、ルールレベル全体 (WHEN)、または特定のアクションレベル (IF) で定義できます。同じ条件でも定義されているレベルによって異なる動作をします。

- WHEN句 - ルールがレコードに適用されるかどうかを判断するために、レコード全体で満たされる必要がある条件です
- IF (アクション内) - 特定のアクションがそのフィールドで実行されるかどうかを判断するためにフィールド単体に適用される条件です

条件は次のとおりです

- containsScript - この条件を使用して、特定の言語を検出します。containsScript 条件では、次のチェック可能な言語の固定リストが使用されます。アラビア語、アルメニア語、ベンガル語、注音符號、点字、プヒッド文字、カナダ先住民文字、チェロキー語、キリル文字、デーバナーガリー語、エチオピア語、グルジア語、ギリシャ語、グジャラート語、グルムキー語、ハン語、ハンゲル語、ハヌヌー、ヘブライ語、ひらがな、インヘリタンス、カンナダ語、カタカナ、クメール語、ラオス語、ラテン語、リンブ語、マラヤーラム語、モンゴル語、ミャンマー語、オガム文字、オリヤー語、ルーン語、シンハラ語、シリア語、タガログ語、タグバンワ語、テール、タミル語、テルグ語、ターナ語、タイ語、チベット語、イー語。次の構文例を参照してください

ルール「典?にCJK」

when

```
containsScript "Han" "1**"
```

then

```
set indication."true"
```

end

- exists <element> - 少なくとも1つの一致が見つかりました
 - exists <element> - データフィールドに適用されます。IF句で使用する場合、アクション要素と条件によってテストされる要素の両方が同じ (データ) フィールドでなければなりません。
 - existsControl <element> - コントロールフィールドに適用されます。IF句で使用する場合、アクション要素と条件によってテストされる要素の両方が同じ (コントロール) フィールドでなければなりません。
- existsMoreThanOnce <element> - 複数の一致が見つかりました。データフィールドに適用されます。IF句で使用する場合、アクション要素と条件によってテストされる要素の両方が同じ (データ) フィールドでなければなりません。
- not exists <element> - 一致するものは見つかりませんでした
 - not exists <element> - データフィールドに適用されます。IF句で使用する場合、アクション要素と条件によってテストされる要素の両方が同じ (データ) フィールドでなければなりません。
 - not existsControl <element> - コントロールフィールドに適用されます。IF句で使用する場合、アクション要素と条件によってテストされる要素の両方が同じ (コントロール) フィールドでなければなりません。
- recordHasDuplicateSubfields (表示ルールの場合、[表示ルールの操作](#)を参照) - フィールド、サブフィールド、および無視する文字 (charsToIgnore) として渡された文字列に従って、現在のレコードで重複するサブフィールド (サブフィールドとその内容) が見つかった場合、次の形式のパラメーターでtrueを返します

```
recordHasDuplicateSubfields "<tag>" "<code>" "<charsToIgnore>"
```

コンマで区切られた複数のタグ (フィールド) を指定することができます。複数のコード (サブフィールド) をスペースなしで指定して、それらを区切ることができます。空白を含まない1つ以上の文字 (英数字または句読点) を、

重複の評価対象のサブフィールドのコンテンツの最後で無視する文字として指定できます。詳細については、[例6](#)を参照してください。

recordHasDuplicateSubfields 条件を満たすレコード (trueを返す) の場合、レコードのセットが作成されず。

各IF句のアクションには、次のいずれかの形式を使用することができます

- 特定の条件がtrueでない場合に適用されます。例：`addControlField "{element}" if(not exists "{condition}")`
- 特定の条件がtrueの場合に適用されます。例：`addControlField "{element}" if(exists "{condition}")`
- 無条件に適用されます。例：`addControlField "{element}"`

ブールまたは演算子は、パイプ (|) 記号を使用して、結果ステートメントで使用できます。次に例を示します：

`removeField "866" if (存在しない "866.8.0 | 99 "`。パイプ記号が値の一部である場合は、4つのバックslash (\\\\)を使用してエスケープします。次に例を示します：`removeField "866" if (存在する "866.8.0 \\\\ | 99 ")`

Note

if (exists...) 条件の値がレコード内のどこかにある場合、条件は満たされます。このルールを使用して、特定の値を持つサブフィールドをコピー/削除することはできません。特定の値を持つサブフィールドを削除またはコピーするには、XSL 変換ルールを使用します。

Note

「|」はエスケープされないという既知の問題があります。この点は次のリリースで修正されます。

アクションリスト

次の表は使用可能なアクションのリストです。

アクションリスト

アクション	フォーマット/例	コメント
フィールドとサブフィールドを他のフィールドとサブフィールドに置換	<code>changeControlField "<tag>" to "<new tag>"</code> 例： <code>changeControlField "007" to "008"</code>	コントロールフィールドのタグ識別子を 変更 します。内容を 変更 しません。
	<code>changeField "<tag>" to "<new tag>"</code> 例： <code>changeField "245" to "246"</code>	タグ識別子を 変更 します。この時、インジケータまたはサブフィールドは 変更 しません。
	<code>changeSubField "<tag>.<code>" to "<new code>"</code> <code>changeSubFieldOnlyFirst "<tag>.<code>" to</code>	サブフィールド (最初のサブフィールドのみ、もしくは最初のサブフィールド以外のすべて) の"<code>"を、フィールド"<tag>"のサブフィールド"<new code>"に 変更 します。

アクション	フォーマット/例	コメント
	<pre>"<new code>" changeSubFieldExceptFirst "<tag>.<code>" to "<new code>" 例： changeSubField "035.b" to "a"</pre>	
	<pre>changeFirstIndicator "<tag>" to "<value>" changeSecondIndicator "<tag>" to "<value>" 例： changeFirstIndicator "245" to "3"</pre>	<p>タグ<tag>の指定されたインジケータの値を設定します。</p>
	<pre>combineFields "<tag>" excluding "<comma- separated subfield list>" 例： combineFields "852" excluding "a,b"</pre>	<p>指定した番号のすべてのフィールドを統合します。名前付きサブフィールドを除く、2行目以降のすべてのサブフィールドを最初の行にコピーします。除外されたサブフィールドの最初の出現のみがコピーされ、それらが最初の行にまだ存在しない場合のみ適用できます。</p>
フィールドとサブフィールドの追加	<pre>addField "<tag>.<code>.<value>" addField "<tag>.{<ind1>,<ind2>}.<code>.<value>" 例： addField "999.a.RESTRICTED"</pre>	<p>MARCレコードにフィールドを追加します。サブフィールドの値を指定された値に設定します。</p>
	<pre>addControlField "<tag>.<value>" 例： addControlField "008.820305s1991#####nyu#####001#0#eng##"</pre>	<p>MARCレコードにコントロールフィールドを追加します。</p>
	<pre>addSubField "<tag>.<code>.<value>" addSubField "<tag>.{<ind1>,<ind2>}.<code>.<value>" 例： addSubField "245.h.[Journal]"</pre>	<p>値<value>を持つサブフィールド<code>をフィールド<tag>に追加します。フィールドが存在しない場合、何も行われません。</p>
	<pre>addSystemNumber "<element>" from "<tag>" prefixed by "<prefix tag>" 例： addSystemNumber "035.a" from "001" prefixed by "003"</pre>	<p>データフィールド<element>を、括弧内の2番目のコントロールフィールド<prefix tag>の内容と等しくし、その後最初のコントロールフィールド<tag>の内容を続けます。</p> <p>たとえば、001の値が9945110100121で、003の値がDAVである場合、左側の例の条件は値が# (DAV) 9945110100121の035を生成します。</p>
フィールドをコピー	<pre>copyField "<tag>" to "<new tag>" copyField "<tag>.<code>" to "<new tag>.<new code>"</pre>	<p>フィールドを別のフィールドにコピーします。最初のバージョンでは、サブフィールドは指定されておらず (<code>および<new code>)、新しいフィールドには古いフィールドと同じサブフィールドが含まれています。2番目のバージョンでは、<new code></p>

アクション	フォーマット/例	コメント
	<pre>copyField "<tag>" to "<new tag>.{<ind1>,<ind2>}"</pre> <p>例: copyField "971.a" to "100.u"</p>	<p>が指定されていない場合、新しいサブフィールドは<code>で指定されたサブフィールドと同じです。</p> <p>copyFieldは、既存のフィールドに追加するのではなく、個別のフィールドを作成します。新しいフィールドを既存のフィールドと組み合わせることもできます (combinedFieldsを参照)。</p>
フィールドとサブフィールドを削除	<pre>removeControlField "<tag>"</pre> <p>例: removeControlField "009"</p>	<p>コントロールフィールドのすべての出現を削除します。</p> <p>コントロールフィールド008を削除し再作成しない場合、Almaはすぐに再作成することに注意してください。フィールドを削除した後再度追加することを検討してください。例:</p> <pre>ルール「008を削除」 when (TRUE) then removeControlField "008" addControlField "008.#####s2013####xx#####r#####000#0#eng#d" end</pre>
	<pre>removeField "<tag>"</pre> <p>例: removeField "880"</p>	<p>フィールドのすべての出現を削除します<tag>.</p>
	<pre>removeSubField "<tag>.<code>"</pre> <p>例: removeSubField "245.h"</p>	<p>指定されたフィールドからサブフィールド<code>のすべての出現を削除します。</p>
フィールドまたはサブフィールドのテキストを置換	<pre>replaceControlContents "<tag>.{<position>,<length>}. <value>" with "<new value>"</pre> <p>例: replaceControlContents "LDR.{7,1}.s" with "m"</p>	<p>開始位置<position>の<value> with "<new value>"をコントロールフィールド<tag>の<position>+<length>に置き換えます。<value>に一致するテキストのみを置き換えます。</p>
	<pre>replaceContents "<tag>.<code>.<value>" with "<new value>"</pre> <pre>replaceContentsOnlyFirst "<tag>.<code>.<value>" with "<new value>"</pre> <pre>replaceContentsExceptFirst "<tag>.<code>.<value>" with "<new value>"</pre> <p>例: replaceContents"245.h.[Journal]" with "[Book]"</p>	<p>フィールド"<tag>"のサブフィールド<code>内の<value>と一致する文字列 (もしくは最初が一致するサブフィールドの一致する文字列の全インスタンス、または最初が一致するサブフィールドを除いた、全一致するサブフィールドの全一致する文字列) を"<new value>"に置き換えます。<value>と一致しない文字列または文字列の一部は変更されません。</p>
	<pre>replaceSubFieldContents "<tag>.<code>" with "<tag>.<code>"</pre>	<p>サブフィールドの内容を別のサブフィールドの内容に置き換えます。</p>

アクション	フォーマット/例	コメント
	<p>例: <code>replaceSubFieldContents "245.b" with "100.a"</code></p>	
	<p><code>replaceFixedContents "<tag>.{<1_ind>,<2_ind>}.<code>.{<position>,<length>}.<value>" with "<new value>"</code></p> <p>例: <code>replaceFixedContents "100.{1,2}.a.{0,8}.20150226" with "20220724"</code></p>	<p>UNIMARCおよびCNMARC 1XX固定位置フィールドの<value> with <new value> を置き換えます。</p> <p>UNIMARC/CNMARC 1XXフィールドにのみ関連します。</p>
サブフィールドにテキストを追加する	<p><code>prefix "<tag>.<code>" with "<value>"</code></p> <p>例: <code>prefix "035.b" with "(0CoLC)"</code></p>	<p>フィールド "<tag>"のサブフィールド "<code>" の値に接頭語を追加します。</p> <p>新しい値<value>の後に、古い値が続きます。</p>
	<p><code>prefixSubField "<tag>.<code>" with "<source tag>.<source code>"</code></p> <p>例: <code>prefixSubField "910.a" with "906.a"</code></p>	<p>フィールド "<source tag>"のサブフィールド "<source code>"の値を、サブフィールド "<code>"の接頭語としてフィールド "<tag>"に追加します。</p> <p>新しい値は、フィールド "<source tag>"のサブフィールド "<source code>"の値と続き、その後に古い値が続きます。</p>
	<p><code>suffix "<tag>.<code>" with "<value>"</code></p> <p>例: <code>suffix "035.b" with "(0CoLC)"</code></p>	<p>フィールド "<tag>"のサブフィールド "<code>" の値に接尾語を追加します。</p> <p>新しい値<value>の後に、古い値が続きます。</p>
	<p><code>suffixSubField "<tag>.<code>" with "<source tag>.<source code>"</code></p> <p>例: <code>suffixSubField "910.a" with "907.c"</code></p>	<p>フィールド "<source tag>"のサブフィールド "<source code>"の値を、フィールド "<tag>"のサブフィールド "<code>"の接尾語として追加します。</p> <p>新しい値は、フィールド "<source tag>"のサブフィールド "<source code>"の値と続き、その後に古い値が続きます。</p>
書誌および典拠レコードにおける機関情報の維持 たとえば、この構文は、保存時にネットワークゾーンの書誌	<p><code>addCreatingAgency "<tag>.<code>"</code></p> <p>例: <code>addCreatingAgency "040.a"</code></p>	<p>作成機関のISILコードを、フィールド "<tag>"のサブフィールド "<code>"に追加します。</p>
	<p><code>addModifyingAgency "<tag>.<code>"</code></p> <p>例: <code>addModifyingAgency "040.d"</code></p>	<p>フィールド "<tag>"のサブフィールド "<code>"に変更する機関ISILコードを追加します。"<tag>.<code>"に既に変更する機関がある場合、これによって別の機関ISILコードが追加されます。</p>
	<p><code>replaceModifyingAgency "<tag>.<code>"</code></p> <p>例: <code>replaceModifyingAgency "040.d"</code></p>	<p>フィールド "<tag>"のサブフィールド "<code>"に変更する機関ISILコードを追加します。"<tag>.<code>"に変更する機関が既に存在する場合、すべて置き換えられます。</p>

アクション	フォーマット/例	コメント
<p>レコードを正規化するために、 〔MARC 21書誌メタデータ設定タスクリスト〕で選択される正規化ルールで使用できます。</p> <hr/> <p>Note</p> <p>この機能は現在作業中です。この構文を有効にするには、 Ex Libris サポートにお問い合わせください。</p> <hr/>		
<p>サブフィールドの分割</p>	<pre>splitSubField "<tag>.{ind1,ind2}.<code>.<delimiter>" to "<tag>.<ind1>,<ind2>}.<code>" addSeq "<code>" 例1: splitSubField "866.a.;" to "555.{0,0}.a" addSeq "8" 例2: splitSubField "555.a.- " to "859.{0,0}.a" addSeq "8"</pre>	<p>このタグは必須です。</p> <p>インジケータは任意です。</p> <p>分割はサブフィールドレベルで行われるため、コードは必須です。</p> <p>区切り文字には任意の文字列を使用することができます。区切り文字が存在しない場合、完全なサブフィールドが最初の（そして</p>

アクション	フォーマット/例	コメント
	<p>例3: <code>splitSubField "859.a.\\\\"</code></p> <p>例4: <code>splitSubField "999.a.;" to "555.a" addSeq "8"</code></p>	<p>唯一の) 発生セグメントとしてコピーされ、シーケンスが追加されます。</p> <p>toコンポーネントは任意です。指定されている場合、to tag.codeが複数発生し、それぞれに区切り文字までのデータが含まれます。例1および例2を参照してください。toコンポーネントが指定されていない場合、サブフィールドは例3に示されるように、同じフィールド内の同じ内容の追加のサブフィールドとして分割されます。</p> <p>addSeqコンポーネントは任意です。toコンポーネントが指定されていない場合は関係ありません。addSeqを指定すると、例1のように、シーケンスを持つサブフィールドが追加されます。また、サブフィールドが元のフィールドに既に存在する場合、例2のように、シーケンス (ピリオドが前に付く) がそのフィールドに追加されます。</p>
重複サブフィールドを削除	<p><code>correctDuplicateSubfields "<tag>" "<code>"</code></p> <p>例:</p> <pre> フィールド610および630から重複するサブフィールド x、y、およびzを削除します。 ルール「重複を削除」 priority 1 when (TRUE) then correctDuplicateSubfields "610,630" "xyz" end </pre>	<p>パラメーターとして渡されたフィールドとサブフィールドに従って、最初の出現を保持し、現在のレコード から 他のサブフィールドを削除することにより、重複したサブフィールド (同じコードおよび同じ値のサブフィールドなど) を修正します。</p> <p>recordHasDuplicateSubfieldsを使用して、correctDuplicateSubfieldsを使用する正規化ルールに提供するセットを作成できます。詳細は、例6を参照してください。</p> <p>異なる値を持つサブフィールドの重複排除については、以下を参照してください。</p> <p>MD エディターで同じフィールド内の複数の類似したサブフィールドのうち1つを除いてすべてを削除する方法。</p>
移動サブフィールド	<p><code>moveSubfieldsToEndOfField "<tag>" "<code>"</code></p> <p>例:</p> <pre> サブフィールド9および2をフィールド650の末尾に移動 します。 ルール「サブフィールドをフィールドの末尾に移動」 priority 1 when (TRUE) then moveSubfieldsToEndOfField "650" "92" end </pre>	<p>各サブフィールドの最初の出現をフィールドの末尾に移動させ、他に出現する同じサブフィールドをすべて削除します。</p> <p>複数のサブフィールドが指定されている場合、ルールで識別されていると同じ順序で最後に配置されます。この例では、サブフィールド9が末尾に配置され、その後サブフィールド2が続いています。</p> <p>ifステートメントはmoveSubfieldsToEndOfField アクションではサポートされていないことに注意してください。</p>
現在のレコードの重複するフィールド	<p><code>correctDuplicateFields "{fields}"</code></p> <p>例:</p> <pre> correctDuplicateFields "610,630,650" </pre>	<p>このアクションが取り入れるパラメーターは、610,630,650などのコマ区切りのフィールド値を含むフィールドのみです。</p> <p>このアクションは、パラメーターとして渡されたフィールドに従って、現在のレコードの重複したフィールドを修正します。</p>

アクション	フォーマット/例	コメント
ドを修正する		<hr/> <p>Note</p> <p>このアクションは、同じフィールドの複数のインスタンス間で見つかった重複データを識別して修正するためにのみ使用できます。</p> <hr/>
重複したフィールドを見つける (表示ルールについては 表示ルールの操作 を参照)	<pre>recordHasDuplicateFields "{fields}"</pre> <p>例：</p> <div style="border: 1px solid black; padding: 5px; width: fit-content;"> <pre>recordHasDuplicateFields "610,630,650"</pre> </div>	<p>このアクションが取り入れるパラメーターは、610,630,650などのコンマ区切りのフィールド値を含むフィールドのみです。</p> <p>このアクションの結果は、trueまたはfalseのいずれかです。パラメーターとして渡されたフィールドに従って、現在のレコードに重複したフィールドが見つかった場合、trueを返します。</p> <hr/> <p>Note</p> <p>このアクションは、同じフィールドの複数のインスタンス間で見つかった重複データを識別して修正するためにのみ使用できます。</p> <hr/>

ワイルドカードと特殊文字

Note

Alma正規化ルールでの正規表現の使用は現在完全にはサポートされていませんが、一部の正規表現は使用できます。これらの例については、[Alma正規化ルールの例](#)を参照してください。

サブフィールド区切り文字 (\$\$) は、ルール名を含め、ルール内のどこにも表示することはできません。

行の先頭にあるハッシュ文字 (#) は、行の残りがコメントであることを示し、ルールの処理時には無視されます。

Note

ハッシュ記号 (#) は、正規化ルール関数 `replaceControlContents` によってスペースとして処理されません。ハッシュ記号 (#) はワイルドカードではなく、この関数ではサポートされていません。

アスタリスク (*) は、長さゼロの文字列を含む任意の文字列と一致させるために使用されます。たとえば、"`<tag>.<*>.<value>`"は、値`<value>`を持つタグ`<tag>`内のすべてのサブフィールドに適用されます。*は「貪欲」機能であるため、文字列内のできるだけ多くの文字に一致しようとします。例：文字列が「a b c b d b e」の場合、「b*b」というパターンは「b c b」だけでなく、「b c b d b」も一致します。

空のインジケータ (フィールドまたはサブフィールドは除く) は、ハイフン (-) で示されます。たとえば、"`<tag> {-`

,<ind2>}は、MARCタグが<tag>で、最初のインジケータが定義されておらず、2番目のインジケータが<ind2>であるすべてのフィールドを返します。

サブフィールドのテキストに最後の文字としてピリオドが含まれる場合、4つのバックスラッシュ記号を使用してピリオドに一致させます。例：

```
ルール「$a (unconditional)の'1 v.'を'Leaves'に置換」
when
(TRUE)
then
replaceContents "300.a.1 v\\\\" with "Leaves"
end
```

二重引用符は条件（のみ）で使用できます。条件の一部として二重引用符を使用するには、二重引用符（"）ではなく、単一引用符を使用してルール内のテキストを囲みます（'）。この方法を使用すれば、二重バックスラッシュ（\\）に続くテキスト内で二重引用符を使用できます。

```
rule "populate 008 7-10 2016"
when
(exists '245.{*, }.c.\"')
then
replaceControlContents "008.{7,4}" with "2016"
end
```

ヘブライ語の日付は、条件（のみ）で使用できます。（ヘブライ語は右から左に読み取るため、次の例の二重バックスラッシュは実際には二重引用符の前にあります。）

```
rule "populate 008 7-10 2016"
when
((exists '260.{*, }.c.1"\vwn') OR (exists '264.{*, }.c.1"\vwn'))
then
replaceControlContents "008.{7,4}" with "2016"
end
```

Note

- ワイルドカードは、条件または値の最初の文字として使用することはできません。
- リテラルバックスラッシュ（\）を使用するには、別のバックスラッシュ\\でエスケープします。
- ピリオドが文字列の最後の文字である場合、4つのバックスラッシュ（\\\\"）を使用してエスケープします。ピリオドの直後に別の文字が続く場合、4つのバックスラッシュは必要ありません（次の例のようにaddField "907.a.F.L.T\\\\"）。ただし、一貫性のある望ましい結果を確保するには、正規化ルールで常に4つのバックスラッシュを使用することをお勧めします。次の例を参照してください。
- 上記のように、二重引用符が単一引用符を使用してエスケープする条件（でのみ）使用される場合や、単一引用符を使用してエスケープする条件で単一引用符を使用する場合、二重引用符または引用符をダブルバックスラッシュでさらにエスケープする必要があります。
- パイプ記号が条件の一部である場合は、4つのバックスラッシュを使用してエスケープします。例：
removeField "866" if (exists "866.8.0\\\\"|99")。これは、条件でパイプ記号を使用する場合にのみ

必要です。

例：replaceContentsを使用した正規化ルールでのピリオドの使用

ピリオドを含むレコードの例：

245 00 \$\$a Feminist literary theory.: \$\$b a reader / \$\$c edited by Mary Eagleton.

246 0# \$\$a F.L.T.

上記のサンプルレコードの正規化ルール

ルール「245および246サブフィールドaのピリオドを削除（およびピリオドを何とも置き換えない）ピリオドの前に4つのバックスラッシュを入れる」

```
when
(TRUE)
then
replaceContents "245.a.\\\\\\." with ""
replaceContents "246.a.\\\\\\." with ""
end
```

前後の例については、下の図を参照してください。

040		\$\$a DLC \$\$c DLC	035	0	\$\$a AEW2509
049		\$\$a BXMM	040		\$\$a DLC \$\$c DLC
050	0 0	\$\$a PN98.W64 \$\$b F44 1996	049		\$\$a BXMM
082	0 0	\$\$a 801/95/082 \$\$2 20	050	0 0	\$\$a PN98.W64 \$\$b F44 1996
245	0 0	\$\$a Feminist literary theory : \$\$b a reader / \$\$c edited by Mary Eagleton.	082	0 0	\$\$a 801/95/082 \$\$2 20
246	0	\$\$a F.L.T.	245	0 0	\$\$a Feminist literary theory : \$\$b a reader / \$\$c edited by Mary Eagleton.
250		\$\$a 2nd ed.	246	0	\$\$a FLT
260		\$\$a Cambridge, Mass. : \$\$b Blackwell, \$\$c 1996.	250		\$\$a 2nd ed.
263		\$\$a 9511	260		\$\$a Cambridge, Mass. : \$\$b Blackwell, \$\$c 1996.
			263		\$\$a 9511

前後の例

例：addFieldを使用した正規化ルールでのピリオドの使用

以下は、ピリオドを追加する必要のあるレコードの例です。

906 \$\$a Architecture.

907 \$\$a F.L.T.

以下は、常にスラッシュを含めるというベストプラクティスを使用した上記のサンプルレコードの正規化ルールです：

```
ルール「フィールド906に文字Architectureとその末尾にピリオドを追加、およびフィールド907にF.L.T.も追加」
saliency 100
when
TRUE
```

```
then
addField "906.a.Architecture\\\\\\\\."
addField "907.a.F\\\\\\\\.L\\\\\\\\.T\\\\\\\\."
end
```

レコードの前後の例については、次の図を参照してください。

504		\$\$a Includes bibliographical references (p. [421]-428) and index.	650	0	\$\$a Feminist literary criticism.
650	0	\$\$a Feminist literary criticism.	700	1	\$\$a Eagleton, Mary.
700	1	\$\$a Eagleton, Mary.	948		\$\$a LTI 08/08/2000
948		\$\$a LTI 08/08/2000	950		\$\$a OL
950		\$\$a OL	906		\$\$a Architecture.
			907		\$\$a F.L.T.

前後の例

正規化ルールの例

100 を超える正規化ルールの例とその他の正規化ルールに関する文書のリストについては、[Alma正規化ルールの例](#)を参照してください。

XSL形式の書誌レコードに関する正規化ルールの作成の詳細については、「[Alma XSL 正規化ルールの例](#)」を参照してください。

Note

Alma は XSL バージョン 3.0 までをサポートします。
