



chapter.1 MLD

プロセッサの登録は、JSP カスタムタグの TLD に似た MLD と呼ぶ XML ファイルによって行います。

1.1. プロセッサの定義

プロセッサの Maya エンジンへの組み込みは、MLD (Maya Library Definition) とよぶ定義済み XML を記述する必要があります。MLD は DTD を持つ Maya 独自の XML です。また、拡張子は .mld とします。

1.1.1 MLD ファイル

MLD は List. 1-1 の DTD によって定義されます。

List. 1-1 mld_1_0.tld

```
<?xml version="1.0" encoding="UTF-8"?>

<!NOTATION mld_1_0 PUBLIC
    "-//The Seasar Project//DTD Maya Library Definition 1.0//EN">

<!ELEMENT library (description?, processor*) >
<!ATTLIST library uri CDATA #REQUIRED>

<!ELEMENT description (#PCDATA)>

<!ELEMENT processor (deprecated?, description?, property*)>
<!ATTLIST processor name CDATA #REQUIRED>
<!ATTLIST processor class CDATA #REQUIRED>

<!ELEMENT property (description?)>
<!ATTLIST property name CDATA #REQUIRED>
<!ATTLIST property expectedType CDATA #IMPLIED>
<!ATTLIST property required (true | false) "false">
<!ATTLIST property defaultValue CDATA #IMPLIED>
```

DTD に従った、サンプルの MLD は List. 1-2 の通りです。

List. 1-2 sample.mld

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE library
    PUBLIC "-//The Seasar Foundation//DTD Maya Library Definition 1.0//EN"
    "http://maya.seasar.org/dtd/mld_1_0.dtd">
<library uri="http://maya.seasar.org/sample">
    <processor name="foo" class="org.seasar.maya.sample.processor.SimpleProcessor">
        <property name="greeting" required="true" expectedType="java.lang.String"/>
        <property name="default" defaultValue="こんにちは"/>
    </processor>
</library>
```

Maya は、MLD を起動時にキャッシュして、テンプレートビルド時の必要に応じてその定義情報を利用します。List. 1-2 の MLD を例に MLD の各ノードの意味を解説します。



Maya

– JavaServer Templates –

1.1.2 library ノード

MLD のルートノードである library ノードは、必須の uri 属性を持ちます。この uri 属性は、MLD に属する定義全体に関わる名前空間 URI となります。

```
<library uri="http://maya.seasar.org/sample">
```

サンプルでは「http://maya.seasar.org/sample」という URI を記述しています。テンプレートや.maya 設定 XML ではこの URI をプレフィックスマッピング宣言をして、用いることになります。たとえば.maya 設定 XML 中で用いた場合は、以下のようになります。

```
<maya xmlns:sample="http://maya.seasar.org/sample">
```

「sample」プレフィックスにマッピングしました。Mayaのエンジンがこのマッピング宣言から該当するsample.mld(List. 1-2)を探します。

1.1.3 processor ノード

processor ノードは一つ一つで、テンプレートプロセッサの定義となります。processor ノードはそれぞれ必須の name 属性と class 属性を持ちます。name 属性はプロセッサの名前となり、library ノードの uri 属性に定義した名前空間 URI 中のローカル名となります。

```
<processor name="foo" class="org.seasar.maya.sample.processor.SimpleProcessor">
```

List. 1-2中の上記記述はlibraryノードと併せ、「{ http://www.seasar.org/maya/sample }foo」という名前で、「org.seasar.maya.sample.processor.SimpleProcessor」というJavaクラスであると示しています。

class 属性値としてとれる Java クラス型は、テンプレートプロセッサの API インターフェイスである、org.seasar.maya.engine.processor.TemplateProcessor を実装しているものの他、テンプレートプロセッサのファクトリクラスとして org.seasar.maya.builder.processor.ProcessorFactory を実装したものをとることができます。Maya エンジンは class 属性値の Java クラスの型をしらべ、TemplateProcessor を実装している場合は、直接インスタンス生成を行い、ProcessorFactory を実装している場合には、ProcessorFactory#createProcessor(Template, SpecificationNode)というファクトリメソッドを通じてテンプレートプロセッサを生成します。

子の property ノードは、必須の name 属性で指定された名前のプロパティを processor ノードの class 属性で指定されたテンプレートプロセッサ (TemplateProcessor) もしくはプロセッサのファクトリ (ProcessorFactory) より探し、やはり必須の value 属性に指定された値を設定します。

1.1.4 property ノード

property ノードは、テンプレートや.maya 設定 XML に記述された属性値をバインディングするために、テンプレートプロセッサのプロパティを公開するために用いられます。property ノードは、必須の name 属性と、オプションの expectedType、required、express、value の各属性があります。

name 属性はテンプレートや.maya 設定 XML に記述する属性の名前であると同時に、テンプレートプロセッサの実装における JavaBean プロパティ名となります。下記の場合、「greeting」プロパティ、すなわち setGreeting(...)によってビルド時にエンジンよりテンプレートプロセッサに値が設定されます。

```
<property name="greeting" required="true" expectedType="java.lang.String"/>
```

required 属性は true もしくは false の値をとり、false が属性省略時の値となります。true のときにはこのプロパティが必須であることを示し、テンプレートや.maya 設定 XML に属性設定の記述が欠如している場合、例外となります。ただし、後述の value 属性を記述しているときにはこの限りではありません。

テンプレートプロセッサのプロパティが動的な式言語の結果を利用したい場合には、プロセッサのプロパティ型を org.seasar.maya.engine.processor.ProcessorProperty にします。そうすると、自動的に Maya は内部で ProcessorProperty のインスタンスを生成してプロセッサに引き渡します。

expectedType 属性は、Java クラス型の完全修飾名を値とします。「java.lang.Object」を属性省略時に適用するデフォルトの値とします。動的な属性処理を行う場合には、ProcessorProperty から値取得メソッド (getDynamicValue(PageContext)メソッド) が提供する値の型を決定します。静的な属性の場合には、expectedType 属性に値を設定しても意味をなしません。

defaultValue 属性は、MLD 上でプロセッサのプロパティ値指定を行うための属性です。defaultValue 属性が記述されているときには、インジェクトするノードに該当属性の記述が無い場合に、値が適用されます。該当属性が記述された場合、優先してインジェクションするノードの属性値が用いられます。プロセッサのプロパティが ProcessorProperty 型である動的属性の場合は、defaultValue 属性値に式言語文字列を記述することができます。



Maya

– JavaServer Templates –

1.2.

MLD の配置

MLD はちょうど、JSP カスタムタグにおける TLD のような位置づけです。配置も TLD のように、ファイルのまま配置する方法のほか、Jar アーカイブ中に配置する方法もあります。

1.2.1

ファイルとして配置

MLD を WEB アプリケーション中にファイルのまま配置する際には、/WEB-INF/mld フォルダ直下に配置します。読み込み順より、この方法によって定義されるものを最優先します。

1.2.2

Jar ファイル中に配置

MLD を Jar アーカイブ中に配置することができます。Maya はクラスパス中の Jar アーカイブ中の /META-INF/ フォルダを検索し、みつかった全ての MLD ファイルを取り込みます。

1.3.

MLD の発展的な内容

MLD によって、組み込み機能を定義することにより、二次的な利点もあります。TLD に対応した TLDDoc(<http://developers.sun.com/dev/coolstuff/tlddoc/>)の様に、MLD からライブラリの情報を自動的にドキュメント化するツールを作成することも可能です。