



## 付録 C

---

この付録の構成は、次のとおりです。

- フォームのフィールドタイプ、1 ページ

## フォームのフィールドタイプ

この付録では、Open Automation モジュールでフォームを作成する際にフォームフィールドのタイプを定義するために使用するフォームのフィールドタイプの一覧を提供します。フォームのフィールドを定義するには、フォームのフィールドのラベルとタイプを指定することが必須です。

### 1 FIELD\_TYPE\_TEXT

FIELD\_TYPE\_TEXT はフィールドをテキストフィールドとして定義します。これは、デフォルトのフィールドタイプです。フォームのフィールド注釈にフィールドタイプが定義されていないと、フォームのフィールドはデフォルトでテキストタイプとして分類されます。

#### 属性

- maxLength : テキストフィールドに入力できる最大文字数を指定します。
- Size : 次のいずれかの値を使用して、テキストフィールドのサイズを設定します。
  - FIELD\_SIZE\_SMALL
  - FIELD\_SIZE\_MEDIUM
  - FIELD\_SIZE\_LARGE
  - FIELD\_SIZE\_MEDIUM\_SMALL
  - FIELD\_SIZE\_LARGE\_SMALL
  - FIELD\_SIZE\_SMALL\_MEDIUM
  - FIELD\_SIZE\_LARGE\_MEDIUM
  - FIELD\_SIZE\_SMALL\_LARGE
  - FIELD\_SIZE\_MEDIUM\_LARGE

**サンプル**

```
@FormField(label = "Name", help = "Name",
size = FormFieldDefinition.FIELD_SIZE_SMALL)
private String name;
```

**2 FIELD\_TYPE\_NUMBER**

**FIELD\_TYPE\_NUMBER** はフィールドを数値フィールドとして定義します。

**属性**

- minValue : 数値フィールドの最小許容値を指定します。たとえば 1 と入力します。
- maxValue : 数値フィールドの最大許容値を指定します。たとえば 65535 と入力します。

**サンプル**

```
@FormField(label = "FIELD_TYPE_NUMBER",
type = FormFieldDefinition.FIELD_TYPE_NUMBER, minValue = 1, maxValue = 65535)
private int number;
```

**3 FIELD\_TYPE\_TABULAR**

**FIELD\_TYPE\_TABULAR** はフィールドをテーブルとして定義します。

**属性**

- table : 表形式フィールドの名前を指定します。
- multiline : この属性はブール型です。テーブルに複数の行を追加できるようにするには、true に設定します。

**サンプル**

```
@FormField(label = "FIELD_TYPE_TABULAR",
type = FormFieldDefinition.FIELD_TYPE_TABULAR,
table = SimpleTabularProvider.SIMPLE_TABULAR_PROVIDER, multiline = true)
private String[] plainTabularValues;
```

**4 FIELD\_TYPE\_BOOLEAN**

**FIELD\_TYPE\_BOOLEAN** はフィールドをブール型として設定します。フィールドを選択するとフィールドの値が true に設定され、フィールドを選択しなければフィールドの値が false に設定されます。

**属性**

このフィールドに固有の属性はありません。

**サンプル**

```
@FormField(label = "FIELD_TYPE_BOOLEAN", type = FormFieldDefinition.FIELD_TYPE_BOOLEAN)
private boolean boolType;
```

**5 FIELD\_TYPE\_LABEL**

**FIELD\_TYPE\_LABEL** 引数を使用して、フィールドのラベルを指定できます。

**属性**

- htmlPopupTag : ポップアップアップ ウィンドウにロードする URL を指定します。
- htmlPopupLabel : ポップアップアップ ウィンドウのラベルを指定します。

- htmlPopupStyle : 次のいずれかの値を使用して、ラベルのポップアップスタイルを設定します。

- INFO\_TAG
- HELP\_TAG
- CUSTOM\_TAG
- INFO\_URL
- HELP\_URL
- CUSTOM\_URL

#### サンプル

```
@FormField(type = FormFieldDefinition.FIELD_TYPE_LABEL, label = "FIELD_TYPE_LABEL",
htmlPopupTag = "http://www.cisco.com", htmlPopupLabel = "http://www.cisco.com",
htmlPopupStyle = HtmlPopupStyles.CUSTOM_URL)
private String dummyLink;
```

### 6 FIELD\_TYPE\_EMBEDDED\_LOV

**FIELD\_TYPE\_EMBEDDED\_LOV** は、フィールドを値のリスト（LOV）が組み込まれるタイプとして定義し、ユーザが値のリストからいずれかの値を選択できるようにします。

#### 属性

- lov または lovProvider のいずれかを属性として指定できます。

#### サンプル

```
@FormField(label = "FIELD_TYPE_EMBEDDED_LOV", help = "Value",
type = FormFieldDefinition.FIELD_TYPE_EMBEDDED_LOV, lovProvider =
SimpleLovProvider.SIMPLE_LOV_PROVIDER)
private String value;
```

### 7 FIELD\_TYPE\_PASSWORD

**FIELD\_TYPE\_PASSWORD** はフィールドをパスワードフィールドとして設定します。パスワードフィールドに入力する文字はマスクされます（アスタリスクまたは円として表示されます）。

#### 属性

このフィールドに固有の属性はありません。

#### サンプル

```
@FormField(label = "FIELD_TYPE_PASSWORD",
type = FormFieldDefinition.FIELD_TYPE_PASSWORD)
private String password;
```

### 8 FIELD\_TYPE\_DATE

**FIELD\_TYPE\_DATE** はフィールドを日付入力フィールドとして定義します。

#### 属性

このフィールドに固有の属性はありません。

#### サンプル

```
@FormField(label = "FIELD_TYPE_DATE", type = FormFieldDefinition.FIELD_TYPE_DATE)
private long dateLong;
```

### 9 FIELD\_TYPE\_DATE\_TIME

## ■ フォームのフィールドタイプ

`FIELD_TYPE_DATE_TIME` はフィールドを日時入力フィールドとして定義します。

### 属性

このフィールドに固有の属性はありません。

#### サンプル

```
@FormField(label = "FIELD_TYPE_DATE_TIME", type = FormFieldDefinition.FIELD_TYPE_DATE_TIME)
private long dateTime;
```

## 10 FIELD\_TYPE\_MULTI\_SELECT\_LIST

`FIELD_TYPE_MULTI_SELECT_LIST` は、複数の値から入力を受け入れる入力フィールドを定義します。

### 属性

- `lovProvider` : 入力フィールドに表示する値のリストを設定します。

#### サンプル

```
@FormField(label = "FIELD_TYPE_MULTI_SELECT_LIST",
type = FormFieldDefinition.FIELD_TYPE_MULTI_SELECT_LIST, lovProvider =
SimpleLovProvider.SIMPLE_LOV_PROVIDER)
private String listView;
```

## 11 FIELD\_TYPE\_HTML\_LABEL

`FIELD_TYPE_HTML_LABEL` はフィールドを HTML ラベルとして定義します。この場合、HTML タグが文字列として受け入れられます。

### 属性

- `size` : HTML ラベルのサイズを設定します。

#### サンプル

```
@FormField(type = FormFieldDefinition.FIELD_TYPE_HTML_LABEL, label =
"FIELD_TYPE_HTML_LABEL", htmlPopupLabel = "<a href='http://www.cisco.com'>Cisco</a>")
private String dummyLink2;
```

## 12 FIELD\_TYPE\_FILE\_UPLOAD

`FIELD_TYPE_FILE_UPLOAD` は、ファイルをアップロードするためのフィールドを定義します。

### 属性

このフィールドに固有の属性はありません。

#### サンプル

```
@FormField(label = "FIELD_TYPE_FILE_UPLOAD", type =
FormFieldDefinition.FIELD_TYPE_FILE_UPLOAD)
private String uploadFileName;
```

## 13 FIELD\_TYPE\_TABULAR\_POPUP

`FIELD_TYPE_TABULAR_POPUP` は、ポップアップタイプの表形式フィールドを定義します。

### 属性

- `table` : Open Automation モジュールに登録済みの表形式フィールド名 (TabularPovider) を指定します。

次のサンプルコードに表形式プロバイダーを Open Automation モジュールに登録する方法を示します。

```
StorageModule.java(Registering Tabular report)
cfr.registerTabularField(SimpleTabularProvider.SIMPLE_TABULAR_PROVIDER,
SimpleTabularProvider.class, "0", "0");
```

### サンプル

```
@FormField(label = "FIELD_TYPE_TABULAR_POPUP",
type = FormFieldDefinition.FIELD_TYPE_TABULAR_POPUP, table =
SimpleTabularProvider.SIMPLE_TABULAR_PROVIDER)
private String tabularPopup;
```

## 14 FIELD\_TYPE\_EMBEDDED\_LOV\_RADIO

**FIELD\_TYPE\_EMBEDDED\_LOV\_RADIO** は、LOV オプションボタンが組み込まれたフィールドを定義します。

### 属性

- lov または lovProvider のいずれかを属性として選択できます。

### サンプル

```
@FormField(label = "FIELD_TYPE_EMBEDDED_LOV_RADIO",
type = FormFieldDefinition.FIELD_TYPE_EMBEDDED_LOV_RADIO, mandatory = true, lov = {
"Mode 1", "Mode 2", "Mode 3"}, validate = true, group="FIELD_TYPE_EMBEDDED_LOV_RADIO")
private String modeType = "Select Mode";
```

## 15 FIELD\_TYPE\_HTML\_TEXT

**FIELD\_TYPE\_HTML\_TEXT** は、HTML テキストタイプのフィールドを定義します。

### 属性

このフィールドに固有の属性はありません。

### サンプル

```
@FormField(label = "FIELD_TYPE_HTML_TEXT", type = FormFieldDefinition.FIELD_TYPE_HTML_TEXT,
editable = true, size=FormFieldDefinition.FIELD_SIZE_MEDIUM_SMALL)
private String status = "<h1>FIELD_TYPE_HTML_TEXT</h1>";
```

## 16 FIELD\_TYPE\_LABEL\_WITH\_SPACE

**FIELD\_TYPE\_LABEL\_WITH\_SPACE** は、スペースを含むラベルフィールドを定義します。

### 属性

このフィールドに固有の属性はありません。

### サンプル

```
@FormField(label = "FIELD_TYPE_LABEL_WITH_SPACE",
help = "Ordering of VNICs", type = FormFieldDefinition.FIELD_TYPE_LABEL_WITH_SPACE)
private String vnicLabel;
```

## 17 FIELD\_TYPE\_IMAGE\_SELECT\_LIST

**FIELD\_TYPE\_IMAGE\_SELECT\_LIST** は、イメージ選択リストからイメージを選択できるフィールドを定義します。

### 属性

このフィールドに固有の属性はありません。

**サンプル**

```
@FormField(label = "FIELD_TYPE_IMAGE_SELECT_LIST", type =
FormFieldDefinition.FIELD_TYPE_IMAGE_SELECT_LIST, mandatory = false, editable = true)
private String catalogIcon;
```

**18 FIELD\_TYPE\_BUTTON\_PANEL**

FIELD\_TYPE\_BUTTON\_PANEL はフィールドをボタンパネルとして定義します。

**属性**

- lov : 登録済み LOV プロバイダーネームを指定します。また、Lov={http,https} として値を直接指定することもできます。詳細については、Open Automation モジュールの SimpleLovProvider.java サンプル LOV プロバイダーを参照してください。

**サンプル**

```
@FormField(label = "FIELD_TYPE_BUTTON_PANEL",
type = FormFieldDefinition.FIELD_TYPE_BUTTON_PANEL,
lov = {"Discover Servers"}, validate = true, group = "UCSM/CIMC Common", mandatory =
false)
private String discoverServers = "Discover Servers";
```

**19 FIELD\_TYPE\_TEXT\_LINE\_NUMS**

FIELD\_TYPE\_TEXT\_LINE\_NUMS はフィールドを行番号付きテキストフィールドとして定義します。

**属性**

- maxlength : テキストフィールドに入力できる最大文字数を指定します。
- multiline : この属性はブール型です。テキストフィールドに複数の行を追加できるようにするには、true に設定します。
- size : 次のいずれかの値を使用して、テキストフィールドのサイズを設定します。
  - FIELD\_SIZE\_SMALL
  - FIELD\_SIZE\_MEDIUM
  - FIELD\_SIZE\_LARGE
  - FIELD\_SIZE\_MEDIUM\_SMALL
  - FIELD\_SIZE\_LARGE\_SMALL
  - FIELD\_SIZE\_SMALL\_MEDIUM
  - FIELD\_SIZE\_LARGE\_MEDIUM
  - FIELD\_SIZE\_SMALL\_LARGE
  - FIELD\_SIZE\_MEDIUM\_LARGE

**サンプル**

```
@FormField(label = "FIELD_TYPE_TEXT_LINE_NUMS",
help = "Error Text to validate", mandatory = false,
multiline = true, maxLength = 8192,
type = FormFieldDefinition.FIELD_TYPE_TEXT_LINE_NUMS,
size = FormFieldDefinition.FIELD_SIZE_SMALL_LARGE)
private String message;
```

## 20 FIELD\_TYPE\_LARGE\_FILE\_UPLOAD

`FIELD_TYPE_LARGE_FILE_UPLOAD` は、サイズの大きなファイルをアップロードするために使用するフィールドを定義します。

### 属性

このフィールドに固有の属性はありません。

### サンプル

```
@FormField(label = "", help = "Upload a file",
mandatory = true, validate = true, type = FormFieldDefinition.FIELD_TYPE_LARGE_FILE_UPLOAD,
annotation = "For module uploads only zip format are supported")
private String uploadFile;
```

## 21 FIELD\_TYPE\_COLORPICKER

`FIELD_TYPE_COLORPICKER` は、色の選択フィールドを定義します。

### 属性

このフィールドに固有の属性はありません。

### サンプル

```
@FormField(label = "FIELD_TYPE_COLORPICKER", help = "Color",
mandatory = true, validate = true, type = FormFieldDefinition.FIELD_TYPE_COLORPICKER)
private String color;
```

### 共通の属性

ここでは、フィールドアクティビティを制御するためにフォームのフィールドで使用できる共通の属性の一覧を提供します。たとえば、名前フィールドを必須フィールドにする場合、`mandatory` 属性の値として `true` を渡す必要があります。

- 1 `validate` : この属性はブール型です。デフォルトでは、この属性の値は `false` に設定されます。フィールドを検証する場合は、属性の値として `true` を渡す必要があります。
- 2 `hidden` : この属性はブール型です。デフォルトでは、この属性の値は `false` に設定されます。フォームでフィールドを非表示にする場合は、属性の値として `true` を渡す必要があります。
- 3 `mandatory` : この属性はブール型です。デフォルトでは、この属性の値は `false` に設定されます。フィールドを必須フィールドにする場合は、属性の値として `true` を渡す必要があります。
- 4 `editable` : この属性はブール型です。デフォルトでは、この属性の値は `false` に設定されます。この属性が適用されるのは、テキストフィールドのみです。テキストフィールドを編集可能にする場合は、属性の値として `true` を渡す必要があります。
- 5 `group` : この属性の型は文字列です。単一のフィールドまたは複数のフィールドをグループとして定義する場合は、属性の値としてグループ名を指定する必要があります。
- 6 `view` : この属性の型は文字列です。ウィザードに複数のフォームがある場合、特定のページ番号にフィールドが表示されるようにするには、属性の値としてページ番号を指定する必要があります。ウィザードに複数のページがない場合は、この属性を使用する必要はありません。
- 7 `help` : この属性の型は文字列です。この属性を使用して、フィールドについて説明するヘルプテキストを指定できます。指定したヘルプテキストは、ユーザがそのフィールドにマウス ポインタを重ねると表示されます。

## ■ フォームのフィールドタイプ