



- 1. 改訂情報
- 2. はじめに
  - 2.1. 本書の目的
  - 2.2. 対象読者
  - 2.3. 本書の構成
- 3. コンテンツ作成仕様
  - 3.1. コンテンツ作成対象
  - 3.2. コンテンツの情報
  - 3.3. ファセット
  - 3.4. 閲覧可能権限
- 4. クローラの種類と仕様
  - 4.1. 共通仕様
  - 4.2. [ ジョブ ] ワークフロークローラ
- 5. テンプレート仕様
  - 5.1. 表示項目
- 6. ワークフロークローラの拡張
  - 6.1. リスナーの呼び出し
  - 6.2. リスナーの設定
  - 6.3. リスナーの作成
  - 6.4. サンプルモジュール

変更年月日	変更内容
-------	------

2014-09-01	初版
------------	----

2017-08-01	第2版
------------	-----

- 「[閲覧可能権限](#)」の閲覧権限の対象者の説明を変更しました。
-

#### 項目

- [本書の目的](#)
- [対象読者](#)
- [本書の構成](#)

## 本書の目的

本書では IM-Workflow のクローラの機能概要と仕組の詳細について説明します。

## 対象読者

次の利用者を対象としています。

- intra-mart Accel Platformの IM-Workflow の全文検索機能（IM-ContentsSearch）を管理する運用担当者
- コンテンツ画面の項目の検索対象への追加などワークフロークローラの拡張開発を行いたい開発者

次の内容を理解していることが必須です。

- intra-mart Accel Platform の仕様
- IM-ContentsSearch の仕様
- IM-Workflow の仕様

## 本書の構成

本書の構成は以下の通りです。

- [コンテンツ作成仕様](#)  
コンテンツ作成仕様について説明します。
- [クローラの種類と仕様](#)  
クローラの種類と仕様について説明します。
- [テンプレート仕様](#)  
検索結果画面のテンプレートについて説明します。
- [ワークフロークローラの拡張](#)  
ワークフロークローラの拡張方法について説明します。



### コラム

#### 用語解説

Storage として使用するディレクトリを %PUBLIC\_STORAGE\_PATH% と略します。

項目

- コンテンツ作成対象
- コンテンツの情報
  - 標準フィールド
  - 動的フィールド
- ファセット
- 閲覧可能権限

## コンテンツ作成対象

ワークフローローラは、IM-Workflow で申請・承認が行われた案件を対象にインデックスを作成します。  
 以下の案件が作成対象です。

- 未完了案件
- 完了案件
- 過去案件

## コンテンツの情報

ワークフローローラでは、案件情報を以下の内容で登録します。

### 標準フィールド

No	フィールド名	設定値	備考
1	ID	workflow_%システム案件ID%	コンテンツを一意に識別するIDです。
2	TYPE	workflow workflow\$actv workflow\$actv\$%フローID% workflow\$cpl workflow\$cpl\$%フローID% workflow\$arc workflow\$arc\$%フローID%	コンテンツのタイプです。
3	URL	im_workflow/common/switch/switch_content_detail	案件の詳細画面のURLです。
4	ID_ORIGINAL	システム案件ID、ユーザーデータID、画面種別 (JSON形式)	コンテンツの元データにおける一意のキーとなるIDです。 【例】 {“imwSystemMatterId”:“ma_5ib903vqwivkefu”,“imwUserDataId”:“5ib903vn3blz
5	TITLE	案件名	案件名です。
6	TEXT	フロー名	フロー名です。
7	ATTACHMENT	添付ファイル名 添付ファイル内のテキストデータ	
8	RECORD_DATE	未完了案件の場合：最終処理日 完了案件、過去案件の場合：案件終了日	



#### コラム

フロー名は、フロー定義で指定したすべてのロケールの名称が検索対象です。

### 動的フィールド

No	フィールド名	設定値	備考
1	MATTER_NUMBER_STRING	案件番号	
2	APPLY_BASE_DATE_DATE	申請基準日	
3	CREATE_DATE_DATE	クローラ起動日時	
4	APPLY_AUTH_USER_NAME_STRING	認証ユーザ名	
5	FLOW_NAME_STRING	フロー名	



#### コラム

タイトル (TITLE)、テキスト (TEXT)、添付ファイル (ATTACHMENT) に登録した内容が検索時に検索対象となるフィールドです。

ワークフローローラではコンテンツ作成時に以下のファセット分類を適用しています。

- ワークフロー

ワークフローローラで作成されたすべてのコンテンツに対して“ワークフロー”というファセットを適用します。

- 案件状態

案件の状態に応じて、以下のファセットを適用します。

- 未完了案件
- 完了案件
- 過去案件

- フロー名

“未完了案件”、“完了案件”、“過去案件”のファセットを適用したコンテンツに対して、更にフロー名のファセットを適用します。

### コラム

#### ワークフローローラが全文検索画面に提供するファセットナビゲーションの例

全文検索画面の検索結果に IM-Workflow のコンテンツがある場合、以下の様なファセットナビゲーションが全文検索画面の左部に表示されます。リンクを選択することで検索結果の絞り込み検索を行うことができます。



ファセットに関する詳細は、「[IM-ContentsSearch プログラミングガイド](#)」を参照してください。

## 閲覧可能権限

ワークフローローラは作成するコンテンツに対して、コンテンツの閲覧を許可する権限を案件の状態に応じて以下のように設定します。

- 未完了案件 の場合

以下の利用者に閲覧権限を設定します。

- 処理権限者

該当の案件に対し、申請・承認等の処理を行った利用者  
いずれの処理も行っていない、または未到達のノードの処理対象者は含まれません。

- 参照権限者

ワークフローシステム管理者から案件の参照や操作を行う権限を付与された利用者

- 確認対象者

案件の確認ができる利用者

- 完了案件 の場合

未完了案件と同様です。

- 過去案件 の場合

以下の利用者に閲覧権限を設定します。

- 処理権限者

該当の案件に対し、申請・承認等の処理を行った利用者  
いずれの処理も行っていない、または未到達のノードの処理対象者は含まれません。

### コラム

アーカイブジョブの標準仕様では、過去案件の参照可能ユーザとして処理権限者を設定しています。過去案件の参照可能ユーザを追加するには、案件退避処理リスナーをカスタマイズする必要があります。詳細は「[IM-Workflow 仕様書](#)」を参照してください。



コラム

案件を参照可能な上記の利用者が変更・更新された場合には、必要に応じてインデックスの再作成を行ってください。



注意

上記の「参照権限者」には、ワークフローシステム管理者、またはワークフロー運用管理者の立場でのみ案件を参照可能なユーザは含まれません。

項目

- 共通仕様
  - 最終クローラ日時
- [ジョブ] ワークフロークローラ
  - [ジョブネット] 差分クローリング
  - [ジョブネット] 削除クローリング
  - [ジョブネット] 再作成クローリング

## 共通仕様

ジョブスケジューラ機能のジョブとして以下のジョブを提供しています。

- ワークフロークローラ

### 最終クローラ日時

ワークフロークローラでは、処理が正常終了した場合にクローラの最終実行日時をファイルに保存します。差分クローリング実行時にはこのファイルを参照し、前回の実行日時より後にコンテンツの登録・更新対象となった案件を抽出します。日時のフォーマットは“**yyyy-MM-dd HH:mm:ss**”です。

最終クローラ日時を保存するファイルは、クローラジョブの初回正常終了時にPublicStorageの以下のディレクトリに保存されます。

- %PUBLIC\_STORAGE\_PATH%/products/im\_contents\_search/store/%テナントID%/last\_crawling\_date/workflow.json

ただし、削除クローリングが実行された場合、最終クローラ日時は“**2000-01-01 00:00:00**”に初期化されます。

#### コラム

##### 初回実行時の動作仕様

初期状態の場合、最終クローラ日時を保存するファイルは存在していません。この場合、既定値として“2000年1月1日0時0分0秒”を最終クローラ日時として動作する仕様です。

初回実行時の最終クローラ日時を明示的に指定したい場合は、上記の「workflow.json」ファイルを作成し、任意の日時を上記の日時フォーマットにて設定してください。

- 「workflow.json」ファイルの設定例

```
{
  "workflow": "2014-09-01 00:00:00"
}
```

## [ジョブ] ワークフロークローラ

IM-Workflow のコンテンツの差分クローリングと削除クローリング、再作成クローリングを行うジョブです。

ジョブカテゴリ	IM-ContentsSearch クローラ
ジョブID	workflow-crawler
ジョブ名	ワークフロークローラ

ワークフロージョブは標準で以下のジョブネットにより登録されています。

- 差分クローリング (ジョブネットID: crawler-delta-jobnet)  
ワークフロークローラの差分クローリングを実行します。
- 削除クローリング (ジョブネットID: crawler-delete-jobnet)  
ワークフロークローラの削除クローリングを実行します。
- 再作成クローリング (ジョブネットID: crawler-reindex-jobnet)  
ワークフロークローラの削除クローリングを実行後に、差分クローリングを実行します。

### [ジョブネット] 差分クローリング

- 差分クローリング実行時の登録・更新対象

差分クローリング実行時にコンテンツの登録・更新の対象となる案件の条件は以下の通りです。最終クローラ日時以降で以下のいずれかの条件を1つでも満たす案件について、コンテンツの登録、または更新が行われます。

- 申請または処理が行われた案件



- 完了した案件
- アーカイブが行われた案件
- 差分クロール実行時の実行パラメータ

差分クロール実行時には、差分クロールのジョブネット、および、ワークフローローラのジョブ情報に設定されている以下の実行パラメータが利用されま  
す。

キー	説明
crawlingType	クローラの動作タイプを設定します。“DELTA”である場合のみ差分クロールが動作します。
groupName	処理対象とする検索サーバグループIDを指定します。
maxSegments	最適化の度合いを設定します。
withCommit	処理終了時に更新データを確定するかを設定します。
withOptimize	処理終了時にデータの最適化をするかを設定します。

### [ ジョブネット ] 削除クロール

ワークフローローラが作成したコンテンツをすべて削除します。

- 削除クロール実行時の実行パラメータ

削除クロール実行時には、削除クロールのジョブネット、および、ワークフローローラのジョブ情報に設定されている以下の実行パラメータが利用されま  
す。

キー	説明
crawlingType	クローラの動作タイプを設定します。“DELETE”である場合のみ削除クロールは動作します。
groupName	処理対象とする検索サーバグループIDを指定します。
maxSegments	最適化の度合いを設定します。
withCommit	処理終了時に更新データを確定するかを設定します。
withOptimize	処理終了時にデータの最適化をするかを設定します。

### [ ジョブネット ] 再作成クロール

ワークフローローラが作成したコンテンツを一旦すべて削除し、再度登録します。

- 再作成クロール実行時の実行パラメータ

再作成クロール実行時には、再作成クロールのジョブネット、および、ワークフローローラのジョブ情報に設定されている以下の実行パラメータが利用され  
ます。

キー	説明
crawlingType	クローラの動作タイプを設定します。“REINDEX”である場合のみ再作成クロールは動作します。
groupName	処理対象とする検索サーバグループIDを指定します。
maxSegments	最適化の度合いを設定します。
withCommit	処理終了時に更新データを確定するかを設定します。
withOptimize	処理終了時にデータの最適化をするかを設定します。

項目

- 表示項目

## 表示項目

ワークフローローラが作成するコンテンツの全文検索画面に表示される項目は、以下の通りです。

- 案件名
- 案件の最終処理日時
- 案件のパンくずリスト
- 案件番号
- フロー名
- 申請基準日
- 申請者

検索結果画面例

### 営業部門向け備品購入の件

2014/08/07 16:19:14 ワークフロー>完了案件>直線ルート[JavaEE開発モデル]  
案件番号: 0000000009 フロー名: 直線ルート[JavaEE開発モデル]  
申請基準日: 2014/08/07 申請者: 関根千香  
発モデル] 備品

## i コラム

検索結果のタイトルリンクをクリックすると、ポップアップで案件詳細画面が表示されます。



ワークフローローラでは、ワークフローの案件情報を「[コンテンツ作成仕様](#)」の内容でSolrサーバに登録しますが、ワークフローローラ用のリスナーを作成して追加することで、登録情報として独自の項目を追加することが可能です。

この章では、独自の項目を登録情報に追加するためのリスナーの作成方法について説明します。

なお、この章で登場する設定例やサンプルプログラムは、IM-Workflow で製品サンプルとして提供している「物品購買」の案件情報を文書の独自項目として追加することを目的としたサンプルです。

#### 項目

- リスナーの呼び出し
- リスナーの設定
- リスナーの作成
  - 実処理の記述
  - ワークフローローラのサンプルプログラム
- サンプルモジュール

## リスナーの呼び出し

ワークフローローラは案件単位で案件情報を文書として登録していきますが、登録処理の直前に設定ファイルに定義されたリスナーを呼び出し、リスナーが存在した場合はリスナーの項目追加メソッドを実行します。

## リスナーの設定

作成したリスナークラスは、plugin.xml で設定します。

- 対象ファイル

```
WEB-INF/plugin/jp.co.intra_mart.sample.contentssearch.workflow.crawler.listener/plugin.xml
```



#### コラム

WEB-INF/plugin 配下のプラグインフォルダ名は任意です。  
すでに存在するプラグインフォルダと重複しないフォルダ名を指定してください。

- 設定例

```
<?xml version="1.0" encoding="UTF-8"?>
<plugin>
  <extension point="jp.co.intra_mart.foundation.contentssearch.workflow.crawler">
    <imwcrawler-add-listener id="jp.co.intra_mart.sample.contentssearch.workflow.crawler.listener">
      <listener-class>jp.co.intra_mart.sample.workflow.purchase.listener.WorkflowItemDataCrawlingAddListner</listener-class>
    </imwcrawler-add-listener>
  </extension>
</plugin>
```

タグについて説明します。

- extension タグ

固定指定です。  
上記のまま指定してください。

- imwcrawler-add-listener タグ

id 属性を必ず指定してください。拡張ポイントが共通のプラグインを複数定義する場合は、プラグイン間で重複しない id を指定してください。

- listener-class タグ

作成したリスナークラスを指定します。

**i** コラム

ひとつの plugin.xml に複数のリスナークラスを指定する場合は、imwcrawler-add-listener タグを複数指定してください。

```
<?xml version="1.0" encoding="UTF-8"?>
<plugin>
  <extension point="jp.co.intra_mart.foundation.contentssearch.workflow.crawler">
    <imwcrawler-add-listener id="jp.co.intra_mart.sample.contentssearch.workflow.crawler.listener.item">
      <listener-class>jp.co.intra_mart.sample.workflow.purchase.listener.WorkflowItemDataCrawlingAddListner</listener-class>
    </imwcrawler-add-listener>
    <imwcrawler-add-listener id="jp.co.intra_mart.sample.contentssearch.workflow.crawler.listener.trip">
      <listener-class>jp.co.intra_mart.sample.workflow.purchase.listener.WorkflowTripDataCrawlingAddListner</listener-class>
    </imwcrawler-add-listener>
  </extension>
</plugin>
```

## リスナーの作成

作成するリスナーは以下のインタフェースを実装している必要があります。

```
jp.co.intra_mart.foundation.contentssearch.workflow.crawler.listener.WorkflowMatterCrawlingAddListner
```

## 実処理の記述

登録情報に独自の項目を追加するための実処理は addInputContent メソッド内に記述します。  
addInputContent メソッドの引数には、文書登録用オブジェクト、システム案件ID、ユーザデータID が渡されます。

リスナーでは以下の処理を行います。

1. 引数のパラメータ情報を元に、アプリケーションデータを取得する。
2. アプリケーションデータを登録情報に追加するか判定を行う。  
※ 登録の要否判定内容はクローラの要件によって決定してください。
3. 引数の文書登録用オブジェクトにアプリケーションデータを追加する。
  - 検索対象となるテキストフィールドにデータを登録したい場合  
jp.co.intra\_mart.foundation.contentssearch.model.content.InputContent # addText メソッドを使用
  - 表示目的等の格納用データを登録したい場合  
jp.co.intra\_mart.foundation.contentssearch.model.content.InputContent # addValue メソッドを使用して動的フィールドにデータを追加します。

**!** 注意

いずれのフィールド値にも null 値を設定することはできませんので、注意してください。

表示用に登録したデータを専用のレイアウトで検索結果画面に表示したい場合には、上記以外に以下の処理を行う必要があります。

- 文書種別の追加  
jp.co.intra\_mart.foundation.contentssearch.model.content.InputContent # addType メソッドを使用
- テンプレートの作成
- 追加した文書種別情報を設定ファイルに設定

テンプレートの作成・設定ファイルへの設定方法についての詳細は、「IM-ContentsSearch プログラミングガイド」を参照してください。

## ワークフロークローラのサンプルプログラム

```

package jp.co.intra_mart.sample.workflow.purchase.listener;

import static jp.co.intra_mart.foundation.contentssearch.model.field.Fields.STRING;
import static jp.co.intra_mart.foundation.contentssearch.web.common.Constants.TREE_FACET_SEPARATOR;
import static jp.co.intra_mart.foundation.contentssearch.workflow.common.Constants.IMW_DOCUMENT_TYPE;
import jp.co.intra_mart.foundation.contentssearch.exception.ContentsSearchCrawlingException;
import jp.co.intra_mart.foundation.contentssearch.model.content.InputContent;
import jp.co.intra_mart.foundation.contentssearch.workflow.crawler.listener.WorkflowMatterCrawlingAddListener;
import jp.co.intra_mart.framework.base.event.EventException;
import jp.co.intra_mart.framework.base.event.EventManager;
import jp.co.intra_mart.framework.base.event.EventManagerException;
import jp.co.intra_mart.framework.base.event.EventPropertyException;
import jp.co.intra_mart.framework.system.exception.ApplicationException;
import jp.co.intra_mart.framework.system.exception.SystemException;
import jp.co.intra_mart.sample.workflow.purchase.model.event.SelectEvent;
import jp.co.intra_mart.sample.workflow.purchase.model.event.SelectEventResult;

/**
 * サンプル 物品購買用 クローラ登録文書追加リスナー
 * @author INTRAMART
 */
public class WorkflowItemDataCrawlingAddListener implements WorkflowMatterCrawlingAddListener {

    @Override
    public void addInputContent(final InputContent inputContent, final String systemMatterId, final String userDataId) throws
    ContentsSearchCrawlingException {
        EventManager eventManager;
        try {
            // 1. 引数のパラメータ情報を元に、アプリケーションデータを取得する。
            eventManager = EventManager.getEventManager();
            final SelectEvent selectEvent = (SelectEvent) eventManager.createEvent("imw_sample_purchase", "select");
            selectEvent.setUserDataId(userDataId);
            final SelectEventResult eventResult = (SelectEventResult) eventManager.dispatch(selectEvent, true);
            // 2. アプリケーションデータを登録情報に追加するか判定を行う。
            if (eventResult.getModelObject() == null) {
                return;
            }
            final String itemName = eventResult.getModelObject().getItemName();
            if (itemName == null) {
                return;
            }
            // 3. 引数の文書登録用オブジェクトにアプリケーションデータを追加する。
            // 3-1. テキストフィールドに品名を追加
            inputContent.addText(itemName);
            // 3-2. 動的フィールドに品名を追加
            inputContent.addValue(STRING.toField("item_name_string"), itemName);
            // 3-3. 文書種別の追加 (静的ファセット)
            inputContent.addType(IMW_DOCUMENT_TYPE + TREE_FACET_SEPARATOR + "purchase");// "workflow$purchase"
        } catch (final EventManagerException e) {
            throw new ContentsSearchCrawlingException(e);
        } catch (final EventPropertyException e) {
            throw new ContentsSearchCrawlingException(e);
        } catch (final EventException e) {
            throw new ContentsSearchCrawlingException(e);
        } catch (final SystemException e) {
            throw new ContentsSearchCrawlingException(e);
        } catch (final ApplicationException e) {
            throw new ContentsSearchCrawlingException(e);
        }
    }
}
    
```

## サンプルモジュール

この章で紹介したリスナーの設定ファイル、およびサンプルプログラムをまとめたサンプルユーザモジュールです。

- サンプルソースコード

[im\\_workflow\\_contents\\_search\\_sample.zip](#)

- サンプルソースコードを含んだユーザ定義モジュール (imm ファイル)

[im\\_workflow\\_contents\\_search\\_sample-8.0.0.imm](#)

### コラム

このユーザモジュールの利用は、IM-Workflow のサンプルプログラムがデプロイされていることが前提です。ユーザモジュールの作成については、[intra-mart e Builder for Accel Platform アプリケーション開発ガイド / モジュール開発の基本機能を参照](#)ください。