3-1.処理詳細説明(全体)		システム名	f013カタログ部品データ作成	システムID		作成日	2025/9/24	作成者	3H.薛宸
		業務機能名	f013WEBカタログ部品情報excelデータ入替作成	バージョン		更新日	2025/11/6	更新者	3H.薛宸
プログラムID	f013-parts-list-applicable-period-reset	プログラム名称	B145_WEBカタログ部品情報Excelデータ加工1：部品の適用期間のリセット_車両モデル単位	プロジェクトNo	12170130-00	区分	専用		

詳細内容	改訂No.	削

・解析プラグイン定義

f013WEBカタログ部品情報excel累積Parquetファイル解析機能は、JSON形式の引数をもとに必要な情報を抽出し、部品一覧Excel累積Parquetファイルを読んで、レコード単位で戻します。

各プラグインは、以下の3段階処理（Before／Main／After）を通じてデータ解析を実施する。

前処理段階（Before）	実行環境の初期化、入力引数（JSON）の解析、指定ファイルの存在チェック等を行う。
本処理段階（Main）	部品一覧Excel累積Parquetファイルを読み込み、加工結果をレコード単位でコールバックメソッドへ返却する。
後処理段階（After）	処理結果の集計およびリソース解放を行い、最終的な終了コード（ExitCode）とメッセージ情報を返却する。

・API定義

public void Before(string pluginParamJson,LogUtil logUtil)

引数：

pluginParamJson	プラグイン実行に必要な情報を格納したJSON文字列。 ※詳細レイアウトは「プログラム基本設計書_プラグイン振分モジュール_解析加工IF定義」を参照。
logUtil	LogUtilインスタンス ※ログ出力を行うための LogUtil インスタンス。

戻り値：

なし

public void Main(Action<DataRow> onDataRow)

引数：

onDataRow	レコードを逐次返却する用コールバック ※読み取ったレコードを一つずつ返却するコールバック

戻り値：

なし

public int After(out string jsonMessage)

引数：

jsonMessgae	異常ありの場合、最後に発生したエラーのメッセージをJSON形式で返却する引数 DataLineageExecuteLogのdetail_message情報を更新するために使用される。

戻り値：

＞ 正常終了の場合、ExitCode=0

＞ 異常発生の場合、最後に発生したエラーの ExitCode

・処理詳細

1. 前処理段階（Before）

1.1 起動ログ情報の出力

プラグインの実行開始ログを出力する

INFO タイプ：「f013WEBカタログ部品情報Excelデータ加工1：部品の適用期間のリセット_車両モデル単位」: プラグインを開始します...

プロセスログ

PROCESS タイプ：「f013WEBカタログ部品情報Excelデータ加工1：部品の適用期間のリセット_車両モデル単位」: 前処理（Beforeメソッド）を実行

1.2 入力引数の解析

引数 pluginParamJson のJSON形式文字列から、使用する情報を解析します。

プロセスログ

PROCESSタイプ：「f013WEBカタログ部品情報Excelデータ加工1：部品の適用期間のリセット_車両モデル単位」: 引数（JSON）内容を解析

1.3 入力引数の内容チェック

指定された Input スキーマファイルに対して、柔軟に対応できるように、最小限の妥当性チェックだけを実施します。

LineageId をもとに転記仕様 YAML（transformYamlInfoDic）から該当のInputスキーマファイルを取得します。

必要な項目IDが含まれているかを確認します。

プロセスログ
PROCESSタイプ:「f013WEBカタログ部品情報Excelデータ加工1:部品の適用期間のリセット_車両モデル単位」: 指定されたInputスキーマファイル「{schemaPath}」のチェックを実行

1.3.1 項目IDチェック

・チェック対象は、必須項目である「項目ID」(item_id) の存在確認に限定します。
＞ Inputスキーマファイルに、定義された必須項目 IDが存在しない場合、

エラーログを出力する(ExitCode:253)
ERRORタイプ:当リネージID「xxxx」の「リソース管理テーブル」において、指定されたInputスキーマYamlファイル「xxxx」に必須項目「xxxx」が存在しません。

必須項目一覧(本プラグインで定義される項目):

項目No.	項目ID	項目名
1	f013_acquisition_date	f013・取得日
2	f013_main_group_code	f013・メイングループコード
3	f013_main_group	f013・メイングループ
4	f013_sub_group_code	f013・サブグループコード
5	f013_sub_group	f013・サブグループ
6	f013_service_parts_news	f013・サービスパーツニュース(SPN)
7	f013_parts_name_code	f013・品名コード
8	f013_parts_number	f013・部品番号
9	f013_parts_name	f013・部品名称
10	f013_parts_quantity	f013・部品数量
11	f013_parts_price_group	f013・部品価格グループ
12	f013_genuine_parts_price	f013・MFTBC価格
13	f013_parts_specific information	f013・部品固有情報
14	f013_parts_remarks	f013・部品備考
15	f013_parts_maker_code	f013・部品メーカーコード
16	f013_supply_mode	f013・部品供給モード
17	f013_options	f013・部品オプション
18	f013_supply_alt_parts_number	f013・供給代替部品番号
19	f013_supply_alt_parts_maker_code	f013・供給代替部品メーカーコード
20	f013_supply_alt_parts_price_group	f013・供給代替部品価格グループ
21	f013_supply_alt_parts_price	f013・供給代替部品希望小売価格
22	f013_parts_start_date	f013・部品開始日
23	f013_parts_end_date	f013・部品終了日
24	f013_d2s_protal_name	f013・D2S部品提供名称
25	f013_car_model	f013・車両モデル
26	f013_car_model_classification_code	f013・類別コード

・それ以外の項目(例:項目名、項目順、型、項目長など)については、個別の検証は行いません。

1.4 入力ファイルチェック

プロセスログ
PROCESSタイプ:「f013WEBカタログ部品情報Excelデータ加工1:部品の適用期間のリセット_車両モデル単位」: 入力ファイルスキーマのチェックを実行

1.4.1 Parquetファイル名リストの取得

命名規則を満たすParquetファイルのパスリストを取得する。

1.4.2 Parquetファイルのレイアウトをチェックする

Parquetファイルのレイアウトチェック(Metadata使用)
Parquet.netを使用してParquetファイルのMetadataを読み取り、列情報(列数、列名、データ型など)を取得する。
取得したMetadataに基づき、列数・列IDなどの構造情報とInputスキーマを検証する。

カラム数のチェック
＞ 一致しない場合は、下記のエラーメッセージを表示し、

エラーログを出力する(ExitCode:242)
ERROR タイプ:「f013WEBカタログ部品情報Excelデータ加工1:部品の適用期間のリセット_車両モデル単位」: 対象データファイル「{parquetFilePath}」の項目数「{actualColumnNameList.Count}」が、プラグインで

カラム名称チェック
　＞　一致しない場合は、下記のエラーメッセージを表示し、

　　エラーログを出力する（ExitCode：242）
　　ERROR タイプ：「f013WEBカタログ部品情報Excelデータ加工1：部品の適用期間のリセット_車両モデル単位」: 対象データファイル「{parquetFilePath}」には、プラグインに定義済みの項目ID:「{parquetHeaderNam

※本プラグインで処理対象となるParuqetファイルの構造（カラム）は、以下の固定構造に従うものとする。

列No.	列名
1	f013_acquisition_date
2	f013_main_group_code
3	f013_main_group
4	f013_sub_group_code
5	f013_sub_group
6	f013_service_parts_news
7	f013_parts_name_code
8	f013_parts_number
9	f013_parts_name
10	f013_parts_quantity
11	f013_parts_price_group
12	f013_genuine_parts_price
13	f013_parts_specific information
14	f013_parts_remarks
15	f013_parts_maker_code
16	f013_supply_mode
17	f013_options
18	f013_supply_alt_parts_number
19	f013_supply_alt_parts_maker_code
20	f013_supply_alt_parts_price_group
21	f013_supply_alt_parts_price
22	f013_parts_start_date
23	f013_parts_end_date
24	f013_d2s_protal_name
25	f013_car_model
26	f013_car_model_classification_code

1.5 Google Cloud Storage 連携の事前準備

プロセスログ
　PROCESSタイプ：「f013WEBカタログ部品情報Excelデータ加工1：部品の適用期間のリセット_車両モデル単位」: DuckDBでGCS（Google Cloud Storage）にアクセスする際の認証

1.5.1 httpfs拡張機能のインストール
　DuckDBでGCSを利用するために、httpfs拡張機能をインストールする
　この作業は初回のみ実行する必要がある

　以下のSQL文を実行し、httpfs拡張機能をインストールする。
　　INSTALL httpfs;
　　LOAD httpfs;

1.5.2 DuckDBへの認証情報（HMAC認証キー）設定

1.5.2.1 HMAC認証情報取得
　　環境変数HMAC_KEY_ID_RESOURCEMANAGEからKey IDを取得
　　環境変数HMAC_SECRET_RESOURCEMANAGEからSecretを取得

1.5.2.2 HMAC認証情報登録
　　取得したHMAC認証情報をDuckDBのSECRETオブジェクトとして登録します。
　　以下のSQL文を実行し、HMAC認証情報を登録する。
　　　CREATE OR REPLACE SECRET gcs_secret (

```
    TYPE gcs,
    KEY_ID 'xxxx',
    SECRET 'xxxx'
);
```

1.6 Before段階の例外処理

前処理段階(Before)にて、予期しない例外が発生した場合の処理を以下に示します。

例外発生時、以下の3条のエラーログを出力します:

ERROR タイプ: Before段階でエラーが発生しました。ファイル: {excelFilePath}
ERROR タイプ: {ex.Message}
ERROR タイプ: {ex.StackTrace}

2. 本処理段階 (Main)

部品一覧Excel累積Parquetファイルを読み込み、加工結果をレコード単位でコールバックメソッドへ返却する。

2.1 前処理段階 (Before)の結果判定:

ExitCode 情報に基づき、前処理段階 (Before) の実行結果を判断します。

＞ 前処理段階(Before)で Error が発生した場合
本処理段階(Main) の実行を スキップ します。
プロセスは直接後処理段階(After)に移行し、該当する例外結果を返却します。

PROCESS タイプ:「f013WEBカタログ部品情報Excelデータ加工1:部品の適用期間のリセット_車両モデル単位」: Before処理でエラーが発生したため、Main処理をスキップします。ExitCode: {exitCode}

＞ 前処理段階(Before)で Error が発生しなかった場合
後続処理を 続行 します。

2.2 加工処理1:部品開始日、部品終了日の補完

プロセスログ
PROCESS タイプ:「f013WEBカタログ部品情報Excelデータ加工1:部品の適用期間のリセット_車両モデル単位」: 本処理(Mainメソッド) 実行

入力ファイル情報を出力します。
INFO タイプ:入力ファイル数: X
INFO タイプ: 入力ファイル: {excelFilePath}

複数のファイルがある場合は、最初の3つのファイルだけを印刷します。
入力ファイル: gs://bucket/a/file1.parquet,gs://bucket/a/file2.parquet,gs://bucket/a/file3.parquet...

2.2.1 加工処理対象車両モデルの箇所、ファイル命名ルールの作成

下記の情報で、インポート対象のフォルダーを決定する。

・「車両モデル」 ← 引数Jsonの入力ファイルリスト「DataFileList」から
・「カタログNo.」、「基準年月」 ← イベントファイルから
「カタログNo.」は、イベントファイルの!FS_CAR_CATALOG!変数です。
「基準年月」は、イベントファイルの!OBTAIN_YYYYMM!変数です。

※当リネージのリソース設定の「repository」、「 filenamerule」を条件で、対象のファイル一覧を決まります。
repositoryとfilenamerule 内のパスの一部を、実際の値で置換する

2.2.2 上記で車両モデル情報、リネージの設定情報で決まったフォルダーから、実際の対象Parquetファイル命名ルールで、DuckDBにインポートする。

※ 車両モデルで決まった累積ParquetファイルリストをDuckDBにインポートする。

プロセスログ
PROCESSタイプ:「f013WEBカタログ部品情報Excelデータ加工1:部品の適用期間のリセット_車両モデル単位」: ParquetファイルをDuckDBにインポート

2.2.2.1 Excel累積Parquetファイルのインポート

取込方式（DuckDB・ワイルドカード一括取込）

対象が GCS 上のファイルであるため、DuckDB の **GCS 接続用拡張（例：gcs / httpfs、HMAC）** を事前にインストールし有効化すること。

例：

gs://<bucket>/CatalogParts/f013/WebEpc/!BL_CATALOG_NO!/Ac/!OBTAIN_YYYYMM!01/!FS_CAR_MODEL!/catalog_parts_info_excel_*.parquet

※「f013カタログ部品情報excelファイル」累積処理のリネージの命名規則に従って、ワイルドカード処理名を設定する。

2.2.3 「部品開始日」、「部品終了日」のセット:

プロセスログ 3

PROCESSタイプ：「f013WEBカタログ部品情報Excelデータ加工1：部品の適用期間のリセット_車両モデル単位」: 部品の適用期間のリセット 3

行毎に、「部品開始日」「部品終了日」の両方が空や、「部品開始日」だけが空のレコードは、日付セットの対象レコードにして、下記のルールで日付の補完をする。

2.2.3.1 基準日付の取得:

・「部品開始日」：当車両モデルのすべてのレコードリストにて、最小日付(最も古い日付)
　最小の日付は、Null、0以外の最小日付です。
・「部品終了日」：当車両モデルのすべてのレコードリストにて、最大日付(最も新しい日付)

2.2.3.2 「部品開始日」「部品終了日」のセット処理

＞「部品開始日」と「部品終了日」の両方が未設定
　部品開始日：最小日付(最も古い日付)
　部品終了日：最大日付(最も新しい日付)
＞「部品開始日」のみ未設定
　部品開始日：最小日付(最も古い日付)

2.3 ページ単位で加工済みデータを読み込み、逐次的に結果を返却する

LineageIDに対応する入力スキーマレイアウトに従い、DataRow形式の解析結果データをコールバック方式（onDataRow）で呼び出し元に逐次返却します。

~~DuckDBにて、上記の補完できたテーブルから、DataRow 形式で逐次的に（yield return による遅延評価）返却する。~~

※メモリコストを抑えるために、1万行単位で検索して、ループ処理する。

~~2.3.1 エラーデータの処理~~

~~行(レコード)単位で、異常が発生すると、最後の異常内容をキャシューして、プラグインの実行結果として処理する。(Afterメソッドで戻します)~~

~~レコードのエラーは、エラーハンドリング設定に従って対応されます。~~

~~まず、異常行のログ出力し、プラグインの実行結果メッセージ、ExitCodeも更新しておいて~~

~~ExitCodeも、エラーハンドリングによってセットする。~~

~~エラーハンドリングモード一覧~~

~~エラーハンドリング~~	~~処理方法~~
~~resume~~	~~エラーレコードをスキップし、次のレコードから処理を続行します。(ExitCode：124)~~ ~~エラーメッセージを出力し~~
~~skip~~	~~現在ファイルをスキップします。(ExitCode：124)~~ ~~エラーメッセージを出力し~~
~~terminate~~	~~エラーメッセージを出力し、全処理を終了します。(ExitCode：255)~~

~~各エラーハンドリングモードで、以下の4種類のログを順次出力します：~~

~~ERRORタイプ：エラーファイル: {excelFilePath}~~
~~ERRORタイプ：エラー行番号:{currentRowNumber}~~
~~ERRORタイプ：エラー明細: {ex.Message}~~
~~ERROR タイプ： {ex.Message}~~
~~ERROR タイプ： {ex.StackTrace}~~

2.4 Main段階の例外処理 3

処理段階(Main)にて、予期しない例外が発生した場合の処理を以下に示します。
ERROR タイプ: Main段階でエラーが発生しました。ファイル: {excelFilePath}
ERROR タイプ: {ex.Message}
ERROR タイプ: {ex.StackTrace}

3. 後処理段階 (After)

処理結果の集計およびリソース解放を行い、最終的な終了コード(ExitCode)とメッセージ情報を返却する。

プロセスログ

PROCESS タイプ: 「f013WEBカタログ部品情報Excelデータ加工1:部品の適用期間のリセット_車両モデル単位」: 後処理(Afterメソッド) 実行

3.1 処理完了時に、実行結果ログを出力する。

メッセージ:

> 正常完了

INFO タイプ: 「f013WEBカタログ部品情報Excelデータ加工1:部品の適用期間のリセット_車両モデル単位」が正常完了しました。

> 異常終了

ERROR タイプ: 「f013WEBカタログ部品情報Excelデータ加工1:部品の適用期間のリセット_車両モデル単位」が異常終了しました。

3.2 ExitCodeおよびメッセージ情報の返却

＞ 正常終了の場合、ExitCode=0 でリターン、jsonMessageは空です。

＞ 異常発生の場合、最後のエラーに対する ExitCode とエラー情報を返却する