

ODIP 4.2 リリースノート

2021/11/01

(株) インテリジェント・モデル

ODIP は、(株) インテリジェント・モデル社の登録商標です。

本書に掲載された情報に基づいた行為の結果として発生した損害、利益の損失、経費などについて、(株) インテリジェント・モデルならびに本書の製作関係者は一切の責任を負いません。

本書は著作権法上の保護を受けています。本書の一部あるいは全部を無断で転載・複製することは法律で定められた場合を除き、禁止されています。

目 次

A. 変更内容	5
1. サポート対象 DBMS の追加・変更	5
(1) Amazon Redshift の追加	5
(2) Snowflake の追加	6
(3) Oracle・SQL Server のロード処理改定	7
(4) DBMS 名の表記変更	8
2. プロセス実行制御の拡張・変更	8
(1) Web API によるプロセス起動の追加	8
(2) execjob コマンドの追加	11
(3) プロセスログ出力機能追加	12
(4) 一時テーブルを使用したファイル結合の改定	13
(5) 固定長 COBOL の出力仕様変更	13
(6) 「データ複写」の仕様変更	13
3. コンポーネント定義の改定	14
(1) クロス集計・グループ集計の Interim データセット廃止	14
(2) “クロス集計併合入力データ”コンポーネントの廃止	14
(3) LIKE 関数の改定	14
(4) MCHAR/SCHAR 関数による特殊文字の変換	14
4. CSV ファイル入出力の改定	15
(1) 設定ファイルのオプション追加	15
(2) CSV ファイル出力オプション	15
(3) NULL と空文字の取扱い	16
(4) エンコーディングエラーの取扱い	16
(5) 改行コードの取扱い	16
5. リッチテキストファイル出力の改定	17
(1) サードパーティコンテンツの変更	17
(2) 詳細設定のエクスポート、インポート機能追加	17
(3) 出力オプションの改定	17
(4) RTF 文書出力仕様の変更	18
(5) 出力項目の改定	18
6. ヘルプメニューの表示変更	18
7. その他の修正	19
(1) ODIP アドミニストレータの改定	19
(2) ODIP オペレーションマネージャの改定	26

(3) ODIP リポジトリマネージャの改定.....	26
(4) ODIP プロセスマネージャの改定.....	26
(5) ODIP リポジトリサーバの改定	28
(6) ODIP トランスフォーマの改定	28
B. バージョンアップによる影響	33
1. 定義内容・検査に関する影響.....	33
2. 処理実行結果に関する影響	33

A. 変更内容

1. サポート対象 DBMS の追加・変更

(1) Amazon Redshift の追加

データソース情報の DBMS 名に、「Amazon Redshift」が追加され、処理の入出力データベース、及び ODIP トランスフォーマ・リポジトリに Amazon Redshift が使用できるようになりました。

① 使用準備

ODIP トランスフォーマの処理に DBMS のローダを使用する場合、次の準備が必要です。

- ・ Amazon S3 パケット、Amazon Identity and Access Management(以下、IAM)、Access Key(ID) 及び Secret Access Key が必要です。
- ・ ODIP トランスフォーマ導入端末に AWS Command Line Interface(以下、AWS CLI)及び PostgreSQL をインストールしてください。
- ・ ODIP トランスフォーマ導入ディレクトリの config/jdbcsample ディレクトリに、初期設定された Amazon Redshift 用の設定ファイルがあります。config/jdbcsample/redshift.properties を config ディレクトリにコピーし、下表のオプションを記述してください。

redshift.properties に設定するオプション

オプション	説明
aws.accesskey.id	IAM から発行した Access Key(ID)を指定します。
aws.accesskey.secret	IAM から発行した Secret Access Key を指定します。
aws.region	利用するリージョンを指定します。
aws.bucketname	ロード用データファイル(dat ファイル)のアップロード先(Amazon S3 パケット)を指定します。「s3://odip/data/」のように文末に/(スラッシュ)を指定する必要があります。

② 制限事項

ODIP 製品で Amazon Redshift を使用する場合、次の制限があります。

- ・ ODIP アドミニストレータの[データセット] > [データセット・インデックス]、[インデックス・カラム]の定義にプライマリー、ユニーク機能を使用することが

できません。

- ・ ODIP アドミニストレータのリレーション属性、再帰開始行の値、再帰開始行の条件で SUBSTRB 関数を使用することができません。
- ・ ODIP アドミニストレータのデータセットから[データ表示]で出力されるデータ表示ダイアログの[< 前ページ]、[<< 最初のページ]、[CSV ファイルに書き出し]機能は使用することができません。※1

※1 データセットの[データベース・ツール]>[データエクスポート]からテーブルデータを CSV ファイルに出力することができます。

(2) Snowflake の追加

データソース情報の DBMS 名に、「Snowflake」が追加され、処理の入出力データベース、及びトランスフォーマ・リポジトリに Snowflake が使用できるようになりました。JDBC ドライバの URL は既定では、次の形式になっています。パラメーターを追加する場合は、config/jdbcsample/snowflake.properties を config ディレクトリにコピーして jdbc.url の値を変更してください。

```
jdbc:snowflake://<account_identifier>.snowflakecomputing.com/?db=<dbname>
```

① 使用準備

ODIP から Snowflake に接続するため、次の準備が必要です。

- ・ SnowSQL (CLI クライアント)導入端末に ODIP 製品をインストールしてください。
- ・ ODIP トランスフォーマのローダ処理を使用する場合、Snowflake に接続するユーザのパスワードを SnowSQL 構成(~/.snowsql/config)ファイルの password オプションまたは環境変数 SNOWSQL_PWD を設定してください。
- ・ ODIP トランスフォーマのローダ処理にて、SnowSQL を介してエラーが発生した場合、クエリの実行を停止させるため SnowSQL 構成(~/.snowsql/config)ファイルに exit_on_error=TRUE オプションを追加してください。
- ・ ODIP 製品のデータソース情報、トランスフォーマリポジトリ情報などでホスト名(host)の入力項目には、Snowflake のアカウント識別子を指定してください。(＜組織名>.<アカウント名>または＜アカウントロケータ>.<リージョン ID>)
- ・ インストールされた ODIP トランスフォーマの config/jdbcsample ディレクトリに初期設定された Snowflake の設定ファイルがあります。ローダ処理を実行する場合、snowflake.properties のファイルをトランスフォーマの config ディレクトリにコピーして、下表のオプションを指定してください。

snowflake.properties に設定するオプション

オプション	説明
loader.stage.name	ロード用データファイル(dat ファイル)をアップロードする内部/外部ステージの名前(名前付きステージ)を指定します。
loader.stage.type	ロード用データファイル(dat ファイル)をアップロードする内部/外部ステージを指定します。デフォルトは内部ステージです(internal)。外部ステージ(Amazon S3)を指定する場合は、値を external に変更してください。
loader.command.external.location	loader.stage.type の値が external の場合のみ有効です。外部ステージの場所を指定します。(Amazon S3 のみ対応)バケット名は、“s3://odip/data/”のように文末に/(スラッシュ)を指定する必要があります。
loader.command.internal.upload	ロード用データファイル(dat ファイル)を Snowflake の内部ステージへアップロードするコマンドです。
warehouse	Snowflake で実行ユーザに DEFAULT_WAREHOUSE が設定されていない場合、使用するウェアハウスを本オプションで指定します。また、本オプションを有効にする場合、snowflake.properties 内でコメントアウトされている jdbc.url 及び loader.command とコメントアウトされていない既存定義のオプションの値を入れ替えてください。

② 制限事項

ODIP 製品で Snowflake を使用する場合、次の制限があります。

- ・ ODIP アドミニストレータの[データベース・ツール] > [データ表示]の[ページ移動]、「CSV ファイルに書き出し」を行うことができません。※1
- ・ インデックスをサポートしていないため、ODIP アドミニストレータの「データセット・インデックス」及び「インデックス・カラム」の定義を利用することができません。
- ・ ODIP アドミニストレータのリレーション属性、再帰開始行の値、再帰開始行の条件で SUBSTRB 関数を使用することができません。
- ・ Snowflake の外部ステージは、Amazon S3 のみ対応しています。

※1 データセットの[データベース・ツール]>[データエクスポート]からテーブルデータを CSV ファイルに出力することができます。

(3) Oracle・SQL Server のロード処理改定

Oracle 及び SQL Server に DBMS のローダを使用してデータ出力を行う場合に、DBMS のローダの制約によって文字型の列値に含まれる改行コードを削除していました。本バージョンでは、ロードコマンドの引数指定及びロードデータファイルの出力仕様変更によって、改行コードを含む文字列をそのままロードできるようになりました。DBMS 設定ファイルの次の追加プロパティによって、改行コードを含む文字列ロードの動作を指定します。

改行コードを含む文字列ロードのためのプロパティ追加

プロパティ名	説明
loader.remove.linefeed	文字型の列値に含まれる改行コードを削除してロードするか否かを true または false で指定します。Oracle、SQL Server を使用する処理で、旧バージョンの動作を維持する場合は、このオプションに true を設定します。
loader.command.extra	SQL Server の場合だけ有効です。改行コードを含む文字データに対応した BCP コマンドのテンプレートを指定します。
loader.control.extra	Oracle の場合だけ有効です。改行コードを含む文字データに対応したコントロールファイルの記述内容のテンプレートを指定します。

(4) DBMS 名の表記変更

データソース情報及びトランスフォーマ・リポジトリ情報の DBMS 名の表記を次のように変更しました。

- ・ SQL Server ⇒ SQL Server / Azure SQL Database
- ・ Azure SQL Data Warehouse ⇒ Azure Synapse Analytics

2. プロセス実行制御の拡張・変更

(1) Web API によるプロセス起動の追加

Web API 経由でプロセスを起動する機能を追加しました。JSON 形式のリクエストを POST することで、プロセスを起動することができます。プロセスはトランスフォーマサーバ上の一つのスレッドとして動きます。指定可能なオプションなど Web API の詳細は「ODIP トランスフォーマリファレンスガイド」を参照してください。

① Web API の概要

次は ODIP が提供する Web API の URI と HTTP メソッドです。

Web API の URI と HTTP メソッド

URI	HTTP メソッド	説明
/jobs	POST	ジョブを実行します。 プロセス ID 及びその他オプションは、リクエストボディに JSON 形式で指定します。
	GET	トランスフォーマサーバで実行中のジョブ一覧を配列で返します。実行中のジョブがない場合、空の配列[]が返ります。
/jobs/{instanceName}	GET	指定されたジョブの情報を返します。 対象のジョブが実行中または待機中でない場合は、HTTP ステータスコード 404 が返ります。
	DELETE	実行中のジョブをキャンセルします。 対象のジョブが実行中または待機中でない場合は、HTTP ステータスコード 404 が返ります。
/server	GET	サーバ情報を返します。 クエリパラメータ checkOnly=true が指定された場合は、空のレスポンスを返します。
/server/jobQueue	GET	実行中、待機中のジョブ情報を返します。 オペレーションマネージャのトランスフォーマ情報の「ジョブキュー」と同じ内容になります。実行中、待機中のジョブがない場合、空の配列[]が返ります。
/server/config	GET	トランスフォーマの odip.ini の設定値を返します。オペレーションマネージャのトランスフォーマサーバ情報の「設定値」と同じ内容になります。クエリパラメータの valueOnly=true が指定された場合は、デフォルト値、説明文は返されません。

② 設定ファイルのオプション追加

設定ファイル (odip.ini) に、Web API に関するオプションが追加になりました。次は追加された主なオプションになります。すべてのオプション及び詳細は「ODIP トランスフォーマリファレンスガイド」を参照してください。

Web API に関連する主な追加オプション

オプション	説明
server.webapi.contextpath=[String]	Web API のコンテキストパスを指定します。
server.webapi.http.enabled=[true false]	HTTP ポートの有効・無効を指定します。 (既定値：false)
server.webapi.http.port=[port number]	HTTP のポート番号を指定します。(既定値：8080)
server.webapi.ssl.enabled=[true false]	HTTPS ポートの有効・無効を指定します。(既定値：false)
server.webapi.ssl.port=[port number]	HTTPS のポート番号を指定します。(既定値：8181)
server.webapi.authtype=[NONE BASIC DIGEST CERT]	Web API の認証方法を指定します。
server.webapi.ssl.keystore.type=[String]	キーストアの種類を指定します。(既定値：JKS)
server.webapi.ssl.keystore.path=[String]	キーストアファイルのパスを指定します。 (既定値：./config/keystore.jks)
server.webapi.ssl.keystore.password=[String]	キーストアへのアクセスに使用されるパスワードを指定します。
server.webapi.cors.enabled=[true false]	CORS (クロスオリジンリソース共有) の有効・無効を指定します。

③ コネクションプールに関する設定の追加

jdbconfig.xml にコネクションプールに関する設定が追加されました。この設定は、Web API またはスレッドモードでプロセスを実行した場合に有効です。次は設定可能なオプションと設定例です。

表 1 コネクションプールに関する jdbcconfig.xml の追加設定

設定項目	説明
connectionpool	initialsize、maxactive、maxidle、minidle、maxwait の親要素です。
initialsize	接続プール開始時に作成されるデータベース接続の初期数を指定します。
maxactive	接続プールから同時に割り当てることができる、アクティブなデータベース接続の最大数を指定します。
maxidle	接続プールに保持する必要がある接続の最大数を指定します。この数を超えた接続プール内のアイドル接続は、約 60 秒経過後に切断されます。
minidle	接続プールに保持する接続の最小数を指定します。
maxwait	接続プールから取り出した接続が maxactive 数を超えていて、利用できる接続が存在しない場合に、接続を要求するジョブが待機する最大時間をミリ秒で指定します。指定された時間を超えても接続を使用できない場合、ジョブはエラーで終了します。

jdbcconfig.xml の設定例

```
<?xml version="1.0" encoding="UTF-8"?>
<sqlparameters>
  <datasource name="default">
    . . .
    <connectionpool>
      <initialsize>10</initialsize>
      <maxactive>100</maxactive>
      <maxidle>100</maxidle>
      <minidle>10</minidle>
      <maxwait>30000</maxwait>
    </connectionpool>
  </datasource>
</sqlparameters>
```

④ 制限事項

Web API では、次に該当するプロセスを実行することはできません。

- ・ 時系列演算を含む管理単位のプロセス
- ・ ODIP オペレーションマネージャの“リカバリ管理”タブに表示されるリカバリ用の受渡、保存、読込を伴うプロセス

(2) execjob コマンドの追加

ODIP トランスフォーマサーバを介さずに、単一のプロセスを実行する `execjob` コマンドを追加しました。このコマンドは、実行環境から直接トランスフォーマをリポジトリ及びデータソースにアクセスし、トランスフォーマエンジンの処理を実行します。コマンドの引数は概ね `startjob` コマンドと同じです。ジョブネット制御（前後関係のチェック、ステータス管理など）は行われません。また、トランスフォーマリポジトリ内の運用管理テーブル及びトランスフォーマサーバのログファイル（`odipjob.log`、`odip_trace.log`）に実行の記録を残しません。実行環境の設定ファイル（`odip.ini`）に次のオプションを追加しています。これらのオプションを設定することで、`odipjob.log` と同様のログを取得することができます。コマンドは実行環境の `command.log` ファイルには、`odip_trace.log` に出力する `startjob` の実行ログと同様のログを出力します。

① 設定ファイルのオプション追加

設定ファイル（`odip.ini`）に、`execjob` コマンドのログ出力に関する次のオプションが追加になりました。

execjob コマンドに関連する追加オプション

オプション	説明
<code>execjob.log.enabled=<true false></code>	ジョブログ出力を有効（true）または無効（false）に設定します。
<code>execjob.log.file.name=<pattern></code>	ジョブログ出力先ファイル名のパターンを指定します。

② 制限事項

`execjob` コマンドでは、次に該当するプロセスを実行することはできません。

- ・ 時系列演算を含む管理単位のプロセス
- ・ ODIP オペレーションマネージャの“リカバリ管理”タブに表示されるリカバリ用の受渡、保存、読込を伴うプロセス

(3) プロセスログ出力機能追加

ODIP トランスフォーマにリクエストを受けて実行したプロセスごとにプロセス ID、リクエスト ID、管理単位名、開始日時、終了日時、終了状態及びエラーメッセージを記録して、ファイルまたはトランスフォーマリポジトリ内のテーブルに保存する機能が追加になりました。設定ファイル（`odip.ini`）に下表のオプションが追加になっています。初期設定ではプロセスログ出力は無効になっています。 `process.log` オプションに `file` または `table` を指定することで有効になります。

プロセスログに関する追加オプション

オプション	説明
process.log=<file table none>	file を指定するとログファイルに、table を指定するとトランスフォーマリポジトリ内の“PROCLOGTBL”テーブルにログを記録します。
process.log.file.interval=<pattern> process.log.file.maxbackupindex=<num> process.log.file.maxsize=<size> process.log.file.name=<filename> process.log.file.pattern=<pattern>	プロセスログをファイルに出力する場合に、ファイル名及びローリングに関する設定を行います。
process.log.table.maxrows=<num>	プロセスログをテーブルに出力する場合に、テーブル内に保存する最大行数を指定します。

(4) 一時テーブルを使用したファイル結合の改定

ODIP4.1(Build-id1040107004976)で追加した「Temporary テーブルを使用したファイル結合の対応」について次の変更を行いました。

- ① 「TEMPORARY テーブル」を「一時テーブル」に名称変更しました。
- ② 一時テーブルの CREATE INDEX 文をジョブログに出力するようにしました。
- ③ 一時テーブルを使用したファイル結合の対応データベース製品に SQLServer を追加しました。startjob コマンドの-tempds オプションには、Oracle、PostgreSQL、SQL Server のいずれかのデータソースを使用することができます。

(5) 固定長 COBOL の出力仕様変更

データセットカラムの COBOL データ型が“PIC S9(n)”、“PIC S9(n)V9(m)”のように、サイン付きゾーン 10 進数の値の出力に際し、エンコーディングの種類が CP930 など EBCDIC 文字セットの場合のサインディジットを F から C に変更しました。例えば、1234 は 16 進数表記で“F1F2F3F4”と出力していましたが“F1F3F3C4”に変わります。

(6) 「データ複写」の仕様変更

処理モード「データ複写」の管理単位実行で、複写先テーブルが既に存在し複写元テーブルとカラム数が異なる場合に、これまではエラーが発生して異常終了しましたが、次のように動作するようになりました。

- ・ 複写元テーブルの列数が複写先テーブルの列数よりも多い場合は、列の定義順に値を

設定し、複写元の余分な列は無視します。

- ・ 複写元テーブルの列数が複写先テーブルの列数よりも少ない場合は、列の定義順に値を設定し、複写元に対応する列のない複写先の列は NULL にします。

3. コンポーネント定義の改定

(1) クロス集計・グループ集計の Interim データセット廃止

クロス集計ユーザビュー、グループ集計を伴う管理単位の実行プロセスは、明細作成、データ分散の二つのジョブから成り、一時的に明細データを格納する Interim データセットが自動的に作成されていましたが、プロセスを構成するジョブを一つに統合し、Interim データセットを廃止しました。時系列演算を伴う管理単位の実行プロセスでは、これまでどおりクロス集計、グループ集計について Interim データセットが自動的に作成されます。

(2) “クロス集計併合入力データ”コンポーネントの廃止

入力データのコンポーネントの一つにクロス集計併合入力データがありましたが、特殊なケースでの利用に限定されていて、利用実績が殆どないことから、定義の新規作成、編集、実行の機能を停止しました。既存のクロス集計併合入力データがあれば、ODIP アドミニストレータを使用して定義を表示することはできますが、定義の変更、実行などの操作を行うことはできません。ODIP プロセスマネージャでは、クロス集計併合入力データを含む管理単位はプロセスとして表示されず、トランスフォーマリポジトリデータ作成の対象にもなりません。

(3) LIKE 関数の改定

演算子の第二オペランドの検索パターンに、属性及び文字列の結合式を記述できるようになりました。例えば、次の式はどちらも指定できます。

- ・ [顧客氏名] LIKE [顧客検索パターン文字列]
- ・ [顧客氏名] LIKE [顧客検索パターン文字列] || “%”

(4) MCHAR/SCHAR 関数による特殊文字の変換

MCHAR/SCHAR 関数（半角文字と全角文字との間の変換関数）は、シフト JIS 系では、単一の文字セット（Shift_JIS）に対応しており、他の文字セット（Windows-31J、X-IBM943 など）のデータを処理する場合、マイナス、チルダなどの特殊文字を変換ができませんでした。本バージョンでは、これら特殊文字の変換が可能になりました。設定及び関数の変更点は次のとおりです。

① 設定ファイル (odip.ini) のオプション追加

“job.mchar.mapping = <default|sjis|ms932|cp943>”のオプションが追加になりました。このオプションは、MCHAR、SCHAR両方の関数に有効です。“default”を指定すると、Java プラットフォームのデフォルト文字セットによって半角/全角の変換を行い、特定の文字セットを指定すると、指定した文字セットによって半角/全角の変換を行います。

② 関数の機能改定

関数の実行に使用する内部マップ表を拡張し、関数ごと、文字セットごとのマップ表を使用するようになりました。トランスフォーマエンジンは、設定ファイルのオプション指定に従ってマップ表を選択します。

4. CSV ファイル入出力の改定

ODIP には、CSV ファイル (TSV ファイルを含む) 入出力を伴う機能として、管理単位の実行、データベースツールのエクスポート・インポート、データ表示からの CSV 出力、トランスフォーマリポジトリのインポート、エクスポートなどがあります。これまでのバージョンでは、機能によって CSV ファイルの入出力のオプション、改行コード及び囲み文字の制御、NULL の取扱い、エンコーディングエラーの取扱いなどに若干の違いがありました。本バージョンでは、CSV ファイル入出力に関する仕様を共通化しました。主な変更点は次のとおりです。

(1) 設定ファイルのオプション追加

設定ファイル (odip.ini) に csv.enclose.option=<standard|minimum|all>オプションを追加しました。オプションには囲み文字の適用方法を指定します。デフォルトは standard で列のデータ型によって囲み文字の有無を決めます。minimum は囲み文字が真に必要な場合 (値が囲み文字または改行コードを含む場合、または値が空文字の場合) だけ囲み文字で囲みます。all は NULL 以外のすべての値を囲み文字で囲みます。

(2) CSV ファイル出力オプション

- ・ CSV ファイル出力オプションは、どの機能でもエンコードの種類、改行コード (LF、CR、CR+LF)、区切り文字 (カンマ、タブ)、文字列の引用符 (ダブルクォテーション、シングルクォテーション) を指定するようになりました。機能によって、各項目の選択肢が異なる場合があります。
- ・ CSV ファイルのデータソース定義ダイアログで、区切り文字に「カンマ」または「タブ」、文字列の引用符に「なし」を同時に選択できないようになりました。区

切り文字がデータ文字列に含まれるファイルの読み込みの誤動作を防ぐためです。区切り文字及び文字列の引用符は共に「なし」（フラットファイル）であるか、または共に「なし」以外の有効なオプションを選択しなければなりません。

(3) NULL と空文字の取扱い

ODIP は、CSV 形式における NULL と空文字を次のように区別します。管理単位での処理では以前のバージョンからの変更はありません。他の CSV ファイル入出力を行う機能に共通して同じ取扱いをするようになりました。

- ① 行の最初の区切り文字の前に値がない、区切り文字が連続する、行の最後の区切り文字の後に値がない列を NULL とします。次の例では、第 2 列及び第 4 列は NULL を表します。

例： “ABC”,,123.24,

- ② 空文字は、連続する二つの囲み文字で表します。次の例は、第 2 列及び第 4 列は空文字を表します。

例 “ABC”,”,123.24,”

(4) エンコーディングエラーの取扱い

ODIP は CSV ファイルの入出力時に、Shift-JIS、UTF-8 などの文字コードで保存されたバイト列と Unicode（1 文字を 4 バイトの整数で表します）との間の変換を行います。変換時に不正なバイト列がある場合またはその文字を Unicode にマップできない場合には、エンコーディングエラーが発生します。ODIP はエンコーディングエラーを検出すると、処理を中断してエラーが発生した文字列またはバイト列をメッセージとして表示します。

(5) 改行コードの取扱い

- ・ CSV ファイルの各行は改行コードで区切られているものとして読み込みを行います。改行コードは、CR (0x0d)、LF (0x0a)、CRLF (0x0d,0x0a) のいずれかで、どの改行コードでも読み込み可能です。旧バージョンでは一部の機能でオプションで指定された改行コードと実際のファイルの改行コードが異なる場合に、読み込みエラーが発生することがありましたが、どの機能でも入力時に改行コードの相違によるエラーは発生しなくなりました。
- ・ CSV ファイルの出力は出力オプションで指定した改行コードで行末を区切ります。出力時の改行コードの取扱いは旧バージョンからの変更はありません。ファイルの最後の行の行末にも改行コードが付きます。CSV ファイルのデータソースを用い

て、区切り文字「なし（フラットファイル）」かつ文字列の引用符「なし」の場合に限って最終行の行末の改行コードは付きません。

5. リッチテキストファイル出力の改定

ODIP アドミニストレータのリッチテキストファイル出力機能は、RTF1.6 以上の様式に対応して内部のライブラリ構成が大幅になりました。これに合わせて、新たに docx 形式の Word 文書の出力機能の追加及びいくつかの出力オプションの改廃を行いました。

(1) サードパーティコンテンツの変更

ODIP 製品がこれまでリッチテキストファイル出力に使用していたサードパーティコンテンツ (iText-2.1.7) の利用を止め、ODIP 独自に RTF1.6 の仕様に準拠した RTF の出力を行うようになりました。一方、Word 文書の出力に使用するサードパーティコンテンツ (poi-5.0.0) の利用を追加しました。

(2) 詳細設定のエクスポート、インポート機能追加

書式の詳細設定ダイアログに「エクスポート」、「インポート」ボタンを追加しました。エクスポートボタンを使用して詳細設定をファイルに保存し、インポートボタンによって保存したファイルの設定を読み込むことができます。

(3) 出力オプションの改定

① リッチテキストファイルダイアログに、出力先ファイルの種類の選択肢 (*.rtf、*.docx)、Word 文書用のコメント欄が追加になりました。入力したコメントは、Word 文書のプロパティ「コメント」に設定されます。

② 文書のプロパティの「見出しのインデントを有効にする」チェックボックスを廃止し、見出し、本文レベルごとにインデントを設定するようになりました。「見出しと本文のインデントを一括設定」ボタンを使用することで、全レベルのインデントをまとめて設定することができます。

③ 詳細設定オプションの改定

詳細設定オプションに次の変更を行いました。変更のあるオプション設定は、ODIP 製品のバージョンアップ後最初に起動するときに自動的に新バージョンの形式に変換されます。

(a) 書式の「フォントファミリー」は「フォント」になり、端末にインストールされているフォントの一覧を表示するようになりました。

- (b) 書式に「スタイル名」の指定欄を追加しました。Word 文書では指定した名前がそのままスタイル名になります。
- (c) 「見出しレベル」及び「本文レベル」を 1～6 から 1～9 に拡張しました。
- (d) 「標準の段落」を「標準」に変更しました。「標準」の書式は他のすべての書式の「基準となるスタイル」になります。
- (e) 表のプロパティの「表の配置」の選択肢は「左揃え」「中央揃え」「右揃え」だけになりました。旧バージョンの「自動」「両端揃え」は「左揃え」になります。
- (f) 表のプロパティの「表をページ内に収める」オプションを廃止し、代わりに「行の幅」オプションを追加しました。このオプションでは、「ページ幅に自動調整」「文字列の幅に自動調整」「ページ幅に対する割合を指定」のいずれかを選択します。
「ページ幅に対する割合を指定」を選択したときに割合 (%) 欄が有効になります。
- (g) 表のプロパティの「列をページ内に収める」オプションを廃止し、「行の途中で改ページする」オプションを追加しました。このオプションは初期値 (false) に設定されています。
- (h) 表のプロパティに「罫線のスタイル」を追加しました。実線、点線、波線などを選択できます。

(4) RTF 文書出力仕様の変更

文書の仕様を RTF1.6 に準じて出力します。主な変更点は次のとおりです。

- ① 見出しの項目番号の文字列は付加せず、RTF のナンバリング機能を使用します。
- ② パラグラフごとに個別に書式を指定せず、文書内に定義したスタイルを参照するようになりました。ただし、表のセルの書式は個別に指定します。
- ③ 文書のプロパティ (タイトル、作成者、コメント、分類、サブタイトル) に値を設定します。

(5) 出力項目の改定

- ① リッチテキスト・ファイルの出力で、データセットの列定義一覧に日付・数値書式の列を追加しました。

6. ヘルプメニューの表示変更

GUI 製品のヘルプメニューの動作環境の表示に共通して次の改定を行いました。

- ・ システム・プロパティダイアログのオペレーティング・システムタブにある行区切り文字 (改行コード) を ¥r (CR)、¥n (LF)、¥r¥n (CRLF) のいずれかで表示するよう

に変更しました。

- ・ システム・プロパティダイアログの Java タブに、デフォルトエンコーディング、ロケール（言語）、ロケール（国）、ロケール（バリエーション）を追加しました。

7. その他の修正

(1) ODIP アドミニストレータの改定

① データセット

- (a) データセットエディタの「スキーマ名」、「テーブル/ファイル名」の入力と表示を次のように修正しました。
 - ・ 「デフォルトのスキーマを使用」のチェックをオンにすると、スキーマ名欄は編集不可になります。スキーマ名の文字列は一時的に保存され、画面のリフレッシュ時にクリアされます。
 - ・ 「デフォルトのスキーマを使用」のチェックをオフにすると、スキーマ名欄は編集可になります。スキーマ名を空欄のままにすると、画面のリフレッシュ時に「デフォルトのスキーマを使用」のチェックはオンに変更されます。
 - ・ 管理単位の実行ダイアログで、入出力データセットのデータセット情報ダイアログから、スキーマ名、テーブル/ファイル名を変更したときに、同じデータセットのデータセットエディタが開いていれば、エディタ上のスキーマ名、テーブル/ファイル名は変更後の状態にリフレッシュされます。
- (b) データセットウィンドウのカラム定義一覧では、カラムのデータタイプに属性のものと異なるデータタイプを指定すると、データタイプの上書きを表すアイコン  表示されますが、データセットカラムのデータタイプを上書きした後で、属性の定義を変更して上書き後のデータタイプにすると、データセットウィンドウの上書きを表すアイコンが消えずに表示されていました。属性の変更と同時にデータセットウィンドウの更新するように変更しました。
- (c) COBOL 固定長ファイル用の FILLER 定義について、次の修正及び変更を行いました。
 - ・ ファイルメニューの「プロジェクト・ファイルのインポート」を行うときに、インポート元とインポート先に同じ名前のデータセットがあり、定義内容が同じであるにも関わらず、一方は一度定義した FILLER をすべて削除し、他方は FILLER を一度も定義していない場合に変更ありのアイコンが表示されました。
 - ・ プロジェクトをリポジトリにコミットすると、FILLER を含むデータセットは、定義変更がなくても変更履歴が記録されていました。不要な履歴を保存しないように修正しました。

- ・ 変更点タブの「詳細」ボタンで表示する変更点の詳細表示ダイアログで、データセットが FILLER を含む場合は、「カラム番号」の代わりに「FILLER を含むインデックス」を表示するようになりました。この項目には、カラムと FILLER の両方を合わせたリスト上の 0 から始まる連続する番号を表示します。
- ・ FILLER が一つもないデータセットに FILLER を追加して削除すると、変更点タブに列定義の変更が残存する問題がありましたが、残存しないように変更しました。
- ・ FILLER のないデータセットにプロジェクトのインポートで FILLER を追加すると、変更点には、追加した FILLER だけが表示されていましたが、FILLER の追加によって「FILLER を含むインデックス」が変わったカラムについても変更点に表示するように変更しました。

② 入力データ

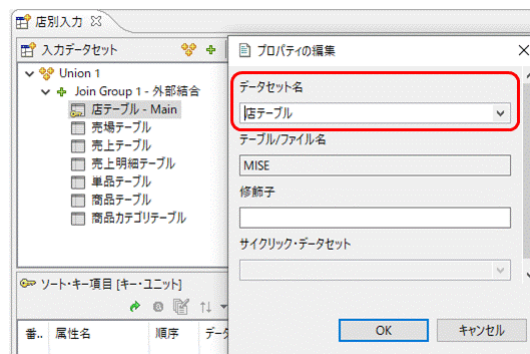
- 入力データの属性選択タブで、「属性の別名指定」を行う際の選択肢の表示上の不具合を修正しました。同じ属性が異なるデータセットのカラムに指定されていると、別名指定の直後に指定と異なるカラムが選択肢の一覧から消されることがありました。
- 一つのフォルダ内の最大フォルダ数及び最大管理単位数（どちらも 99 個）のチェックを定義の検査に追加しました。これはフォルダ、管理単位のコピー、ペーストによって最大数を超えて作成することができる不具合に対応したものです。既に最大数を超える定義が存在する場合は、本バージョンでは定義の検査でエラーが検出されます。
- 一つの管理単位内に、ピボット変換（縦横変換・横縦変換）とグループ集計の両方を定義することができないように、入力時のチェック及び定義の検査を変更しました。
- 入力データ定義において、入力データセットを異なる Join Group に移動すると、移動前の Join Group に定義されていた入力抽出条件が定義上削除された状態になり、不正な参照が検出される問題を修正しました。既にこの問題によって不正な参照が発生している場合には、定義の検査時に不正な参照を検出し、自動的に修復を行うように改定しました。修復時には、問題点タブに次の例のような警告メッセージを表示します。

例： 「"Join Group 1 - 外部結合"内のデータセット"売上明細テーブル"への壊れた参照をもつ 3 行の抽出条件式が削除されました。」

- 複数の Union を定義した入力データのある Union の選択属性タブで、「全属性選択」を押したときに、他の Union の導出演算で導出される属性は自動追加の対象になりませんでした。また、別名指定された属性と同じカラムも自動追加されませんでした。

た。これらの属性は、全属性選択によって追加されるようになりました。

- (f) 入力データに二つ以上の Union があるとき、いずれかの Union の選択属性タブで「未使用属性の選択解除」ボタンを押すと、ユーザビュー、出力前抽出条件などで参照されている属性も合わせて解除されました。これは、ユーザビュー、出力前抽出条件から選択属性への ODIP の内部的な参照関係は、複数 Union の同じ属性はいずれか一つの Union のものだけを保持していて、それ以外の参照関係を保持していない選択属性は、未使用と判定されました。本バージョンでは、入力データの選択属性とユーザビューなどの多対多の参照関係を識別し、「未使用属性の選択解除」で参照のある属性を選択解除しないように変更しました。一方、旧バージョンでは、上のように未使用と判定される選択属性を削除することができましたが、本改定によって参照を識別された選択属性は、どの Union でも削除できなくなります。
- (g) 入力データセットの「プロパティの編集」ダイアログ（下図）では、管理単位を構成するデータセットであっても他のデータセットに変更することができるようになりました。定義の整合性はこれまでどおり、定義の検査によってチェックされます。



③ 導出演算

- (a) 導出演算式で TRIM、LTRIM、RTRIM 関数の第一引数に CONVERT、REPLACE など一部の関数を用いた式を使用すると、第二引数を指定できなくなる問題を修正しました。
- (b) 導出項目にデータタイプの長さが 1 の属性を、初期値に“null”を指定すると、定義が正しいにも関わらず「初期値に設定された値の長さが出力属性の長さを超えています」としてエラーが表示される問題を修正しました。
- (c) 出力前導出演算の新規作成時に、初期値に導出項目の長さを超える値を指定しても、エラーが検出されず、変更時にエラーが表示される問題を修正しました。
- (d) 導出演算の初期化処理で AVG、MIN、MAX 関数に入力属性を含む引数を二つ指定したときのエラーメッセージを「初期化処理では、集計関数の引数を二つ指定する場合、引数に入力データの選択属性を指定することはできません。」に変更しました。

- (e) 導出演算の初期化処理では、COUNT 関数の引数は一つだけしか指定できないように修正しました。以前のバージョンでは引数を二つ指定できましたが、二つ目の引数は使用されませんでした。二つの引数を指定した定義があれば、定義の検査でエラーを検出します。

④ データソース情報

プロジェクトメニューのデータソース情報ウィンドウに次の改定を行いました。

- (a) [Ctrl+Shift+c]キーには、ヘッダー行と選択行をクリップボードにコピーする機能が割り当てられていましたが、データソースのタイプによっては機能しない場合があります。このキーの動作は ODIP の標準的なものと異なっていて、利用者に分かりにくいとため、キーの割り当てを解除しました。
- (b) データソース情報のコピー&ペーストで、CSV ファイル、固定長テキストファイル、固定長 COBOL ファイルの項目のうち画面に表示されない項目はペーストされませんでした。データソース情報画面に、すべての項目を表示するように変更しました。

⑤ データ名称エクスポート・インポート

ファイルメニューのデータ名称エクスポート、インポートにおいて、データセットのインポート時の確認ダイアログ表示及び更新上の次の改定を行いました。

- (a) 固定長 COBOL ファイル用の FILLER を含むデータセット定義のエクスポート、インポートを行うと、インポートの際に更新オプションを有効に指定しても、FILLER の削除が正しく反映されない問題を修正しました。
- (b) データセットのインポートを行うときに、確認画面に「カラムタイプが、既に定義されているデータタイプのカラムタイプ"ts"と異なっています。この入力値は無視されます。」のように、実際には問題のないインポートデータに警告メッセージが表示される問題を修正しました。
- (c) データセット定義をインポートする際に、COBOL データ型、の指定のない定義に相違が表示される場合がある問題を修正しました。
- (d) データ名称インポートを用いてデータセットをインポートする際、インポート対象のデータタイプ、長さなどに表示される「この入力値は無視されます」のワーニング・メッセージが、日付/数値書式にも表示されるように変更しました。
- (e) データセット・インデックスのインデックス名が小文字に変換されていました。ファイル上のインデックス名をそのままインポートするように修正しました。
- (f) データ名称エクスポート及びデータ名称インポートのダイアログから「ヘルプ」ボタンを削除しました。エクスポートファイルのレイアウト情報はヘルプメニューのヘルプコンテンツを参照してください。

⑥ プロジェクトインポート

ファイルメニューのプロジェクトインポートについて、次の改定を行いました。

- (a) データセットを選択して「選択した項目を上書きする」を押した後、任意のフォルダを選択して「選択した項目を上書きする」を押すと、先に選択したデータセットに、実際にはエラーではないにも関わらず重複する列があるとのエラーメッセージが表示される問題を修正しました。
- (b) データセット及びカテゴリ階層に実際には相違がないにも関わらず相違が表示される場合がある問題を修正しました。
- (c) カラムの定義順だけが変わっているデータセットをインポートすると、インポート先のプロジェクトでのデータセット定義が変更されているにも関わらず、リポジットリへのコミットで変更が反映されない問題を修正しました。

⑦ データ表示

データセットウィンドウのデータベースツール及び実行ダイアログのデータ表示機能について次の改定を行いました。

- (a) 固定長テキストファイルのデータソースで「NULL を表す文字」を指定してデータ表示を実行すると、例外が発生してデータを表示できない問題を修正しました。
- (b) ファイル（CSV、固定長テキスト、固定長 COBOL）のデータを表示したとき、2 ページ目以降に進んでから最初のページに戻ると、最初の一行が表示されず、次ページ以降行がずれて表示される問題がありました。
- (c) PostgreSQL 上のテーブルデータ表示で、例えばテーブルが存在しないなどのエラーが一度発生すると、それ以降のデータベースへのアクセスは「transaction aborted...」のエラーになる問題を修正しました。
- (d) ユーザビューモードでクロス集計表のデータを表示実行すると、一部の行が表示されない場合がある問題を修正しました。
- (e) COBOL 固定長ファイルのデータを表示すると、小数部に 16 進数でなくゼロ（例：“3030.00”）が表示される問題を修正しました。

⑧ データベースツール

データセットウィンドウのデータベースツールの機能に次の改定を行いました。

- (a) データ複写を用いてテーブルにデータを複写するときに、複写先テーブルのスキーマとして、ダイアログで指定したスキーマ名、データセットで指定したスキーマ名ではなく、デフォルトのデータソースのユーザ名を使用していました。複写先のスキーマ名を次の(i)～(iv)の順に優先して使用するように変更しました。また、複写先のテーブルを作成するときに、プリファレンスの「スキーマとテーブルスペース」ページで指定したテーブルスペース名及びインデックススペース名が有効になるように修正しました。

- (i) データ複写画面のテーブル名をスキーマ名で修飾（例：odipschema.table1）した場合はそのスキーマ名（例：odipschema）
 - (ii) データセットウィンドウで、「デフォルトのスキーマを使用」を無効にして、スキーマ名を指定した場合はそのスキーマ名
 - (iii) プリファレンスの「スキーマとテーブルスペース」ページでスキーマ名を指定した場合はそのスキーマ名
 - (iv) データソース情報のユーザ
- (b) 複数のデータセットを選択してトランスフォーマサーバでデータ複写をリモート実行すると、正常に終了した場合でもダイアログに警告やエラーが表示されることがありました。終了状態を正しく表示するように修正しました。
- (c) スキーマチェックを使用して、データセット定義とデータベース内のテーブルをチェックしたときに、双方のカラム数またはカラムの定義順の違いによって、タイムスタンプ型の精度（小数桁数）の相違がエラーとして検出される場合とされない場合とがありました。スキーマチェックに次の二つの変更を行いました。
- ・ カラム名が同じで列の定義順が異なるカラムはエラーを表示していましたが、入力データセットはカラム番号が異なる旨のワーニングを表示し、出力データセットはエラーを表示するようになりました。
 - ・ あるカラムをデータセットにタイムスタンプ型（ts 型）、SQL Server のテーブルに DATETIME 型と定義した場合、小数桁数の指定はデータセット定義では有効に、テーブル定義では無効になります。両者の型について、どちらか一方の小数桁数が無効であれば、小数桁数をチェックしないように変更しました。
- (d) データ複写のオプション追加
- ODIP アドミニストレータのデータベース・ツールのデータ複写のダイアログに「同じ列名の列値をコピーする」オプションを追加しました。オプションの指定によって、次のように動作が変わります。
- ・ オプションを有効にすると、複写元と複写先の列名が同じ列の値を複写します。複写元テーブルと複写先テーブルの列数が異なる場合、または列の定義順が異なる場合でも複写を行うことができます。同名の列が一つもない場合はエラーになります。
 - ・ オプションを無効にすると、複写元テーブルと複写先テーブルそれぞれの列の定義順に従って列値を複写します。双方の列数が異なっている場合は、定義順の対応する列があるものだけを複写し、対応する列がなければ NULL になります。定義順に対応する列のデータ型、有効桁数の相違によって、列値を複写できない場合はエラーになります。

⑨ 実行

- (a) 時系列演算を含む管理単位の実行において、Interim データセットのデータソースに

ファイル（CSV、固定長テキスト、固定長 COBOL）を指定すると、エラーを表示するようになりました。以前のバージョンではファイルの指定はできますが、そのまま実行すると、処理は異常終了しました。ODIP アドミニストレータでは実行時に、ODIP プロセスマネージャではデータセットの定義の検査でエラーが検出されます。

- (b) ODIP アドミニストレータのユーザビューのプロパティで、出力データセットに「ユーザビューに合わせて更新」を設定している管理単位を実行したとき、ユーザビューとデータセットのレイアウトの相違を自動修正して実行を継続するための確認のダイアログが表示されない問題を修正しました。
- (c) アドミニストレータで「データセットの入出力依存関係に従う」オプションを指定して、インタラクティブ/バックグラウンドモードで実行しても、ジョブ実行順に依存関係が反映されない問題を修正しました。
- (d) 情報ウィンドウの実行ログで、ログ詳細ダイアログを開いたまま、他の実行ログの行に移動すると、ダイアログのメッセージ欄に移動前のメッセージが残存する問題を修正しました。

⑩ 定義の検査

- (a) 処理モードが追加・更新・削除の管理単位の導出演算(開始)または導出演算(終了)に SETMODE 関数を定義すると、定義の検査で警告メッセージが出る問題を修正しました。
- (b) グループ集計で、グループ型の設定項目、階層型の親子識別属性を、入力データの選択属性から削除すると、定義の検査でエラー検出されないにもかかわらず、実行時に表示時に例外が発生して処理が異常終了しました。定義の検査でエラーを検出するように修正しました。
- (c) カテゴリ階層定義に次の定義の検査と検査結果表示を追加しました。この検査対象のエラーは通常の入力では発生しませんが、旧バージョンで作成したプロジェクトに残存する可能性があります。このエラーの有無は管理単位の処理に影響しません。データ名称のインポートの際に重複エラーが表示されます。定義の検査でこのエラーが検出された場合は、重複する定義の一方を削除してください。
 - ・ 定義の検査（全体）の実行時に、カテゴリ階層の重複定義（一つの入力カテゴリに対応する出力カテゴリが、複数の異なるレベルで割り当てられている）を検出します。エラーを検出した場合は問題点タブに、例えば「カテゴリ階層[年齢層区分]-[管理年齢層区分]の<Level-2>に定義された[40代]-[中年]の割り当てが重複しています。」のようにメッセージを表示します。
 - ・ 問題点の行をダブルクリックするとカテゴリ階層ダイアログを開きます。
 - ・ カテゴリ階層ダイアログの左方の入力属性と出力属性のツリーのアイコンにエラ

ーを表示します。

⑪ その他

- (a) プロジェクトメニューからカテゴリ階層ウィンドウを開き、リッチテキスト・ファイルを出力するダイアログで、出力対象に“選択されたカテゴリ階層”を指定して出力を行うと例外が発生する問題を修正しました。
- (b) 明細ユーザビューのエディタで、「データセットに合わせてユーザビューを更新」する操作の直後に、フォルダタブツリー上の変更表示 (>) が更新されませんでした。変更操作の直後にツリー表示を更新するように修正しました。

(2) ODIP オペレーションマネージャの改定

- ① テーブル/ファイル管理タブから COBOL 固定長ファイルをデータソースとするデータセットを選択してテーブル作成を実行すると、NULL 例外が発生する問題を修正しました。
- ② ツールメニューのデータソース情報ウィンドウから編集ダイアログを開き DBMS を変更したときに、ホスト名、ポート番号の設定のない DBMS を選択すると、これらの入力フィールドは、グレースアウトされているにもかかわらず、変更前の DBMS 名への設定値がそのまま表示されていました。入力フィールドクリア時の問題を修正しました。
- ③ ジョブ起動ダイアログの「パラメタ」項目の表示位置を修正しました。
- ④ 入力データのリレーションに関数を用いた式の記述が一定の長さを超えると、トランスフォーマリポジトリ作成時に「テーブル"REPKB135"の挿入時に例外が発生しました。管理単位名: ...」のエラーメッセージが表示されて実行できませんでした。トランスフォーマリポジトリの列定義を変更しました。
- ⑤ 入力データのリレーション属性、再帰開始行の条件、再帰開始行の値に SUBSTRB 関数を定義した管理単位を、SQL の対応する関数のサポートのないデータソース (H2、PostgreSQL、ファイル) を使用して実行した場合には、"This operation is not supported for 'dbms'. #Substrb"のエラーメッセージを出力するように変更しました。

(3) ODIP リポジトリマネージャの改定

- ① 更新履歴ウィンドウで変更内容の表示を行い、選択したバージョンがユーザ関数定義の変更を含む場合に、例外発生ダイアログが表示され変更内容を表示できない場合があります。リポジトリの変更内容収集時の問題を修正しました。

(4) ODIP プロセスマネージャの改定

- ① GUI を起動してサイクル表示ツリーのプロジェクトを選択していない状態で proc ファイルを開いたとき、ツリー上のサイクルを選択するまで右側のプロセス一覧には何も表示されませんでした。プロジェクトを選択状態にすることで、最初から一覧にプロセスが表示されるようになりました。
- ② “プロセス作成”機能によって複製したプロセスを、プロセス一覧から削除すると、内部の例外の発生によって削除できない問題を修正しました。
- ③ ツールメニューのテーブル・スキーマのチェックを実行すると、データセット情報で指定したスキーマ名が適用されず、チェック対象テーブルが存在しているにも関わらずエラーが発生する問題を修正しました。
- ④ プロセス一覧タブで、入出力の連鎖が循環する一連のプロセスが、“随時”サイクルのプロセスを含む場合に、先頭プロセスを設定できない問題を修正しました。
- ⑤ ツールメニューからリスト表示を開いた状態でプロセスマネージャを終了すると、リスト表示ダイアログが表示されたままになる問題を修正しました。
- ⑥ ツールメニューのトランスフォーマ・リポジトリ情報で表示される一覧画面で、タイトル列をクリックしてソートを行うと、選択した列と異なる列でソートされる問題を修正しました。
- ⑦ トランスフォーマリポジトリ作成を実行して既存のトランスフォーマリポジトリを更新すると、同じ出力データセットに対するリカバリ「受渡」が、「サイクリック」及び「サイクル間サイクリック」の両方にあると、ODIP オペレーションマネージャで「サイクリック」に登録したリカバリ条件が、「サイクル間サイクリック」に登録した条件に置換して更新される問題を修正しました。
- ⑧ ODIP4.1 以降のバージョンでリカバリ受渡、保存、読込のいずれかの処理を伴うジョブの実行において、受渡、保存、読込のデータソースにファイル（CSV、固定長テキスト、固定長 COBOL）を使用すると、リカバリ処理で例外が発生して異常終了する問題を修正しました。
- ⑨ データセット情報ダイアログに次の修正を行いました。
 - ・ データセットを選択してデータセット編集ダイアログを開いたまま親となるデータセット情報ダイアログを閉じると、内部的な例外が発生して再度ダイアログを開くことができなくなりました。データセット編集ダイアログを開いているときに、親となるデータセット情報ダイアログを閉じることができなくなりました。
 - ・ 「Cancel」ボタンの表記は「キャンセル」になりました。

- ⑩ ツールメニューの「テーブル・スキーマチェック」を実行して、テーブルスキーマ・チェック・エラータブにエラーメッセージが表示されているとき、エラーの行を選択してダブルクリックすると、そのデータセットの入出力を行うプロセスに「先頭」チェックボックスにチェックが入る場合があります。エラーメッセージ行のダブルクリックは無効になりました。

(5) ODIP リポジトリサーバの改定

- ① ODIP アドミニストレータからの次の操作によって、コミット済みの定義の一部が上書き消去される場合がある問題を修正しました。
- (a) ODIP リポジトリサーバから定義をチェックアウトしてプロジェクトファイル (*.odp) に保存した後にリリースする。
 - (b) ODIP リポジトリサーバを停止して data¥repdata を過去にバックアップした repdata でファイルごと置換する。
 - (c) ODIP リポジトリサーバを起動して ODIP アドミニストレータで定義をチェックアウトする。
 - (d) (a)で保存したプロジェクトファイル (*.odp) を ODIP アドミニストレータで開く。
 - (e) (d)に定義を追加してコミットする。

(6) ODIP トランスフォーマの改定

- ① トランスフォーマエンジンの修正
- (a) 計算式の除算で、分母が固定値 0 の場合、分母が外部変数属性で実行時にパラメタで 0 を渡した場合、トランスフォーマエンジンは導出演算の初期化でゼロ除算によるエラーを検出して異常終了しました。条件式に分母が 0 でない条件があってもやはりエラーになりました。初期化処理ではエラーを検出せず、計算実行時にエラーを検出し、job.zero.div.zero オプション設定に従って動作するように修正しました。
 - (b) 導出演算の計算式で、EXIT 関数の第 2 引数を省略して「#Exit(251)」のように定義して処理を実行すると、実行時に Null 例外が発生して処理が異常終了する問題を修正しました。
 - (c) AVG 関数の結果が引数に指定した属性の小数桁数に合わせて切り捨てられる問題を修正しました。例えば、Avg ([属性 A]) の式について、属性 A のタイプが p(10,0)、導出項目の属性のタイプが p(12,2)の場合、引数の p(10,0)に合わせて平均値の小数以下が切り捨てられていました。計算過程では小数以下 38 桁まで求め、導出項目の属性の桁数に合わせて値を設定するように変更しました。
 - (d) 導出演算の計算式で CONCATB 関数の引数に数値型の属性を指定したとき、実行時の数値型属性の引数の値が負で、それを符号を含む文字列にした長さがデータタ

イブで指定した長さと同一であるときに、文字列の桁数が大きすぎる旨のエラーになっていました。符号つきの場合でも、文字列の長さがデータタイプの長さ以下であればエラーにならないように修正しました。また、長さがオーバーフローした場合に次のエラーメッセージを表示するようになりました。

「カラム[<column_name>]の値[<value>]の桁数[<value_length>]がデータセット定義[<defined_length>]に対して大きすぎます。」

- (e) 管理単位内に複数の明細ユーザビューがあり、そのうちの一つに出力前削除条件の指定があると、実行時にエラーが発生することがありました。削除条件初期化の問題を修正しました。
- (f) ODIP4.1.3 以降、SUBSTRB/CONCATB 関数で使用する文字コードが全 Union で最後に読み込まれるファイルの文字コードになっていましたが、Union ごとに最初に読み込まれるファイルの文字コードを使用するように修正しました。
- (g) 階層型グループ集計の処理で、親の結果行が複数行出力される場合がある問題を修正しました。
- (h) ロードタイプが「行追加」のユーザビューでファイルのデータソース（CSV、固定長テキスト、固定長 COBOL）にデータを出力する場合に、既存データ最終行の行末に改行コードがないときには、改行コードを付加した上で出力するように変更しました。
- (i) 出力データの PKEY による一意性制約チェックにハッシュ索引を使用して、大量データ（例えば一意なキー値が 1 千万個を超える場合）を処理し、キー値に対応するハッシュ値の同値によるコリジョンが多数発生すると、ページオーバーフローエラーによって処理が異常終了する場合があります。エラーが発生しないようにページングの方式を修正しました。これに合わせて、ハッシュ索引を用いたキー値のチェックの性能改善を図りました。
- (j) 明細ユーザビューの出力属性に対応する“元属性/設定値”に外部変数属性を指定した定義を実行すると、入力データソースがデータベースの場合には、パラメタで指定した外部変数の値が出力され、入力データソースが CSV、固定長などのファイルの場合には、NULL が出力されました。入力データソースがファイルの場合にも、パラメタで指定した外部変数の値が出力されるように修正しました。
- (k) データソース情報の固定長テキストファイル編集ダイアログにおいて、「小数点をつける」オプションを有効にしたとき、エンコーディングの種類にデフォルト以外のものを指定すると、「1 行目のカラム[COL1]の値[+1+]が数値ではありません」というように小数点が正しく識別できずに、エラーが発生しました。小数点を正しく識別するように修正しました。

② データベースアクセス

- (a) DB2 for luw データベース上のテーブル名が全角文字を含み、JDBC 設定ファイル (jdbcconfig.xml) の<enclosemultibyte>オプションを true に設定している場合に、テーブルへのロード処理で自動生成する CLP ファイルのテーブル名がダブルクォテーションで囲まれておらず、不正なテーブル名としてロードコマンドがエラーになっていました。CLP ファイル生成時に全角文字を含むスキーマ名及びテーブル名をダブルクォテーションで囲むように修正しました。
- (b) SQL Server データベース上のテーブル名が全角文字を含み、JDBC 設定ファイル (jdbcconfig.xml) の<enclosemultibyte>オプションを true に設定している場合に、テーブルへのロード処理でロードコマンド (bcp) に指定するテーブル名の囲み文字の相違によってエラーが発生しました。bcp コマンド生成時に全角文字を含むスキーマ名及びテーブル名を [] (中括弧) で囲むように修正しました。
- (c) Cache の短縮名を Cache から cache に変更しました。DBMS ごとのプロパティファイルは Cache.properties から cache.properties になります。

③ ログ

ジョブログ (odipjob.log)、トレースログ (odip_trace.log) の出力メッセージを次のように改定しました。

- (a) 時系列演算タスクの初期化開始時のメッセージを「グループ演算初期化 開始/終了」から「初期化 開始/終了」に変更しました。
- (b) 出力データの初期化、一意性チェックなど、出力データセットに関連するメッセージには、共通して「テーブル/ファイル[<tablename>]」を先頭に表示するように変更しました。
- (c) 時系列演算のデータ収集件数及び出力処理時間のメッセージ出力を追加しました。
- (d) 明細データ分散にデータ収集件数メッセージ出力を追加しました。
- (e) ジョブログ中の行数、件数の値が一部カンマ編集されていました。カンマ編集のない数字を表示するようにしました。
- (f) startjob のデータソースの置換オプションを使用して、固定長 COBOL に変更した場合にジョブログのメッセージにデータソース名だけが出力されていました。他のデータソースと同様に、括弧内に付加的な情報を出力するように変更しました。
- (g) odip_trace.log に出力される Approximate elapsed time が実際の経過時間よりも 9 時間多く表示される問題を修正しました。

④ コマンド

- (a) トランスフォーマサーバのポート番号に 32768 以上の整数値を用いると、startjob などのコマンドの実行時のコマンドプロンプトのメッセージに、実際と異なるポート番号が表示される問題を修正しました。例えば、ポート番号 41302 のときに startjob

を実行すると、「connected to Transformer Server at host:imkk023.imkk.local, port:-24234.」とメッセージ表示されていました。

- (b) repimp コマンドで過去のバージョンで廃止された LOGMNGTBL、OPTIONTBL、REPKB181、REPKB186 の各テーブル作成を停止しました。
- (c) repimp コマンドに-patch オプションを指定して実行すると、トランスフォーマリポジトリのデータベースが SQL Server の場合に、ODIP が発行する SQL 文がエラーになる場合があります。動的 SQL 文生成時の問題を修正しました。
- (d) repimp コマンドに-patch オプションを指定して実行すると、導出演算の初期値に定義された“null”の文字列が、NULL になる問題がありました。“null”は文字列としてそのままインポートされるように修正しました。
- (e) repcomp コマンドに-remote オプションを指定して、トランスフォーマリポジトリの比較を連続して複数回実行すると、前回実行分の比較データがメモリ上に残存していることで、実際と異なる相違点が検出される場合があります。メモリ上に残存する比較データを参照することのないように修正しました。
- (f) config コマンドのオプション追加
トランスフォーマサーバの設定ファイル (odip.ini) の自動作成コマンド (config) に、更新オプション (-u|-update) を追加しました。このオプションを指定して config コマンドを実行すると、本バージョンで追加になったオプションが設定ファイルに追加されます。既存のオプションの設定値は保持され、追加オプションにはデフォルト値が設定されます。-c|-comment オプションを指定すると、全オプションのコメントが最新の情報に置換されます。
- (g) startjob コマンドの-datasource (-ds) オプションにファイル (CSV、固定長テキスト、固定長 COBOL) のデータソースへの置換を指定して、処理モード「追加・更新・削除」の管理単位に対応するプロセスを実行すると、NULL 例外の発生により適切なエラーメッセージが返却されな場合があります。次のエラーメッセージを返すように修正しました。
「"<tablename>"のデータソース"<datasource_from>"は-datasource オプションで "<datasource_to>"に指定されています。追加・更新・削除管理単位の出力データにファイルを指定することはできません。」
- (h) encrypt コマンドの実行結果に Web API の realm.properties 用の MD5 ハッシュ値、OBF 暗号化形式の文字列を追加しました。コマンドの使い方に変更はありません。

⑤ トランスフォーマリポジトリ

- (a) repimp コマンドの-patch オプションを使用したときに、更新対象となるテーブルのキー構成列による索引を作成するようになりました。この機能はトランスフォーマリポジトリのデータベースに Oracle、DB2、PostgreSQL、SQL Server、Symfoware

のいずれかを使用する場合に有効です。

- (b) 入力データのソートキー項目に定義したカラムのカラム名の長さが半角文字で 40 文字を超えると、トランスフォーマ・リポジトリ作成でエラーが発生しました。ソートキー項目を格納するカラムの長さを 40 から 255 に拡張しました。

B. バージョンアップによる影響

1. 定義内容・検査に関する影響

① “クロス集計併合入力データ”コンポーネントの廃止

既存の ODIP プロジェクトにクロス集計併合入力データの定義があり、その定義を引き続きご使用になる場合は、誠に恐れ入りますが弊社製品サポート (support@imkk.jp) にお問い合わせください。

② 既存のプロジェクトファイルに一つのフォルダに 99 個を超える管理単位またはフォルダがある場合、定義の検査でエラーが検出されます。管理単位を他のフォルダに移動するなど、99 個以内に収まるように定義を変更してください。

③ ピボット変換とグループ集計の両方を定義した管理単位がある場合、定義の検査でエラーが検出されます。該当する管理単位は、実行時にエラーが発生するか正常終了しても期待する結果を得ることができません。どちらか一方を削除するなどの定義変更を行ってください。

④ 導出演算の初期化処理で COUNT 関数に二つの引数を指定した定義がある場合、定義の検査で「引数が多すぎます。#Count(属性名)」のエラーが検出されます。第二引数は無効ですので、削除してください。

⑤ カテゴリ階層に重複する定義があると、定義の検査でエラーが検出されます。エラーのあるカテゴリ階層を表示し、同じ入力カテゴリと出力カテゴリの組が異なるレベルに指定されて定義を探して、一方を削除してください。

2. 処理実行結果に関する影響

① クロス集計・グループ集計の Interim データセット廃止の影響

(a) 出力順

従来のグループ集計が定義された処理は、一度 Interim データセットに出力した全データを管理単位属性順に読み込みながらユーザビューに出力していました。Interim データセットの廃止により、出力の順序が、入力データの Union 順、ソート項目順、グループ集計の集計行となるように変更されました。そのため、定義によってはファイルに書き出した場合の出力順が旧バージョンと結果が異なる可能性

があります。また、クロス集計の Interim データセットの廃止により、クロス集計の出力データセットへの出力順も、明細処理と同じく Union 順になります。

(b) 行番号属性

ユーザビューのプロパティに、行番号属性の指定がある場合、次のいずれかに該当する定義では、ジョブの統合によって行番号の出力値が旧バージョンと異なる可能性があります。

- ・ グループ集計を伴う明細ユーザビューについて、グループ集計結果行の行番号が異なる可能性があります。
- ・ クロス集計ユーザビューについて、管理単位と異なる連番単位を指定している場合、管理単位の出力順の違いによって行番号が異なる可能性があります。

(c) 導出演算の終了処理

旧バージョンでは、明細処理/クロス集計の最後に導出演算の「終了処理」が実行され、その後グループ集計が実行されていました。Interim データセットの廃止により、明細処理/クロス集計、グループ集計が行われた後に導出演算の「終了処理」が実行されるようになります。出力データに影響はありませんが、XFUNC 関数を使用したロギングなどを行っている場合、実行のタイミングがグループ集計前から、グループ集計まで行われた処理の最後になります。

(d) グループ集計の設定値<入力データ>

グループ集計の入力行は、Interim データセットを管理単位属性でソートした順序から、入力データ定義どおりの順序に変更されました。そのため、グループ集計の「設定値」が”<入力データ>”で、入力データには複数の Union が定義されているとき、旧バージョンでは全 Union で管理単位順の先頭レコードの値がセットされていましたが、本バージョンからは Union1 の先頭レコードの値がセットされます。同じ結果を出力するには、固定値を設定するか、要件によっては一度出力したユーザビューを入力とした管理単位、グループ集計を追加し、追加したグループ集計で<入力データ>を指定してください。

② Oracle・SQL Server のロード処理改定

バージョンアップによって、改行コードを含む文字列を Oracle 及び SQL Server のテーブルにロードした結果が異なる場合があります。旧バージョンの動作を継続したい場合は、config ディレクトリの DBMS 設定ファイル（Oracle は oracle9i.properties、SQL Server は sqlserver2008.properties）に次の行を設定してください。ファイルがない場合は、ODIP トランスフォーマー導入ディレクトリの config/jdbcamples ディレクトリから同名のファイルを複写してください。

```
loader.remove.linefeed = true
```

③ 固定長 COBOL ファイルの出力仕様変更

次のすべてに該当する処理がある場合、処理の出力ファイル上のサイン付きゾーン十進数のサインディジットは 0xF から 0xC に変わります。出力ファイルの利用先で変更後のファイルが読み込み可能であることをご確認ください。

- ・ 固定長 COBOL ファイルへの出力を行う管理単位の処理がある。
- ・ データセットの COBOL データ型に、S9(n)、S9(n)V9(m)のように、サイン付きゾーン十進数を指定している。
- ・ データソースで、固定長 COBOL ファイルのエンコーディングの種類に、CP930、CP939 など EBCDIC 文字セットのエンコーディングを指定している。

④ 「データ複写」の仕様変更

データ複写の管理単位による処理で、複写元と複写先のテーブルの列数が異なる場合でも処理は正常終了するようになります。このケースでの異常終了を前提にバッチ処理を運用することはできませんのでご注意ください。

⑤ CSV ファイル入出力の改定

CSV ファイルを入出力する処理では、バージョンアップによって次の処理結果の相違が発生します。

(a) 前後に半角スペースを含む文字列の読み込み

囲み文字で囲まれていない文字型の列値の前後の半角スペースは、旧バージョンでは削除されましたが、本バージョンでは値としてそのまま読み込みます。次の例 1 の行の第 2 列を読み込んだ値は、旧バージョンでは“DEF”、本バージョンでは“ DEF ”になります。例 2 のように囲み文字で囲まれた値は、どちらのバージョンでも“ DEF ”になります。

例 1 : ABC, DEF ,

例 2 : “ABC”, “ DEF ”,

(b) 数値型の小数点以下の表示

本バージョンでは、データセットで数値型 (p) と定義されたカラムの CSV ファイルへの出力は、小数桁数に合わせてゼロを表示するようになりました。旧バージョンでは、例えば p(5,2)のカラムの値 123.00 は、CSV ファイルに 123 と出力される場合と 123.00 と出力される場合とがありました。表記の違いは主にデータベースから JDBC を介して取得する数値データ (BigDecimal) の特性に依存していました。本バージョンでは、データセット定義に合わせて常に 123.00 と出力します。

(c) COUNT 関数のヘッダ行除外

導出演算の初期化処理で COUNT 関数によって CSV ファイルの行数を取得する場合

合、旧バージョンではヘッダ行を含むファイル内の行数をカウントしていましたが、本バージョンではヘッダ行を除くデータ行の行数だけをカウントするようになりました。

⑥ 導出演算の計算式及び条件式の定義と実行が次のすべてに該当する場合、これまでエラーになった処理が、正常終了するようになります。

- ・ 計算式の除算の分母に外部変数属性を定義している。
- ・ 同じ導出演算の条件式に、除算の分母に定義した外部変数属性 $\neq 0$ の条件がある。
- ・ 実行時のパラメタで除算の分母に定義した外部変数属性に 0 を渡すことがある。
- ・ トランスフォーマの設定ファイルに、`job.zero.div.zero=zero` を指定している。

⑦ 導出演算の AVG 関数の修正によって、AVG 関数の引数に指定した属性の小数桁数が導出項目の属性の小数桁数よりも小さい場合、本バージョンでは正しい精度の数値を出力するので、旧バージョンと処理結果が異なる場合があります。

⑧ 複数ファイル（CSV、固定長テキスト、固定長 COBOL）の入出力を伴う処理で、次のすべての条件に該当する処理は、文字列とバイト列の変換に用いる文字セットの違いによって、処理結果が異なる場合があります。

- ・ 入力データに複数データセットの結合があり、二つ以上のデータセットのデータソースがファイルである。または複数ユーザビューの出力があり、そのうちの二つ以上のデータソースがファイルである。
- ・ 入力時導出演算または出力前導出演算に SUBSTRB または CONCATB 関数がある。
- ・ ファイルのデータソースに二つ以上の異なるエンコーディングの種類が混在している。

以 上