FUJITSU

FUJITSU Software Enterprise Postgres 13



リリース情報

Windows

B1WS-1436-01Z0(00)
2021年4月

まえがき

本書の目的

本書は、FUJITSU Enterprise Postgresのリリース情報を記載しています。

本書の構成

本書の構成と内容は以下のとおりです。

第1章 新機能および改善項目

本バージョンで追加した新機能および改善項目について説明しています。

第2章 互換に関する情報

互換に関する情報について説明しています。

第3章 プログラム修正一覧

本バージョンで吸収した障害修正について説明しています。

輸出管理規制について

本ドキュメントを輸出または第三者へ提供する場合は、お客様が居住する国および米国輸出管理関連法規等の規制をご確認のうえ、必要な手続きをおとりください。

出版年月および版数

2021年　4月　初版

著作権

Copyright 2015-2021 FUJITSU LIMITED

目　次

第1章 新機能および改善項目 1
1.1 13で追加された機能 1
1.1.1 アプリケーション開発 1
1.1.1.1 OpenJDK対応 1
1.1.2 OSS 1
1.1.2.1 PostgreSQLのリベース 1
1.1.2.2 提供するOSSの更新 1

第2章 互換に関する情報 2
2.1 インストール／セットアップに関する非互換 2
2.1.1 クライアントの動作OSの削除 2
2.1.2 クライアントの動作OSの削除 2
2.1.3 サーバの動作OSの削除 3
2.1.4 OSSのセットアップの扱いの変更 3
2.1.5 WebAdminでインスタンスを作成した場合のコアファイルおよびログファイルのパスの変更 3
2.1.6 WebAdminのサービス名の変更 4
2.1.7 64ビット環境でのインストール対象機能の変更 4
2.1.8 Windowsクライアントのインストールディレクトリの権限変更 4
2.1.9 pg_upgrade後のハッシュインデックス再構築 4
2.1.10 WebAdminのインストールの扱いの変更 5
2.2 アプリケーションの移行に関する非互換 5
2.2.1 ecpg/ecobpgの「elif」文の動作の変更 9
2.2.2 ltreeでlquery構文のNOT(!)の処理の変更 9
2.2.3 txid_で始まる関数名の変更 10
2.2.4 psqlの¥dのTOASTテーブルに対する出力の改善 10
2.2.5 psqlの¥eコマンドの編集内容の再表示の改善 11
2.2.6 全文検索の問い合わせの結果の変更 12
2.2.7 DECLARE CURSOR ... WITH HOLDの動作の変更 12
2.2.8 psqlの¥gsetコマンドで特別変数の変更の防止 13
2.2.9 異なるデータ型の日付を比較する際の動作の変更 13
2.2.10 to_date()とto_timestamp()関数の変更 14
2.2.11 pg_stat_statementsのSELECT問い合わせ実行の統計処理の変更 15
2.2.12 CREATE EXTENSIONのFROMオプションをサポートしないように変更 15
2.2.13 pg_available_extension_versions()の結果に情報を追加 16
2.2.14 pg_pltemplateカタログの削除 16
2.2.15 psqlコマンドの「--help」の出力情報の追加 16
2.2.16 psqlコマンドの¥d+の出力情報の追加 17
2.2.17 psqlのデフォルトプロンプトにトランザクション状態の追加 17
2.2.18 CHECK制約の「NO INHERIT」プロパティを伝播するように変更 18
2.2.19 opaque型がサポートされないように変更 18
2.2.20 旧版の外部キー制約の定義がサポートされないように変更 18
2.2.21 旧版の演算子クラスの定義がサポートされないように変更 19
2.2.22 スーパーユーザーのみssl_passphrase_commandの設定値を参照できるように変更 19
2.2.23 pg_triggerカタログにtgparentid列を追加 20
2.2.24 bt_metap()/bt_page_items()関数の変更 20
2.2.25 統計情報ビューpg_stat_activityにleader_pid列を追加 20
2.2.26 デフォルト以外の照合順序を使用するインデックスの制約動作の変更 20
2.2.27 ALTER MATERIALIZED VIEW ... RENAME COLUMN実施時のコマンドタグの変更 21
2.2.28 ALTER FOREIGN TABLE ... RENAME COLUMN実施時のコマンドタグの変更 21
2.2.29 補助プロセスのPIDはpg_stat_sslに記録しないように変更 22
2.2.30 補助プロセスのPIDはpg_stat_gssapiに記録しないように変更 22
2.2.31 ltreeにおいて、中括弧付きアスタリスクの処理を変更 22
2.2.32 json_to_tsvector/jsonb_to_tsvector関数のfilterのチェックの強化 23
2.2.33 「SIMILAR TO ... ESCAPE NULL 」が必ず NULL を返すように変更 23
2.2.34 Oracleデータベース互換機能のto_multibyte()にバックスラッシュを入力した際の出力結果の変更 24

2.2.35 Oracleデータベース互換機能のto_char()に浮動小数点型を入力した際の出力桁数の変更 24
2.2.36 データ秘匿化が可能なデータ型でない型を戻り値として返す演算子または関数を実行した場合に、エラーとなるように変更25
2.2.37 Oracleデータベース互換機能のUTL_FILEの削除 25
2.2.38 psqlコマンドの¥conninfoの出力の変更 25
2.2.39 psqlコマンドの¥helpの出力の変更 26
2.2.40 XML関数のさまざまな不具合の修正 26
2.2.41 to_timestamp()/to_date()関数の変更 27
2.2.42 SERIALIZABLE分離モードでパラレルクエリが使用できるように変更 28
2.2.43 単一の子しか持たないパーティションテーブルおよびUNION ALLの問い合わせの最適化の改善 28
2.2.44 psqlコマンドの¥dと¥zのパーティションテーブルに対する表示の改善 29
2.2.45 システムビューpg_indexesにパーティションインデックスを含むように変更 30
2.2.46 パーティションテーブルにテーブルスペースの指定ができるように変更 30
2.2.47 パーティション境界に任意の式を記述できるように変更 30
2.2.48 パーティションテーブルを参照する外部キーのサポート 31
2.2.49 動的共有メモリを無効化する設定の廃止 31
2.2.50 information_schemaビューで、オブジェクト名の列をvarchar型ではなく、name型として扱うように変更 32
2.2.51 テーブルのname型の列がデフォルトでC照合順序を持つように変更 32
2.2.52 psqlの「¥pset format ..」コマンドでユニークでない略記の禁止 32
2.2.53 pg_restoreの動作の変更 33
2.2.54 pg_verify_checksumsの改名 33
2.2.55 xpath()関数 や XMLTABLE による値の作成で出力を変更 33
2.2.56 SQLスタイルの substring(.. from … for ..)関数の動作の変更 33
2.2.57 random()、setseed()関数がプラットフォーム間で統一 34
2.2.58 新たなアルゴリズムを使用することで、real型、double precision型の値を出力する際の性能を改善 34
2.2.59 幾何関数と演算子のコードリファクタリング 35
2.2.60 timetravel拡張の廃止 35
2.2.61 abstime、reltime、tinterval の各データ型の廃止 35
2.2.62 oid列の特別な振る舞いの廃止 36
2.2.63 Python言語のサポートの変更 36
2.2.64 power関数のPOSIX準拠 37
2.2.65 to_number関数のテンプレート区切り文字が一致しない場合の動作変更 37
2.2.66 to_date、to_number、to_timestamp関数でマルチバイトテンプレート文字列を指定した場合の動作変更 38
2.2.67 to_char、to_number、to_timestamp関数のダブルクォート内でバックスラッシュを指定した場合の動作変更 38
2.2.68 XML関数における相対パスの扱いの変更 39
2.2.69 拡張問い合わせプロトコルにおけるstatement_timeoutの適用範囲の変更 40
2.2.70 ラージオブジェクトの権限チェックタイミングの変更 40
2.2.71 adminpackのpg_file_read、pg_file_length、pg_logfile_rotate関数の削除 40
2.2.72 拡張機能におけるGUC_LIST_QUOTE指定のカスタムパラメータ定義時の動作の変更 41
2.2.73 red-blackツリーサポート関数の名前の変更 41
2.2.74 chkpass拡張の削除 41
2.2.75 ダブルクォートで括ったコマンドオプションの小文字への自動変換の廃止 41
2.2.76 psqlコマンドの¥d+の出力情報の追加 42
2.2.77 pg_basebackupの動作の変更 43
2.2.78 いくつかの関数で変動性区分を変更 43
2.2.79 いくつかの関数をPARALLEL UNSAFEに変更 43
2.2.80 アプリケーションの接続先切り替え機能の指定方法の変更 44
2.2.81 C言語およびCOBOL言語による埋め込みSQLアプリケーションのOSS PostgreSQL互換 44
2.2.82 Vertical Clustered Index(VCI)の選択条件変更 44
2.2.83 システム管理関数の名称変更 45
2.2.84 システム情報関数の返却項目の名称変更 45
2.2.85 FROM句にシーケンスを指定した場合の返却列の変更 46
2.2.86 範囲外の値を入力した場合のto_date関数とto_timestamp関数の動作変更 47
2.2.87 選択リストに集合を返す関数を指定した場合の動作変更 47
2.2.88 同一テーブルが複数回更新される場合の文単位のトリガの動作変更 48
2.2.89 C言語関数におけるVersion 0呼び出し規約の削除 50
2.2.90 Oracleデータベース互換機能のrpad関数にマルチバイト文字を入力した場合の返却結果の変更 50
2.2.91 money型をbigint型で除算する場合の演算方法の変更 50

2.2.92 cubeモジュールの~>演算子の振る舞いの変更....51
2.2.93 tsearch2モジュールの削除....51
2.2.94 psqlコマンドの特別変数に不当な値を指定した場合の動作変更....52
2.2.95 psqlコマンドの¥dおよび¥ddの出力形式の変更....52
2.2.96 PL/Tclの自動ロード機能の削除....53
2.2.97 PL/Pythonにおける複合型配列の扱いの変更....53
2.2.98 DHパラメータのサイズ変更....53
2.2.99 to_char関数におけるマイナス符号の扱いの変更....54
2.2.100 無限大のタイムスタンプを入力した場合のextract関数の戻り値の変更....54
2.2.101 三角関数のPOSIX準拠....55
2.2.102 行集合を返す関数がNULLを返せるように変更....55
2.2.103 PL/pgSQLのRAISEコマンドにおけるCONTEXTメッセージを出力するように変更....56
2.2.104 テキスト検索パーサにおける数字から始まるメールアドレスやホスト名の解析方法の変更....57
2.2.105 関数をORDER BYソートの順序で実行するように変更....58
2.2.106 日付やタイムスタンプの限界値チェックの強化....59
2.2.107 OIDを取得するシステムカタログ情報関数における引数のデータ型の変更....59
2.2.108 不当な値を入力した場合のシステムカタログ情報関数の動作変更....60
2.2.109 レプリケーションSQL関数pg_replication_origin_xact_resetの引数の削除....61
2.2.110 psqlコマンドの-cオプションを指定した場合に起動用ファイルを読み込むように変更....62
2.2.111 画面表示モードが自動設定の場合のpsqlコマンドの表示方法の変更....62
2.2.112 psqlコマンドの¥df+の出力情報の追加....62
2.2.113 環境変数PAGERに空文字列または全て空白文字の文字列を指定した場合のpsqlコマンドの表示方法の変更....62
2.2.114 pgbenchコマンドの独自スクリプトの書式変更....63
2.2.115 pgbenchコマンドのスクリプトファイルメタコマンド¥setrandomの削除....63
2.2.116 hstoreのhstore_to_jsonb_loose関数の数値判別の厳格化....64
2.2.117 pageinspectのheap_page_items関数の返却情報の追加....65
2.2.118 Oracleデータベース互換機能利用時の変更....65
2.3 運用の移行に関する非互換....65
2.3.1 pg_settingsビューのcategory列の値の変更....69
2.3.2 統計情報ビューpgx_stat_lwlockの変更....70
2.3.3 pgx_rcvallの動作の変更....71
2.3.4 postgresql.confファイルのmax_files_per_processパラメータの指定値の範囲を変更....71
2.3.5 REINDEX CONCURRENTLYの前後でのプランナ統計情報の変更....71
2.3.6 ALTER TABLE ONLY ... ENABLE/DISABLE TRIGGERが子テーブルを再帰的に処理しないように変更....72
2.3.7 認証設定ファイルにUnixドメインソケットの認証設定の追加....73
2.3.8 統計情報ビューpg_stat_statementsの変更....74
2.3.9 統計情報ビューpg_stat_activityの変更....75
2.3.10 createuserコマンドのオプション「-a/--adduser」と「-A/--no-adduser」の削除....78
2.3.11 pg_waldumpの解析結果に「PREPARE TRANSACTION」に関連する情報を追加....78
2.3.12 EXPLAIN文にプラン作成時のバッファ使用量を追加....78
2.3.13 指定するリカバリ目標に到達しない場合の動作を変更....79
2.3.14 最小TLSバージョンのデフォルト値の変更....80
2.3.15 EXPLAIN文に子テーブルに対応する列別名の追加....80
2.3.16 ログ出力先がcsvlogの場合のフォーマットを変更....81
2.3.17 postgresql.confファイルのwal_keep_segmentsパラメータ名をwal_keep_sizeに変更....81
2.3.18 postgresql.confファイルのeffective_io_concurrencyパラメータの設定方法の変更....82
2.3.19 Mirroring Controllerによるデータベースプロセスのダウン検知時の監視リトライの廃止....82
2.3.20 Mirroring Controllerの昇格後コマンドの名称とパラメータ名の変更....82
2.3.21 Mirroring Controllerのユーザーコマンドの入力値変更....82
2.3.22 postgresql.confファイルのdefault_with_oidsパラメータの削除....83
2.3.23 postgresql.confファイルのjitパラメータのデフォルト値の変更....83
2.3.24 pg_rewindの「--dry-run」オプションの動作の変更....83
2.3.25 pg_upgradeコマンドの動作の変更....83
2.3.26 外部キーのデフォルトの制約名の変更....84
2.3.27 max_connectionsの計算方法の変更....84
2.3.28 設定パラメータのデータ型の変更....85
2.3.29 autovacuum_vacuum_cost_delayのデフォルト値の変更....85

2.3.30 共通テーブル式(CTE)をインライン化するように変更 85
2.3.31 サーバの起動時にサーバのバージョン番号を出力するように変更 85
2.3.32 権限のないユーザーにpg_stat_sslビューの行の表示を制限するように変更 86
2.3.33 pg_stat_sslシステムビューのclientdn列の改名 86
2.3.34 pg_stat_sslシステムビューにクライアント認証に関する情報を追加 86
2.3.35 グローバルオブジェクトをシステムビューpg_stat_databaseで追跡できるように変更 86
2.3.36 システムビューpg_stat_databaseにchecksum_failures列とchecksum_last_failure列を追加 87
2.3.37 pg_attrdefシステムテーブルからadsrc列を削除 87
2.3.38 pg_constraintシステムテーブルからconsrc列を削除 87
2.3.39 pg_statistic_extシステムテーブルの分割 87
2.3.40 DROP IF EXISTS FUNCTION/PROCEDURE/AGGREGATE/ROUTINEのメッセージの変更 87
2.3.41 Btreeインデックスの最大エントリ長の変更 88
2.3.42 postgresql.confファイルのrecovery_target_timelineパラメータのデフォルト値の変更 88
2.3.43 recovery_target_*パラメータの複数指定の不可 88
2.3.44 recovery.confの廃止 89
2.3.45 pg_procカタログのprotransform列をprosupport列に変更 89
2.3.46 コマンドの出力メッセージのフォーマットの変更 89
2.3.47 pg_dumpおよびpg_restoreコマンドのデフォルト動作の変更 90
2.3.48 pg_dumpallコマンドのデフォルト動作の変更 90
2.3.49 pg_dumpallの--globals-onlyオプションの動作の変更 91
2.3.50 非一時テーブルに対する一時テーブルのパーティション設定不可 91
2.3.51 関数と列参照のデフォルト動作の変更 91
2.3.52 テーブルとドメイン制約の命名規約を変更 92
2.3.53 pg_classカタログからrelhaspkey列を削除 93
2.3.54 情報スキーマのtablesビューのtable_type列のテーブル種類名の変更 93
2.3.55 pg_procカタログのproisagg列とproiswindow列をprokind列に統合 93
2.3.56 非スーパーユーザーによる共有カタログのインデックス再作成の防止 94
2.3.57 psコマンドで出力されるプロセス名をpg_stat_activityに同期 94
2.3.58 ロジカルレプリケーション時のTRUNCATE文の動作の変更 94
2.3.59 情報スキーマtriggersのaction_order列、action_reference_old_table列、action_reference_new_table列に格納する値の変更 95
2.3.60 情報スキーマtable_constraintsへのenforced列の追加 95
2.3.61 CREATE TABLE文 のLIKE句オプション INCLUDING ALL 指定の動作変更 96
2.3.62 postgresql.confファイルのreplacement_sort_tuplesパラメータの削除 97
2.3.63 CREATE FUCNTION 文のWITH句の廃止 97
2.3.64 EXPLAIN ANALYZE 文でのパラレルワーカーのソート処理情報の追加 97
2.3.65 プランナが抽出条件「カラム名 = カラム名」を「IS NOT NULL」へ置換する処理に変更 98
2.3.66 pg_rewindコマンドがOSのスーパーユーザー(root)で実行不可に変更 99
2.3.67 pg_stat_statements_resetの実行がデータベースのスーパーユーザー権限を持つユーザー以外で禁止に変更 99
2.3.68 pg_stat_statementsのqueryid列を64ビットに拡張 99
2.3.69 ビューに対する SELECT FOR UPDATE の動作変更 100
2.3.70 VACUUM のオプションに列リストを指定する場合の動作変更 100
2.3.71 システムビューpg_stat_wal_receiverへのsender_host列、sender_port列の追加 101
2.3.72 pg_dumpが出力するCREATE TRIGGER ... REFERENCINGコマンドの遷移テーブル名をダブルクォートで括るように変更 101
2.3.73 パーティションテーブルに対するCOPYコマンドの動作変更 102
2.3.74 論理レプリケーションスロットが存在する場合のサーバ起動可否判定の変更 102
2.3.75 1サーバ内での最大接続数の変更 103
2.3.76 ALTER TABLESPACE文による暗号化設定の変更 103
2.3.77 データ格納先ディレクトリのサブディレクトリの名称変更 103
2.3.78 コマンドおよびオプションの名称変更 103
2.3.79 pg_ctlコマンドの操作完了待ちのデフォルト動作の変更 104
2.3.80 pg_ctlコマンドのstartモードとrestartモードにおけるタイムアウト時の終了ステータスの変更 104
2.3.81 pg_basebackupコマンドのデフォルト動作の変更と-xオプションの削除 105
2.3.82 createlangコマンドとdroplangコマンドの削除 105
2.3.83 クライアントコマンドのsearch_path設定の変更 105
2.3.84 postgresql.confファイルのパラメータのデフォルト値の変更 106

2.3.85 postgresql.confファイルのmin_parallel_relation_sizeパラメータの削除とmin_parallel_table_scan_sizeパラメータおよびmin_parallel_index_scan_sizeパラメータの追加 107
2.3.86 postgresql.confファイルのsql_inheritanceパラメータの削除 107
2.3.87 postgresql.confファイルのpassword_encryptionパラメータの指定値の変更 107
2.3.88 postgresql.confファイルのshared_preload_librariesパラメータと関連するパラメータ群の大文字小文字区別の変更 108
2.3.89 RADIUS認証の設定オプションの名称変更 108
2.3.90 統計情報ビューpg_stat_activityの変更 108
2.3.91 統計情報ビューpg_stat_replicationの変更 110
2.3.92 システムビューpg_user_mappingsのumoptions列の可視性の厳格化 110
2.3.93 pg_stat_statementsビューに格納するSQL文の定数の置換方法の変更 111
2.3.94 CREATE ROLE文と同種のコマンド群のUNENCRYPTEDオプションの削除 111
2.3.95 CREATE ROLE文と同種のコマンド群における空パスワード指定時の動作変更 112
2.3.96 親テーブルにPRIMARY KEY制約を追加する場合の動作変更 113
2.3.97 レプリケーション接続認証のデフォルト設定の変更 114
2.3.98 論理レプリケーションのクライアント認証の設定方法の変更 114
2.3.99 データベース多重化機能によるクラスタシステムのデフォルト構成の変更 115
2.3.100 mc_ctlコマンドのオプション省略時のデフォルト動作の変更 115
2.3.101 Mirroring Controllerがインスタンスに接続する際の接続設定の変更 115
2.3.102 Mirroring Controllerのサーバの状態表示の変更 116
2.3.103 データベース多重化運用中のsynchronous_standby_namesパラメータ変更時の動作の変更 116
2.3.104 CREATE ROLE文と同種のコマンド群のCREATEUSERオプションとNOCREATEUSERオプションの削除 117
2.3.105 pg_で始まるロール名を使用不可にする変更 117
2.3.106 pg_controldataおよび関連箇所におけるNextXIDの表示形式の変更 118
2.3.107 統計情報ビューpg_stat_activityのwaiting列の削除とwait_event_type列とwait_event列の追加 118
2.3.108 pg_amカタログの列の変更 118
2.3.109 システムビューpg_replication_slotsへのconfirmed_flush_lsn列の追加 118
2.3.110 システムビューpg_settingsのunit列に格納する値の変更 119
2.3.111 情報スキーマのroutinesビューのresult_cast_character_set_name列名の変更 120
2.3.112 update_process_titleパラメータのデフォルト値の変更 120
2.3.113 未サポートデータ型に対する秘匿化ポリシー定義の変更 120
2.4 JDBCドライバに関する非互換 120
2.4.1 targetServerTypeの値の変更 122
2.4.2 PgDatabaseMetaDataクラスのgetTypeInfoメソッドの返却値の変更 122
2.4.3 Java LocalTimeを変換する場合の動作の変更 122
2.4.4 PgDatabaseMetaDataクラスのgetIndexInfoメソッドの返却値の変更 123
2.4.5 PgDatabaseMetaDataクラスのgetProceduresメソッドの動作の変更 123
2.4.6 DatabaseMetaDataクラスのgetColumnsメソッドの返却値の変更 123
2.4.7 PgConnectionクラスのgetTypeMapメソッドでエントリが存在しない場合の返却値の変更 124
2.4.8 text配列の引用符で囲まれていない要素の返却値の変更 124
2.4.9 ConnectionクラスのcreateArrayメソッドが作成する配列の動作の変更 124
2.4.10 データベース列の値がNULL、かつ、JREアサーションが有効な場合の動作の変更 125
2.4.11 PgTokenizerクラスの返却値の変更 125
2.4.12 gssEncModeパラメータのデフォルト値の変更 125
2.4.13 1582年10月の時間を変換時の返却値の変更 126
2.4.14 getNotificationsメソッドの返却値の変更 126
2.4.15 PgResultSetクラスのrelativeメソッドの返却値の変更 126
2.4.16 PGXAConnectionクラスのcommitPreparedメソッドのException処理の変更 127
2.4.17 BaseDataSourceクラスのsetUrlメソッドでエラーのURLを指定した場合の動作の変更 127
2.4.18 PgConnectionクラスのabortメソッドで引数にNULLを指定した場合の動作の変更 127
2.4.19 Driverクラスのconnectメソッドで1つ目の引数にNULLを指定した場合のException処理の変更 128
2.4.20 1970年より前かつ時刻を持つタイムスタンプの解析結果の変更 128
2.4.21 紀元前の時間の解析結果の変更 128
2.4.22 PGCopyInputStreamクラスのreadメソッドおよびReaderInputStreamクラスのreadメソッドの返却値範囲の変更 129
2.4.23 PgDatabaseMetaDataクラスのgetFunctionColumnsメソッドの返却値の変更 129
2.4.24 検索結果に浮動小数点数のNaNがある場合の返却値の変更 130
2.4.25 リソースバンドル取得時のロケールを変更 130
2.4.26 PGObject型のgetStringメソッドの返却値の変更 130

2.4.27 sslオプションにtrueを指定した場合のsslmodeのデフォルト値の変更 130
2.4.28 接続パラメータの削除 131
2.4.29 ログの取得方法の変更 131
2.4.30 カレンダを使用した場合のgetTimeメソッドの返却値の変更 131
2.4.31 24:00:00以上の時刻を入力した場合のgetTimeメソッドの返却値の変更 132
2.4.32 DriverManagerクラスのgetConnectionメソッドのURLへのデータベース名および接続引数の値の指定方法の変更 132
2.4.33 connectTimeoutパラメータのデフォルト値の変更 133
2.5 ODBCドライバに関する非互換 133
2.5.1 SQLGetDescField関数の変更 134
2.5.2 SQLDescribeCol関数で取得するカラムの属性の変更 134
2.5.3 PGAPI_GetInfo関数の引数rgbInfoValueの戻り値の変更 134
2.5.4 クライアントのロケール選定方法の変更 135
2.5.5 SQLTables関数およびSQLTablePrivileges関数でパーティションテーブルを表示するよう変更 136
2.5.6 存在しないテーブルを指定した場合のSQLSpecialColumns関数の動作変更 136
2.5.7 ODBCドライバのデフォルト値の変更 136
2.5.8 クライアント符号化方式のデフォルト変更 137
2.5.9 UTF-8の1文字あたりの最大バイト数の変更 137
2.5.10 通信リンク障害発生時のSQLSTATEの変更 137
2.6 .NET Data Providerに関する非互換 137
2.6.1 TargetServerTypeの値の変更 140
2.6.2 空コマンドの実行時の例外発生に関する変更 140
2.6.3 tsquery型の値の表示の変更 140
2.6.4 Npgsql.NetTopologySuiteに必要なパッケージの変更 141
2.6.5 UseSslStreamプロパティの廃止 141
2.6.6 SnakeCaseNameTranslatorの動作の変更 141
2.6.7 .NET Frameworkアプリケーションの開発および動作に必要な環境の変更 142
2.6.8 組み込みのPostGISデータ型をNpgsql.LegacyPostGISプラグインに変更 142
2.6.9 NpgsqlParameterクラスのEnumTypeプロパティおよびSpecificTypeプロパティの削除 142
2.6.10 NpgsqlDataReaderクラスのGetDataTypeNameメソッドの返却値の変更 142
2.6.11 パラメータの型のチェックの強化 143
2.6.12 パラメータ名のチェックの強化 143
2.6.13 パラメータ名を省略した場合の動作の変更 143
2.6.14 関数を呼び出す場合の変更 143
2.6.15 NpgsqlInet構造体の削除 144
2.6.16 DateTimeマッピングの変更 144
2.6.17 DateTimeOffSetのDateの変更 144
2.6.18 小数点以下が28桁を超えるdecimal/numeric(浮動小数点でない)の変更 144
2.6.19 NpgsqlBinaryImporterの動作の変更 145
2.6.20 NpgsqlBinaryImporter.Cancelの削除 145
2.6.21 Enlistのデフォルト値の変更 145
2.6.22 マップされていないenum型を文字列として書き込む場合の設定方法の変更 146
2.6.23 TLS/SSLの内部実装の変更 146
2.6.24 独立性レベルの変更 146
2.6.25 IsAutoIncrementの返却値の変更 146
2.6.26 .NET Data ProviderのVisual Studio連携でのサーバーエクスプローラーの表示変更 147
2.6.27 エンティティフレームワークの名前の変更 147
2.6.28 SSLキーワードの削除 147
2.6.29 SslmodeキーワードのAllowモードの削除 147
2.6.30 PreloadReaderキーワードの削除 148
2.6.31 UseExtendedTypesキーワードの削除 148
2.6.32 AlwaysPrepareキーワードの削除 148
2.6.33 Protocolキーワードの削除 148
2.6.34 Compatibleキーワードの削除 148
2.6.35 SyncNotificationキーワードの削除 148
2.6.36 ConnectionLifeTimeキーワードの名前とデフォルト値の変更 149
2.6.37 MinPoolSizeキーワードとMaxPoolSizeキーワードのデフォルト値の変更 149
2.6.38 CommandTimeoutのデフォルト値の変更 149

2.6.39 NpgsqlTypes.NpgsqlMacAddress構造体の削除 149
2.6.40 NpgsqlTypes.BitString構造体の削除 150
2.6.41 NpgsqlTypes.NpgsqlTime構造体の削除 150
2.6.42 NpgsqlTypes.NpgsqlTimeZone構造体の削除 150
2.6.43 NpgsqlTypes.NpgsqlTimeTZ構造体の削除 150
2.6.44 NpgsqlDataReaderクラスのHasOrdinalメソッドの削除 150
2.6.45 NpgsqlParameterCollectionクラスのAdd(string, object)メソッドの削除 150
2.6.46 NpgsqlConnectionクラスのSupportsHexByteFormatプロパティの削除 151
2.6.47 NpgsqlExceptionクラスのBaseMessageプロパティの削除 151
2.6.48 NpgsqlConnectionクラスのSupports_E_StringPrefixプロパティの名前変更 151
2.6.49 NpgsqlInet構造体のフィールドの変更 151
2.6.50 通信データの暗号化を使用する場合にサーバ証明書を検証するように変更 151
2.6.51 NpgsqlParameterクラスのValueプロパティへのNULL値の設定方法の変更 152
2.6.52 パラメータへのIEnumerable<T>の値の設定不可 152
2.6.53 整数型のパラメータへの列挙体の設定不可 152
2.6.54 NpgsqlTypes.NpgsqlTimeStamp構造体の無限大の扱いの変更 152
2.6.55 NpgsqlDataReaderクラスのGetFieldTypeメソッドおよびGetProviderSpecificFieldTypeメソッドの返却値の変更 153
2.6.56 NpgsqlDataReaderクラスのGetDataTypeNameメソッドの返却値の変更 153
2.6.57 カーソルを返却するユーザ定義関数の使用不可 153
2.6.58 データベース接続時の送信情報の変更 153
2.6.59 NpgsqlConnectionクラスのCloneメソッドで複製されるNpgsqlConnectionオブジェクトの接続状態の変更 154
2.6.60 CommandType.TableDirectを指定して検索する場合の表の指定方法の変更 154
2.6.61 ユーザ定義関数を呼び出す場合の関数名の指定方法の変更 154
2.6.62 UTF-8として無効な文字を入力した場合にエラーにするように変更 155
2.6.63 STDINまたはSTDOUTを指定した場合のCOPY文の実行方法の変更 155
2.6.64 NpgsqlParameterクラスのSizeプロパティに指定されたサイズでパラメータの値を設定するように変更 155
2.6.65 トランザクションの隔離レベルにChaosを指定した場合の動作変更 155
2.6.66 NpgsqlConnectionクラスのStateプロパティとFullStateプロパティの設定値の追加 156
2.6.67 NpgsqlConnectionクラスのDataSourceプロパティの形式変更 156
2.6.68 コネクション接続後に接続文字列からパスワードの情報を除去するように変更 156
2.6.69 サーバでエラーが発生した場合にPostgresExceptionを返すように変更 157
2.6.70 SQLSTATEの取得方法の変更 157
2.6.71 NpgsqlDataReaderオブジェクトのオープン中に準備文を破棄できないように変更 157
2.6.72 環境変数PGCLIENTENCODINGを指定した場合にクライアント側符号化方式を設定するように変更 157
2.6.73 レコード型を返す関数の返却値のデータ型の変更 158
2.6.74 ユーザ定義関数を呼び出す場合の名前付きパラメータのサポート 158
2.6.75 NpgsqlConnectionクラスのコンストラクタの引数の変更 158
2.7 C言語用ライブラリ(libpq)に関する非互換 158
2.7.1 接続状態関数PQhostの動作の変更 159
2.7.2 ホスト名を省略した場合における接続状態関数PQhostの返却値の変更 159
2.8 oracle_fdwに関する非互換 160
2.8.1 外部テーブルのパーティションテーブルに対してINSERT ... ON CONFLICTを実行した時のエラーメッセージを変更 160
2.8.2 外部テーブルのパーティションテーブルに対してINSERT ... RETURNINGを実行した時に挿入された通りの行を返却するように変更 161
2.8.3 外部テーブル使用時にOracleからPostgreSQLへのデータ型の変換チェック処理を追加 162
2.8.4 外部テーブルの外部結合をOracleにプッシュダウンするように変更 162
2.9 pgAdmin4に関する非互換 164
2.9.1 デスクトップアプリケーションからシステムトレイアプリケーションに変更 164
2.9.2 データベースとの接続が切断された時の動作の変更 164
2.9.3 設定の項目SQL エディタの名称をクエリツールに変更 164
2.9.4 クエリツールのクエリの履歴パネルでの表示内容の変更 165
2.9.5 SQLタブのスクリプト表示仕様の変更 165
2.9.6 関数の実行でエラーが発生した場合の出力メッセージの変更 166
2.10 pgauditに関する非互換 166
2.10.1 トリガ関数実行時に、監査ログにNEW、OLDの値を追加で出力するように変更 166
2.10.2 パーティションテーブルまたはそのインデックスのオブジェクトタイプの変更 167
2.10.3 監査ログのコマンドタグの出力値を「UNKNOWN」から「???」に変更 167

第3章 プログラム修正一覧.....168
索引.....169

第1章 新機能および改善項目

本バージョンで追加したFUJITSU Enterprise Postgresの新機能の対応製品および改善項目について説明します。

表1.1 新機能と改善項目

バージョンレベル	分類	機能名	AE	SE
13	アプリケーション開発	OpenJDK対応	○	○
	OSS	PostgreSQLのリベース	○	○
		提供するOSSの更新	○	○

○: 機能あり

1.1 13で追加された機能

13で追加された新機能および改善項目について説明します。

1.1.1 アプリケーション開発

アプリケーション開発に関する新機能および改善項目を説明します。

- OpenJDK対応

1.1.1.1 OpenJDK対応

FUJITSU Enterprise Postgresクライアント機能の動作環境において、これまでのOracleのJDKまたはJREでの動作に加えて、OpenJDKでの動作も保証します。

参照

詳細は、“導入ガイド(クライアント編)”の“関連ソフトウェア”を参照してください。

1.1.2 OSS

OSSに関する新機能を説明します。

- PostgreSQLのリベース
- 提供するOSSの更新

1.1.2.1 PostgreSQLのリベース

FUJITSU Enterprise Postgresがベースとするpostgresqlのバージョンが13.1にリベースされました。

参照

詳細は、“導入ガイド(サーバ編)”の“FUJITSU Enterprise Postgresがベースとするpostgresqlのバージョン”を参照してください。

1.1.2.2 提供するOSSの更新

FUJITSU Enterprise Postgresが提供するOSSが更新されました。

参照

詳細は、“解説書”の“FUJITSU Enterprise Postgresがサポートするoss”を参照してください。

第2章 互換に関する情報

旧バージョンからFUJITSU Enterprise Postgres 13に移行する場合の非互換項目と対処について説明します。移行の際には互換性を確認し、必要に応じて対処してください。

2.1 インストール／セットアップに関する非互換

項目	移行前のバージョン				
	9.5	9.6	10	11	12
クライアントの動作OSの削除	―	―	◎	◎	◎
クライアントの動作OSの削除	◎	◎	◎	◎	―
サーバの動作OSの削除	◎	◎	◎	◎	―
OSSのセットアップの扱いの変更	◎	◎	◎	◎	―
WebAdminでインスタンスを作成した場合のコアファイルおよびログファイルのパスの変更	◎	◎	◎	◎	―
WebAdminのサービス名の変更	◎	◎	◎	◎	―
64ビット環境でのインストール対象機能の変更	◎	◎	―	―	―
Windowsクライアントのインストールディレクトリの権限変更	◎	◎	―	―	―
pg_upgrade後のハッシュインデックス再構築	○	○	―	―	―
WebAdminのインストールの扱いの変更	◎	―	―	―	―

◎:FUJITSU Enterprise Postgresによる非互換あり

○:OSSによる非互換あり

―:非互換なし

2.1.1 クライアントの動作OSの削除

非互換

FUJITSU Enterprise Postgres 13以降では、以下のOSが動作OSから削除されました。

- SLES 12 SP4 以前

対処方法

ありません。

2.1.2 クライアントの動作OSの削除

非互換

FUJITSU Enterprise Postgres 12以降では、以下のOSが動作OSから削除されました。

- Windows(R) 7
- Windows Server(R) 2008
- RHEL6

また、RHEL6が動作OSから削除されたことにより、32ビット版のLinuxクライアントがインストールできなくなりました。

対処方法

ありません。

2.1.3 サーバの動作OSの削除

非互換

FUJITSU Enterprise Postgres 12以降では、以下のOSが動作OSから削除されました。

- Windows Server(R) 2008

対処方法

ありません。

2.1.4 OSSのセットアップの扱いの変更

非互換

FUJITSU Enterprise Postgres 12以降では、OSSの拡張モジュールが実行用ディレクトリに配置されません。OSSをセットアップする際にOSSの拡張モジュールを実行用ディレクトリに配置してください。

詳細は、“導入ガイド(サーバ編)”の“OSSのセットアップおよびアンセットアップ”を参照してください。

対処方法

ありません。

2.1.5 WebAdminでインスタンスを作成した場合のコアファイルおよびログファイルのパスの変更

非互換

FUJITSU Enterprise Postgres 12以降では、WebAdminでインスタンスを作成した場合のコアファイルおよびログファイルのパスを変更します。

FUJITSU Enterprise Postgres 11以前の場合

ログファイルのパス: <ユーザープロファイルのフォルダ>¥<ローカル設定のフォルダ>¥Fujitsu¥fsep_バージョン¥インスタンス名ポート番号¥log

コアファイルのパス: <ユーザープロファイルのフォルダ>¥<ローカル設定のフォルダ>¥Fujitsu¥fsep_バージョン¥インスタンス名ポート番号¥core

バージョン:製品のバージョン_エディション_アーキテクチャ

［例］

ログファイルのパス: C:¥Users¥naomi¥AppData¥Local¥Fujitsu¥fsep_110_AE_64¥myinst27599¥log

コアファイルのパス: C:¥Users¥naomi¥AppData¥Local¥Fujitsu¥fsep_110_AE_64¥myinst27599¥core

FUJITSU Enterprise Postgres 12の場合

ログファイルのパス: <ユーザープロファイルのフォルダ>¥<ローカル設定のフォルダ>¥Fujitsu¥fsep_バージョン¥インスタンス名ポート番号¥log

コアファイルのパス: <ユーザープロファイルのフォルダ>¥<ローカル設定のフォルダ>¥Fujitsu¥fsep_バージョン¥インスタンス名ポート番号¥core

バージョン:製品のバージョン_WA_アーキテクチャ

［例］

ログファイルのパス: C:¥Users¥naomi¥AppData¥Local¥Fujitsu¥fsep_120_WA_64¥myinst27599¥log

コアファイルのパス: C:¥Users¥naomi¥AppData¥Local¥Fujitsu¥fsep_120_WA_64¥myinst27599¥core

対処方法

ありません。

2.1.6 WebAdminのサービス名の変更

非互換

FUJITSU Enterprise Postgres 12以降では、WebAdminをセットアップした際に登録するサービス名を変更します。

FUJITSU Enterprise Postgres 11以前の場合

fsep_xSPz_エディション_64_WebAdmin_Port1

fsep_xSPz_エディション_64_WebAdmin_Port2

FUJITSU Enterprise Postgres 12以降の場合

fsep_xSPz_WA_64_WebAdmin_Port1

fsep_xSPz_WA_64_WebAdmin_Port2

対処方法

ありません。

2.1.7 64ビット環境でのインストール対象機能の変更

非互換

FUJITSU Enterprise Postgres 10以降では、RHEL7で32ビット版のLinuxクライアントがインストールできなくなりました。

備考) FUJITSU Enterprise Postgres 11以前ではRHEL6で32ビット版のLinuxクライアントが利用可能です。

対処方法

ありません。

2.1.8 Windowsクライアントのインストールディレクトリの権限変更

非互換

インストールディレクトリをデフォルトから変更した場合、インストールされたファイルやディレクトリのアクセス権限は、システム管理者以外読み取りと実行のアクセス権になります。

本非互換は、Windowsクライアントが動作可能なクライアントOSが対象となります。動作環境については、“導入ガイド(クライアント編)”の“前提基本ソフトウェア”を参照してください。

対処方法

セキュリティ強化を目的とした変更のため、通常は対処不要です。

ただし、従来のアクセス権限が必要となる場合は、インストール後に、“インストールフォルダ¥setup¥revert_cacls.bat”を管理者権限で実行することで、本変更を無効にできます。

2.1.9 pg_upgrade後のハッシュインデックス再構築

非互換

FUJITSU Enterprise Postgres 10以降では、pg_upgradeコマンドでインスタンスをアップグレードした後、ハッシュインデックスを再構築する必要があります。再構築しない場合、そのハッシュインデックスは一切使用されません。

対処方法

ありません。

pg_upgradeコマンドを実行するとハッシュインデックスを再構築するスクリプトが生成されるため、スクリプトを実行してハッシュインデックスを再構築してください。

2.1.10 WebAdminのインストールの扱いの変更

非互換

FUJITSU Enterprise Postgres 9.6以降では、サーバのインストール時にWebAdminは自動的にインストールされません。そのため、WebAdminを利用したい場合は、サーバのインストールとは別にWebAdminをインストールしてください。

対処方法

ありません。

2.2 アプリケーションの移行に関する非互換

項目	移行前のバージョン				
	9.5	9.6	10	11	12
ecpg/ecobpgの「elif」指示子の動作の変更	◎	◎	◎	◎	◎
ltreeでlquery構文のNOT(!)の処理の変更	○	○	○	○	○
txid_で始まる関数名の変更	○	○	○	○	○
psqlの¥dのTOASTテーブルに対する出力の改善	○	○	○	○	○
psqlの¥eコマンドの編集内容の再表示の改善	○	○	○	○	○
全文検索の問い合わせの結果の変更	○	○	○	○	○
DECLARE CURSOR ... WITH HOLDの動作の変更	○	○	○	○	○
psqlの¥gsetコマンドで特別変数の変更の防止	○	○	○	○	○
異なるデータ型の日付を比較時の動作の変更	○	○	○	○	○
to_date()とto_timestamp()関数の変更	○	○	○	○	○
pg_stat_statementsのSELECT問い合わせ実行の統計処理の変更	○	○	○	○	○
CREATE EXTENSIONのFROMオプションをサポートしないように変更	○	○	○	○	○
pg_available_extension_versions()の結果に情報の追加	○	○	○	○	○
pg_pltemplateカタログの削除	○	○	○	○	○
psqlコマンドの「--help」の出力情報の追加	○	○	○	○	○
psqlコマンドの¥d+の出力情報の追加	○	○	○	○	○
psqlのデフォルトプロンプトにトランザクション状態の追加	○	○	○	○	○
CHECK制約の「NO INHERIT」プロパティを伝播するように変更	○	○	○	○	○
opaque型がサポートしないように変更	○	○	○	○	○
旧版の外部キー制約の定義をサポートしないように変更	○	○	○	○	○
旧版の演算子クラスの定義をサポートしないように変更	○	○	○	○	○
スーパーユーザのみssl_passphrase_commandの設定値を参照できるように変更	○	○	○	○	○
pg_triggerカタログにtgparentid列の追加	○	○	○	○	○
bt_metap()/bt_page_items()関数の変更	○	○	○	○	○
統計情報ビューpg_stat_activityにleader_pid列の追加	○	○	○	○	○
デフォルト以外の照合順序を使用するインデックスの制約動作の変更	○	○	○	○	○

項目	移行前のバージョン				
	9.5	9.6	10	11	12
ALTER MATERIALIZED VIEW ... RENAME COLUMN実施時のコマンドタグの変更	○	○	○	○	○
ALTER FOREIGN TABLE ... RENAME COLUMN実施時のコマンドタグの変更	○	○	○	○	○
補助プロセスのPIDはpg_stat_sslに記録しないように変更	―	―	○	○	○
補助プロセスのPIDはpg_stat_gssapiに記録しないように変更	―	―	―	―	○
ltreeにおいて、中括弧付きアスタリスクの処理の変更	○	○	○	○	○
json_to_tsvector/jsonb_to_tsvector関数のfilterのチェックルールの変更	―	―	―	○	○
SIMILAR TO ... ESCAPE NULL が必ず NULL を返すように変更	○	○	○	○	○
Oracleデータベース互換機能のto_multibyte()にバックスラッシュを入力した際の出力結果の変更	○	○	○	○	○
Oracleデータベース互換機能のto_char()に浮動小数点型を入力した際の出力桁数の変更	○	○	○	○	○
データ秘匿化が可能なデータ型でない型を戻り値として返す演算子または関数を実行した場合に、エラーとなるように変更	◎	◎	◎	◎	―
Oracleデータベース互換機能のUTL_FILEの削除	◎	◎	◎	◎	―
psqlコマンドの¥conninfoの出力の変更	○	○	○	○	―
psqlコマンドの¥helpの出力の変更	○	○	○	○	―
XML関数のさまざまな不具合の修正	―	―	○	○	―
to_timestamp()/to_date()関数の変更	○	○	○	○	―
SERIALIZABLE分離モードでパラレルクエリが使用できるように変更	○	○	○	○	―
単一の子しか持たないパーティションテーブルおよびUNION ALLの問い合わせの最適化の改善	―	―	○	○	―
psqlの¥dと¥zのパーティションテーブルに対する表示の改善	―	―	○	○	―
システムビューpg_indexesにパーティションインデックスを含むように変更	―	―	―	○	―
パーティションテーブルにテーブルスペースの指定ができるように変更	―	―	○	○	―
パーティション境界に任意の式を記述できるように変更	―	―	○	○	―
パーティションテーブルを参照する外部キーのサポート	―	―	○	○	―
動的共有メモリを無効化する設定の廃止	○	○	○	○	―
information_schemaビューで、オブジェクト名の列をvarchar型ではなく、name型として扱うように変更	○	○	○	○	―
テーブルのname型の列がデフォルトでC照合順序を持つように変更	○	○	○	○	―
psqlの「¥pset format ..」コマンドでユニークでない略記の禁止	○	○	○	○	―
pg_restoreの動作の変更	○	○	○	○	―
pg_verify_checksumsの改名	―	―	―	○	―
xpath()関数 や XMLTABLE による値の作成で出力を変更	○	○	○	○	―

項目	移行前のバージョン				
	9.5	9.6	10	11	12
SQLスタイルの substring(.. from … for ..)関数の動作の変更	○	○	○	○	－
新たなアルゴリズムを使用することで、real型、double precision型の値を出力する際の性能を改善	○	○	○	○	－
幾何関数と演算子のコードリファクタリング	○	○	○	○	－
timetravel拡張の廃止	○	○	○	○	－
abstime、reltime、tintervalの各データ型の廃止	○	○	○	○	－
oid列の特別な振る舞いの廃止	○	○	○	○	－
Python言語のサポートの変更	○	○	○	○	－
power関数のPOSIX準拠	○	○	○	－	－
to_number関数のテンプレート区切り文字が一致しない場合の動作変更	○	○	○	－	－
to_date、to_number、to_timestamp関数でマルチバイトテンプレート文字列を指定した場合の動作変更	○	○	○	－	－
to_char、to_number、to_timestamp関数のダブルクォート内でバックスラッシュを指定した場合の動作変更	○	○	○	－	－
XML関数における相対パスの扱いの変更	○	○	○	－	－
拡張問い合わせプロトコルにおけるstatement_timeoutの適用範囲の変更	○	○	○	－	－
ラージオブジェクトの権限チェックタイミングの変更	○	○	○	－	－
adminpackのpg_file_read、pg_file_length、pg_logfile_rotate関数の削除	○	○	○	－	－
拡張機能におけるGUC_LIST_QUOTE指定のカスタムパラメータ定義時の動作の変更	○	○	○	－	－
red-blackツリーサポート関数の名前の変更	○	○	○	－	－
chkpass拡張の削除	○	○	○	－	－
ダブルクォートで括ったコマンドオプションの小文字への自動変換の廃止	○	○	○	－	－
psqlコマンドの¥d+の出力情報の追加	○	○	○	－	－
pg_basebackupの動作の変更	○	○	○	－	－
いくつかの関数で変動性区分を変更	○	○	○	－	－
いくつかの関数をPARALLEL UNSAFEに変更	○	○	○	－	－
アプリケーションの接続先切り替え機能の指定方法の変更	◎	◎	－	－	－
C言語およびCOBOL言語による埋め込みSQLアプリケーションのOSS PostgreSQL互換	◎	◎	－	－	－
Vertical Clustered Index(VCI)の選択条件変更	◎	◎	－	－	－
システム管理関数の名称変更	○	○	－	－	－
システム情報関数の返却項目の名称変更	○	○	－	－	－
FROM句にシーケンスを指定した場合の返却列の変更	○	○	－	－	－
範囲外の値を入力した場合のto_date関数とto_timestamp関数の動作変更	○	○	－	－	－
選択リストに集合を返す関数を指定した場合の動作変更	○	○	－	－	－

項目	移行前のバージョン				
	9.5	9.6	10	11	12
同一テーブルが複数回更新される場合の文単位のトリガの動作変更	○	○	－	－	－
C言語関数におけるVersion 0呼び出し規約の削除	○	○	－	－	－
Oracleデータベース互換機能のrpad関数にマルチバイト文字を入力した場合の返却結果の変更	○	○	－	－	－
money型をbigint型で除算する場合の演算方法の変更	○	○	－	－	－
cubeモジュールの~>演算子の振る舞いの変更	○	○	－	－	－
tsearch2モジュールの削除	○	○	－	－	－
psqlコマンドの特別変数に不当な値を指定した場合の動作変更	○	○	－	－	－
psqlコマンドの¥dおよび¥ddの出力形式の変更	○	○	－	－	－
PL/Tclの自動ロード機能の削除	○	○	－	－	－
PL/Pythonにおける複合型配列の扱いの変更	○	○	－	－	－
DHパラメータのサイズ変更	○	○	－	－	－
to_char関数におけるマイナス符号の扱いの変更	○	－	－	－	－
無限大のタイムスタンプを入力した場合のextract関数の戻り値の変更	○	－	－	－	－
三角関数のPOSIX準拠	○	－	－	－	－
行集合を返す関数がNULLを返せるように変更	○	－	－	－	－
PL/pgSQLのRAISEコマンドにおけるCONTEXTメッセージを出力するように変更	○	－	－	－	－
テキスト検索パーサにおける数字から始まるメールアドレスやホスト名の解析方法の変更	○	－	－	－	－
関数をORDER BYソートの順序で実行するように変更	○	－	－	－	－
日付やタイムスタンプの限界値チェックの強化	○	－	－	－	－
OIDを取得するシステムカタログ情報関数における引数のデータ型の変更	○	－	－	－	－
不当な値を入力した場合のシステムカタログ情報関数の動作変更	○	－	－	－	－
レプリケーションSQL関数pg_replication_origin_xact_resetの引数の削除	○	－	－	－	－
psqlコマンドの-cオプションを指定した場合に起動用ファイルを読み込むように変更	○	－	－	－	－
画面表示モードが自動設定の場合のpsqlコマンドの表示方法の変更	○	－	－	－	－
psqlコマンドの¥df+の出力情報の追加	○	－	－	－	－
環境変数PAGERに空文字列または全て空白文字の文字列を指定した場合のpsqlコマンドの表示方法の変更	○	－	－	－	－
pgbenchコマンドの独自スクリプトの書式変更	○	－	－	－	－
pgbenchコマンドのスクリプトファイルメタコマンド¥setrandomの削除	○	－	－	－	－
hstoreのhstore_to_jsonb_loose関数の数値判別の厳格化	○	－	－	－	－

項目	移行前のバージョン				
	9.5	9.6	10	11	12
pageinspectのheap_page_items関数の返却情報の追加	○	－	－	－	－
Oracleデータベース互換機能利用時の変更	◎	－	－	－	－

◎:FUJITSU Enterprise Postgresによる非互換あり

○:OSSによる非互換あり

－:非互換なし

2.2.1 ecpg/ecobpgの「elif」文の動作の変更

非互換

FUJITSU Enterprise Postgres 13では、「elif」文が正しく動作されるようになります。

FUJITSU Enterprise Postgres 12以前の場合

「ifdef/ifndef」の条件が満たされるかどうか問わず、「elif」文の判断が実施されます。

[例]

```
000000     EXEC SQL ifdef DEF1 END-EXEC.
000000     操作(1).
000000     EXEC SQL elif DEF2 END-EXEC.
000000     操作(2).
000000     EXEC SQL else END-EXEC.
000000     操作(3).
```

「DEF1」および「DEF2」が真の場合、「操作(1)」と「操作(2)」の両方が実行されます。

FUJITSU Enterprise Postgres 13の場合

「ifdef/ifndef」の条件が満たされる場合、「elif」文の判断が実施されません。

[例]

```
000000     EXEC SQL ifdef DEF1 END-EXEC.
000000     操作(1).
000000     EXEC SQL elif DEF2 END-EXEC.
000000     操作(2).
000000     EXEC SQL else END-EXEC.
000000     操作(3).
```

「DEF1」および「DEF2」が真の場合、「操作(1)」だけが実行されます。

対処方法

既存のアプリケーションを確認し、elif文が正しく動作するように修正してください。

2.2.2 ltreeでlquery構文のNOT(!)の処理の変更

非互換

FUJITSU Enterprise Postgres 13では、ltreeでlquery構文のNOT(!)の処理を変更します。

これまでlquery構文のNOT(!)の処理結果は誤りでしたが、修正します。

FUJITSU Enterprise Postgres 12以前の場合

[例]

```
=# SELECT 'a.b.c.d.e'::ltree ~ '*.!d.*';
 ?column?
```

```
----------
 f
(1 row)
```

FUJITSU Enterprise Postgres 13の場合

[例]

```
=# SELECT 'a.b.c.d.e'::ltree ~ '*.!d.*';
 ?column?
----------
 t
(1 row)
```

対処方法

ありません。

2.2.3 txid_で始まる関数名の変更

非互換

FUJITSU Enterprise Postgres 13では、txid_で始まるトランザクションID関数名を変更します。

互換性のために、変更前の関数名も利用できますが、変更後の関数名を利用することを推薦します。

関数名の変更

FUJITSU Enterprise Postgres 12以前	FUJITSU Enterprise Postgres 13
txid_current()	pg_current_xact_id()
txid_current_if_assigned()	pg_current_xact_id_if_assigned()
txid_current_snapshot()	pg_current_snapshot()
txid_snapshot_xip()	pg_snapshot_xip()
txid_snapshot_xmax()	pg_snapshot_xmax()
txid_snapshot_xmin()	pg_snapshot_xmin()
txid_visible_in_snapshot()	pg_visible_in_snapshot()
txid_status()	pg_xact_status()

対処方法

ありません。

2.2.4 psqlの¥dのTOASTテーブルに対する出力の改善

非互換

FUJITSU Enterprise Postgres 13では、TOASTテーブルに対して、psqlの¥dで情報を表示するとき、インデックス情報も報告されるようになります。

FUJITSU Enterprise Postgres 12以前の場合

psqlコマンドの「¥d」の出力情報に、列の情報だけが表示されます。

[例]

```
=# ¥d pg_toast.pg_toast_2619
TOAST table "pg_toast.pg_toast_2619"
   Column   |  Type
------------+---------
 chunk_id   | oid
```

```
 chunk_seq  | integer
 chunk_data | bytea
```

FUJITSU Enterprise Postgres 13の場合

psqlコマンドの「¥d」の出力情報に、インデックスなどの情報が追加されます。

[例]

```
=# ¥d pg_toast.pg_toast_2619
TOAST table "pg_toast.pg_toast_2619"
   Column   |  Type
------------+---------
 chunk_id   | oid
 chunk_seq  | integer
 chunk_data | bytea
Owning table: "pg_catalog.pg_statistic"
Indexes:
    "pg_toast_2619_index" PRIMARY KEY, btree (chunk_id, chunk_seq)
```

対処方法

ありません。

2.2.5 psqlの¥eコマンドの編集内容の再表示の改善

非互換

FUJITSU Enterprise Postgres 13では、psqlの¥eコマンドのエディタを終了するとき、SQL文が終了していない場合(例:セミコロンがない場合)、SQL文の内容を表示されるように変更します。

FUJITSU Enterprise Postgres 12以前の場合

SQL文の内容が表示されません。

[例]

```
=# ¥e
--編集画面が表示されて、下記のような「;」がないSQL文を入力して、編集を終了します。
select 1,
2
-#
```

先の編集画面に入力したSQL文が表示されません。

FUJITSU Enterprise Postgres 13の場合

SQL文の内容が表示されます。

[例]

```
=# ¥e
--編集画面が表示されて、下記のような「;」がないSQL文を入力して、編集を終了します。
select 1,
2
=# select 1,
2
-#
```

先の編集画面に入力したSQL文が表示されます。

対処方法

ありません。

2.2.6 全文検索の問い合わせの結果の変更

非互換

FUJITSU Enterprise Postgres 13では、全文検索の問い合わせを正しく処理するように修正します。

これまで正しく動作できない問い合わせがありましたが、正しく結果が取得できるように修正します。

例:「NOT」の問い合わせ文は正しい結果が取得できるようになります。

FUJITSU Enterprise Postgres 12以前の場合

インデックスを使う場合、問い合わせで正しい結果が取得できない場合があります。

[例]

```
=# CREATE TABLE test (test tsvector);
=# INSERT INTO test VALUES ('crew:1C shuttl:2C'::tsvector);
=# CREATE INDEX ON test USING gin (test);
=# SET enable_seqscan TO off;
=# SELECT * FROM test WHERE test @@ to_tsquery('shuttle & !crew:a');
 test
-----
(0 rows)
```

FUJITSU Enterprise Postgres 13の場合

インデックスを使う場合、問い合わせで正しい結果が取得できます。

[例]

```
=# CREATE TABLE test (test tsvector);
=# INSERT INTO test VALUES ('crew:1C shuttl:2C'::tsvector);
=# CREATE INDEX ON test USING gin (test);
=# SET enable_seqscan TO off;
=# SELECT * FROM test WHERE test @@ to_tsquery('shuttle & !crew:a');
          test
-----------------------
 'crew':1C 'shuttl':2C
(1 row)
```

対処方法

ありません。

2.2.7 DECLARE CURSOR ... WITH HOLDの動作の変更

非互換

FUJITSU Enterprise Postgres 13では、インデックス式およびマテリアライズドビューの定義内で、DECLARE CURSOR ... WITH HOLDと遅延トリガの発行を禁止します。

FUJITSU Enterprise Postgres 12以前の場合

マテリアライズドビューの定義内で、DECLARE CURSOR ... WITH HOLDが実施可能です。

[例]

```
=# CREATE GROUP regress_priv_group1;
=# CREATE ROLE regress_sro_user;
=# CREATE FUNCTION unwanted_grant() RETURNS void LANGUAGE sql AS
-# 'GRANT regress_priv_group1 TO regress_sro_user';
=# CREATE FUNCTION mv_action() RETURNS bool LANGUAGE sql AS
-# 'DECLARE c CURSOR WITH HOLD FOR SELECT unwanted_grant(); SELECT true';
=# CREATE MATERIALIZED VIEW sro_mv AS SELECT mv_action() WITH NO DATA;
=# REFRESH MATERIALIZED VIEW sro_mv;
REFRESH MATERIALIZED VIEW
```

FUJITSU Enterprise Postgres 13の場合

マテリアライズドビュー定義内で、DECLARE CURSOR ... WITH HOLDが実施不可です。

[例]

```
=# CREATE GROUP regress_priv_group1;
=# CREATE ROLE regress_sro_user;
=# CREATE FUNCTION unwanted_grant() RETURNS void LANGUAGE sql AS
-# 'GRANT regress_priv_group1 TO regress_sro_user';
=# CREATE FUNCTION mv_action() RETURNS bool LANGUAGE sql AS
-# 'DECLARE c CURSOR WITH HOLD FOR SELECT unwanted_grant(); SELECT true';
=# CREATE MATERIALIZED VIEW sro_mv AS SELECT mv_action() WITH NO DATA;
=# REFRESH MATERIALIZED VIEW sro_mv;
ERROR:  cannot create a cursor WITH HOLD within security-restricted operation
CONTEXT:  SQL function "mv_action" statement 1
```

対処方法

ありません。

2.2.8 psqlの¥gsetコマンドで特別変数の変更の防止

非互換

FUJITSU Enterprise Postgres 13では、psqlの¥gsetコマンドで特別変数を変更できないようにします。

FUJITSU Enterprise Postgres 12以前の場合

「AUTOCOMMIT」などの特別変数は「¥gset」より変更できます。

[例]

```
=# SELECT 'off' as "AUTOCOMMIT"; ¥gset
 AUTOCOMMIT
------------
 off
(1 row)

=# ¥echo :AUTOCOMMIT
off
```

FUJITSU Enterprise Postgres 13の場合

「AUTOCOMMIT」などの特別変数は「¥gset」よりの変更は無視されます。

[例]

```
=# SELECT 'off' as "AUTOCOMMIT"; ¥gset
 AUTOCOMMIT
------------
 off
(1 row)

attempt to ¥gset into specially treated variable "AUTOCOMMIT" ignored
postgres=# ¥echo :AUTOCOMMIT
on
```

対処方法

ありません。

2.2.9 異なるデータ型の日付を比較する際の動作の変更

非互換

FUJITSU Enterprise Postgres 13では、異なるデータ型の日付を比較する際の動作を変更します。

date型とtimestamp型、timestamp型とtimestamptz型、またはdate型とtimestampz型のデータを比較するとき、有効範囲を超える場合でも、エラーにならないように変更します

FUJITSU Enterprise Postgres 12以前の場合

有効範囲を超えると、エラーが発生します。

[例]

```
postgres=# SELECT '2202020-10-05'::date > '2020-10-05'::timestamp;
ERROR:  date out of range for timestamp
```

FUJITSU Enterprise Postgres 13の場合

有効範囲を超えても、比較できます。

[例]

```
postgres=# SELECT '2202020-10-05'::date > '2020-10-05'::timestamp;
 ?column?
----------
 t
(1 row)
```

対処方法

ありません。

2.2.10 to_date()とto_timestamp()関数の変更

非互換

FUJITSU Enterprise Postgres 13では、to_date()とto_timestamp()で、マイナス年から紀元前日付への変換処理を正しく修正します。

FUJITSU Enterprise Postgres 12以前の場合

マイナス年から紀元前日付が誤って変換されます。

[例]

```
=# SELECT to_date('-1-01-01','yyyy-mm-dd');
    to_date
---------------
 0002-01-01 BC
(1 row)

=# SELECT to_timestamp('-44-02-01 11:12:13 BC', 'YYYY-MM-DD HH24:MI:SS BC');
          to_timestamp
-----------------------------------
 0045-02-01 11:12:13-04:56:02 BC
(1 row)
```

FUJITSU Enterprise Postgres 13の場合

マイナス年から紀元前日付が正しく変換されます。

[例]

```
=# SELECT to_date('-1-01-01','yyyy-mm-dd');
    to_date
---------------
 0001-01-01 BC
(1 row)

=# SELECT to_timestamp('-44-02-01 11:12:13 BC', 'YYYY-MM-DD HH24:MI:SS BC');
```

```
         to_timestamp
------------------------------
 0044-02-01 11:12:13-04:56:02
(1 row)
```

対処方法

ありません。

2.2.11 pg_stat_statementsのSELECT問い合わせ実行の統計処理の変更

非互換

FUJITSU Enterprise Postgres 13では、pg_stat_statementsでSELECT問い合わせ実行に関して、「FOR UPDATE」オプションがあるかどうかを区別して、統計情報を収集します。

FUJITSU Enterprise Postgres 12以前の場合

SELECT問い合わせの「FOR UPDATE」オプションを区別せず、統計情報を収集します。

[例]

```
=# SELECT * FROM pgss_a JOIN pgss_b ON pgss_b.a_id = pgss_a.id;
...
=# SELECT * FROM pgss_a JOIN pgss_b ON pgss_b.a_id = pgss_a.id FOR UPDATE;
...
=# SELECT calls, query FROM pg_stat_statements ORDER BY query COLLATE "C";
 calls |                            query
-------+-----------------------------------------------------------------
     2 | SELECT * FROM pgss_a JOIN pgss_b ON pgss_b.a_id = pgss_a.id;
     1 | SELECT pg_stat_statements_reset();
(2 rows)
```

FUJITSU Enterprise Postgres 13の場合

SELECT問い合わせの「FOR UPDATE」オプションを区別して、統計情報を収集します。

[例]

```
=# SELECT * FROM pgss_a JOIN pgss_b ON pgss_b.a_id = pgss_a.id;
...
=# SELECT * FROM pgss_a JOIN pgss_b ON pgss_b.a_id = pgss_a.id FOR UPDATE;
...
=# SELECT calls, query FROM pg_stat_statements ORDER BY query COLLATE "C";
 calls |                                 query
-------+-------------------------------------------------------------------------
     1 | SELECT * FROM pgss_a JOIN pgss_b ON pgss_b.a_id = pgss_a.id
     1 | SELECT * FROM pgss_a JOIN pgss_b ON pgss_b.a_id = pgss_a.id FOR UPDATE
     1 | SELECT pg_stat_statements_reset()
(3 rows)
```

対処方法

ありません。

2.2.12 CREATE EXTENSIONのFROMオプションをサポートしないように変更

非互換

FUJITSU Enterprise Postgres 13では、CREATE EXTENSIONのFROMオプションをサポートしないように変更します。

対処方法

ありません。

2.2.13 pg_available_extension_versions()の結果に情報を追加

非互換

FUJITSU Enterprise Postgres 13では、pg_available_extension_versions()の結果に「trusted」列を追加します。

対処方法

ありません。

2.2.14 pg_pltemplateカタログの削除

非互換

FUJITSU Enterprise Postgres 13では、pg_pltemplateカタログが削除されます。

対処方法

ありません。

2.2.15 psqlコマンドの「--help」の出力情報の追加

非互換

FUJITSU Enterprise Postgres 13では、psqlコマンドの「--help」の出力情報にPostgreSQLのホームページのURLが追加されます。

FUJITSU Enterprise Postgres 12以前の場合

psqlコマンドの「--help」の出力情報にPostgreSQLのホームページのURLがありません。

[例]

```
$ psql -help
psql is the PostgreSQL interactive terminal.

Usage:
  psql [OPTION]... [DBNAME [USERNAME]]

eneral options:
(省略)
Report bugs to <pgsql-bugs@lists.postgresql.org>.
```

FUJITSU Enterprise Postgres 13の場合

psqlコマンドの「--help」の出力情報にPostgreSQLのホームページのURLが追加されます。

[例]

```
$ psql -help
psql is the PostgreSQL interactive terminal.

Usage:
  psql [OPTION]... [DBNAME [USERNAME]]

General options:
(省略)
Report bugs to <pgsql-bugs@lists.postgresql.org>.
PostgreSQL home page: <https://www.postgresql.org/>
```

対処方法

ありません。

2.2.16 psqlコマンドの¥d+の出力情報の追加

非互換

FUJITSU Enterprise Postgres 13では、psqlコマンドの「¥d+」の出力情報に永続性を表す「Persistence」列が追加されます。

FUJITSU Enterprise Postgres 12以前の場合

「Persistence」列の情報がありません。

[例]

```
=# ¥dt+
                        List of relations
 Schema |      Name       | Type  |  Owner   |    Size    | Description
--------+-----------------+-------+----------+------------+-------------
 public | test1           | table | postgres | 0 bytes    |
 public | utest1          | table | postgres | 0 bytes    |
```

FUJITSU Enterprise Postgres 13の場合

「Persistence」列の情報が追加されます。

[例]

```
=# ¥dt+
                                List of relations
 Schema |      Name       | Type  |  Owner   | Persistence |    Size    | Description
--------+-----------------+-------+----------+-------------+------------+-------------
 public | test            | table | postgres | permanent   | 8192 bytes |
 public | utest1          | table | postgres | unlogged    | 0 bytes    |
```

対処方法

ありません。

2.2.17 psqlのデフォルトプロンプトにトランザクション状態の追加

非互換

FUJITSU Enterprise Postgres 13では、psqlのデフォルトプロンプトにトランザクション状態(「*」など)が追加されます。

これまではトランザクション状態はプロンプト変数に「%x」で指定可能でしたが、デフォルトで追加するように修正します。

FUJITSU Enterprise Postgres 12以前の場合

トランザクション状態(「*」など)が表示されません。

[例]

```
=# begin;
BEGIN
=# commit;
COMMIT
```

FUJITSU Enterprise Postgres 13の場合

トランザクション状態(「*」など)が表示されます。

[例]

```
=# begin;
BEGIN
=*# commit;
COMMIT
```

対処方法

必要に応じて、psql のデフォルトプロンプトを変更してください。

2.2.18 CHECK制約の「NO INHERIT」プロパティを伝播するように変更

非互換

FUJITSU Enterprise Postgres 13では、「CREATE TABLE LIKE」より作成したテーブルにCHECK制約の「NO INHERIT」プロパティを伝播するように変更します。

FUJITSU Enterprise Postgres 12以前の場合

CHECK制約の「NO INHERIT」プロパティが伝播されません。

[例]

```
=# CREATE TABLE noinh_con_copy (a int CHECK (a > 0) NO INHERIT);
CREATE TABLE
=# CREATE TABLE noinh_con_copy1 (LIKE noinh_con_copy INCLUDING CONSTRAINTS);
CREATE TABLE
=# ¥d noinh_con_copy1
          Table "public.noinh_con_copy1"
 Column |  Type   | Collation | Nullable | Default
--------+---------+-----------+----------+---------
 a      | integer |           |          |
Check constraints:
    "noinh_con_copy_a_check" CHECK (a > 0)
```

FUJITSU Enterprise Postgres 13の場合

CHECK制約の「NO INHERIT」プロパティが伝播されます。

[例]

```
=# CREATE TABLE noinh_con_copy (a int CHECK (a > 0) NO INHERIT);
CREATE TABLE
=# CREATE TABLE noinh_con_copy1 (LIKE noinh_con_copy INCLUDING CONSTRAINTS);
CREATE TABLE
=# ¥d noinh_con_copy1
          Table "public.noinh_con_copy1"
 Column |  Type   | Collation | Nullable | Default
--------+---------+-----------+----------+---------
 a      | integer |           |          |
Check constraints:
    "noinh_con_copy_a_check" CHECK (a > 0) NO INHERIT
```

対処方法

必要に応じて、LIKE文で「INCLUDING CONSTRAINTS」を指定せず、「EXCLUDING CONSTRAINTS」を指定してテーブルを作成します。テーブルを作成した後、チェック制約を個別に追加してください。

2.2.19 opaque型がサポートされないように変更

非互換

FUJITSU Enterprise Postgres 13では、PostgreSQL 7.3より前でサポートされていたopaque型をサポートされないようになります。

対処方法

ありません。

2.2.20 旧版の外部キー制約の定義がサポートされないように変更

非互換

FUJITSU Enterprise Postgres 13では、PostgreSQL 7.3より前の構文を使用した外部キー制約の定義がサポートされないようになります。

対処方法

ありません。

2.2.21 旧版の演算子クラスの定義がサポートされないように変更

非互換

FUJITSU Enterprise Postgres 13では、PostgreSQL 8.0より前の構文を使用した下記の演算子クラスの定義がサポートされないようになります。

- network_ops
- timespan_ops
- datetime_ops
- lztext_ops
- timestamp_ops
- bigbox_ops

FUJITSU Enterprise Postgres 12以前の場合

インデックスを作成するとき、上記のサポートされない演算子クラスを指定すると、デフォルトの演算子クラスを使用して、インデックスを作成します。

[例]

```
=# CREATE TABLE test(c1 int);
CREATE TABLE
=# CREATE INDEX index_test ON test(c1 network_ops);
CREATE INDEX
```

FUJITSU Enterprise Postgres 13の場合

インデックスを作成するとき、上記のサポートされない演算子クラスを指定すると、エラーメッセージが出力されます。

[例]

```
=# CREATE TABLE test(c1 int);
CREATE TABLE
=# CREATE INDEX index_test ON test(c1 network_ops);
ERROR:  operator class "network_ops" does not exist for access method "btree"
```

対処方法

ありません。

2.2.22 スーパーユーザーのみssl_passphrase_commandの設定値を参照できるように変更

非互換

FUJITSU Enterprise Postgres 13では、スーパーユーザーのみssl_passphrase_commandの設定値を参照できるように変更します。

対処方法

ありません。

2.2.23 pg_triggerカタログにtgparentid列を追加

非互換

FUJITSU Enterprise Postgres 13では、pg_triggerカタログにtgparentid列を追加します。

対処方法

ありません。

2.2.24 bt_metap()/bt_page_items()関数の変更

非互換

FUJITSU Enterprise Postgres 13では、bt_metap()/bt_page_items()関数で返却するカラムを追加します。

- bt_metap()関数は「allequalimage」カラムを追加で返却します。
- bt_page_items()関数は「dead」、「htid」と「tids」3つのカラムを追加で返却します。

対処方法

ありません。

2.2.25 統計情報ビューpg_stat_activityにleader_pid列を追加

非互換

FUJITSU Enterprise Postgres 13では、統計情報ビューpg_stat_activityにleader_pid列が追加されます。

対処方法

ありません。

2.2.26 デフォルト以外の照合順序を使用するインデックスの制約動作の変更

非互換

FUJITSU Enterprise Postgres 13では、デフォルト以外の照合順序を使用するインデックスは、テーブルのユニークキーまたはプライマリキーの制約として、追加できないように変更します。

FUJITSU Enterprise Postgres 12以前の場合

デフォルト以外の照合順序を使用するインデックスは、テーブルのユニークキーまたはプライマリキーの制約として追加できます。

[例]

```
=# CREATE TABLE cwi_test( a int , b varchar(10), c char);
CREATE TABLE
=# CREATE UNIQUE INDEX cwi_uniq4_idx ON cwi_test(b collate "POSIX");
CREATE INDEX
=# ALTER TABLE cwi_test ADD UNIQUE USING INDEX cwi_uniq4_idx;
ALTER TABLE
```

FUJITSU Enterprise Postgres 13の場合

デフォルト以外の照合順序を使用するインデックスは、テーブルのユニークキーまたはプライマリキーの制約として追加できません。

[例]

```
=# CREATE TABLE cwi_test( a int , b varchar(10), c char);
CREATE TABLE
=# CREATE UNIQUE INDEX cwi_uniq4_idx ON cwi_test(b collate "POSIX");
CREATE INDEX
=# ALTER TABLE cwi_test ADD UNIQUE USING INDEX cwi_uniq4_idx;
```

```
ERROR:  index "cwi_uniq4_idx" column number 1 does not have default sorting behavior
LINE 1: ALTER TABLE cwi_test ADD UNIQUE USING INDEX cwi_uniq4_idx;
                                                ^
DETAIL:  Cannot create a primary key or unique constraint using such an index.
```

対処方法

ありません。

2.2.27 ALTER MATERIALIZED VIEW ... RENAME COLUMN実施時のコマンドタグの変更

非互換

FUJITSU Enterprise Postgres 13では、「ALTER MATERIALIZED VIEW ... RENAME COLUMN」を実施時に、コマンドタグが正しく表示されるようになります。

FUJITSU Enterprise Postgres 12以前の場合

「ALTER TABLE」のコマンドタグが表示されます。

[例]

```
=# ALTER MATERIALIZED VIEW sales_summary rename seller_no to selno;
ALTER TABLE
```

FUJITSU Enterprise Postgres 13の場合

「ALTER MATERIALIZED VIEW」のコマンドタグが表示されます。

[例]

```
=# ALTER MATERIALIZED VIEW sales_summary rename seller_no to selno;
ALTER MATERIALIZED VIEW
```

対処方法

ありません。

2.2.28 ALTER FOREIGN TABLE ... RENAME COLUMN実施時のコマンドタグの変更

非互換

FUJITSU Enterprise Postgres 13では、「ALTER FOREIGN TABLE ... RENAME COLUMN」を実施時に、コマンドタグが正しく表示されるようになります。

FUJITSU Enterprise Postgres 12以前の場合

「ALTER TABLE」のコマンドタグが表示されます。

[例]

```
=# ALTER foreign table ftb rename name to name1;
ALTER TABLE
```

FUJITSU Enterprise Postgres 13の場合

「ALTER FOREIGN TABLE」のコマンドタグが表示されます。

[例]

```
=# ALTER foreign table bb rename name to name1;
ALTER FOREIGN TABLE
```

対処方法

ありません。

2.2.29 補助プロセスのPIDはpg_stat_sslに記録しないように変更

非互換

FUJITSU Enterprise Postgres 13では、補助プロセスのPIDはpg_stat_sslに記録しないように変更します。

対処方法

ありません。

2.2.30 補助プロセスのPIDはpg_stat_gssapiに記録しないように変更

非互換

FUJITSU Enterprise Postgres 13では、補助プロセスのPIDはpg_stat_gssapiに記録しないように変更します。

対処方法

ありません。

2.2.31 ltreeにおいて、中括弧付きアスタリスクの処理を変更

非互換

FUJITSU Enterprise Postgres 13では、ltreeにおいて、中括弧付きアスタリスクの処理を変更します。

例えば、「*{2}.*{3}」と「.*{5}」は同じ結果になります。

これまで、「*{2}.*{3}」のように指定する場合の動作は誤りでしたが、正しく修正します。

FUJITSU Enterprise Postgres 12以前の場合

[例]

```
=# SELECT 'a.b.c.d.e'::ltree ~ 'a.*{1}.*{4}';
 ?column?
----------
 t
(1 row)
=# SELECT 'a.b.c.d.e'::ltree ~ 'a.*{5}';
 ?column?
----------
 f
(1 row)
```

FUJITSU Enterprise Postgres 13の場合

[例]

```
=# SELECT 'a.b.c.d.e'::ltree ~ 'a.*{1}.*{4}';
 ?column?
----------
 f
(1 row)
=# SELECT 'a.b.c.d.e'::ltree ~ 'a.*{5}';
 ?column?
----------
 f
(1 row)
```

対処方法

ありません。

2.2.32 json_to_tsvector/jsonb_to_tsvector関数のfilterのチェックの強化

非互換

FUJITSU Enterprise Postgres 13では、json_to_tsvector/jsonb_to_tsvector関数のfilterのチェックを強化します。

FUJITSU Enterprise Postgres 12以前の場合

json_to_tsvector/jsonb_to_tsvector関数のfilterに「strin」から始まる文字列を指定した場合、「string」として処理します。

[例]

```
=# select json_to_tsvector('english', '{"a": "The Fat Rats", "b": 123}'::json, '["strinX", "numeric"]');
    json_to_tsvector
--------------------------
 '123':5 'fat':2 'rat':3
(1 row)
```

FUJITSU Enterprise Postgres 13の場合

json_to_tsvector/jsonb_to_tsvector関数のfilterに「strin」から始まる「string」以外の文字列を指定した場合、エラーになります。

[例]

```
=# select json_to_tsvector('english', '{"a": "The Fat Rats", "b": 123}'::json, '["strinX", "numeric"]');
ERROR:  wrong flag in flag array: "strinX"
HINT:  Possible values are: "string", "numeric", "boolean", "key", and "all".
```

対処方法

ありません。

2.2.33「SIMILAR TO ... ESCAPE NULL 」が必ず NULL を返すように変更

非互換

FUJITSU Enterprise Postgres 13では、「SIMILAR TO ... ESCAPE NULL」 が常に NULL を返すように変更します。

これまでの動作は標準SQLに従っていませんでしたが、正しく標準SQLに従うように変更します。

FUJITSU Enterprise Postgres 12以前の場合

「SIMILAR TO ... ESCAPE NULL」にする場合、デフォルトのエスケープ文字('¥')が使われます。

[例]

```
=# SELECT 'hello¥nworld' similar to 'hello¥¥nworld' escape null;
?column?
----------
t
(1 row)
```

FUJITSU Enterprise Postgres 13の場合

「SIMILAR TO ... ESCAPE NULL」にする場合、NULL を返します。

[例]

```
=# SELECT 'hello¥nworld' similar to 'hello¥¥nworld' escape null;
?column?
----------

(1 row)
```

対処方法

必要に応じて、「ESCAPE NULL」を削除してください。

2.2.34 Oracleデータベース互換機能のto_multibyte()にバックスラッシュを入力した際の出力結果の変更

非互換

FUJITSU Enterprise Postgres 13では、to_multibyte()に半角バックスラッシュが入力された際の出力を変更します。

FUJITSU Enterprise Postgres 12以前の場合

「¥」が出力されます。

[例]

```
postgres=# select to_multi_byte('¥');
 to_multi_byte
---------------
 ¥
(1 row)
```

FUJITSU Enterprise Postgres 13の場合

「＼」が出力されます。

[例]

```
postgres=# select to_multi_byte('¥');
 to_multi_byte
---------------
 ＼
(1 row)
```

対処方法

ありません。

2.2.35 Oracleデータベース互換機能のto_char()に浮動小数点型を入力した際の出力桁数の変更

非互換

FUJITSU Enterprise Postgres 13では、to_char()を用いた浮動小数点の文字列化処理を変更します。そのため出力される結果が変更される可能性があります。

FUJITSU Enterprise Postgres 12以前の場合

real型を使用した場合でも精度が保証されない桁まで表示されます。

[例]

```
select to_char(123.456::real);
  to_char
------------
123.456001
(1 row)
```

FUJITSU Enterprise Postgres 13の場合

real型を使用した場合には精度が保証される6桁が表示されます。

[例]

```
select to_char(123.456::real);
 to_char
```

```
---------
 123.456
(1 row)
```

対処方法

ありません。

2.2.36 データ秘匿化が可能なデータ型でない型を戻り値として返す演算子または関数を実行した場合に、エラーとなるように変更

非互換

FUJITSU Enterprise Postgres 12では、データ秘匿化が可能なデータ型でない型を戻り値として返す演算子または関数を実行した場合に、エラーとなるように変更します。

FUJITSU Enterprise Postgres 11以前の場合

下記の場合、3)の演算子または関数が、秘匿化されずに実行される場合があります。

これは、秘匿化対象列を含むデータが秘匿化されずに実行されるため、誤った結果です。

1) 秘匿化ポリシーを作成する。かつ、

2) SELECT文を実行する。かつ、

3) 2)のSELECT文の中に、演算子または関数を実行する。かつ、

4) 3)の演算子または関数の引数に副問い合わせを含む。かつ、

5) 3)の演算子または関数の引数に、1)で作成した秘匿化対象の列を含む。かつ、

6) 3)の演算子または関数の戻り値が、“運用ガイド”の“データ秘匿化が可能なデータ型”に記載されていない型を返す場合。

FUJITSU Enterprise Postgres 12の場合

3)の演算子または関数が、下記のエラーとなります。

```
ERROR:  The output data type is incompatible with the confidential policy.
HINT:  Data type of the result value(s) produced by expression/function using confidential columns is not supported by Data masking module. Consider removing confidential columns from such expressions/functions.
```

これは、“運用ガイド”の“データ秘匿化が可能なデータ型”に記載されていないデータ型を演算子または関数が返すため正しい結果です。

対処方法

下記のいずれかの対処をすることで、FUJITSU Enterprise Postgres 11以前と同一の実行結果を出力できるようにします。

1) 秘匿化ポリシーを変更し、当該SQLを実行するユーザに対して秘匿化が実施されないように設定する。

2) SQLを変更し、“運用ガイド”の“データ秘匿化が可能なデータ型”に記載されていない型を返す演算子または関数を使用しないようにする。

2.2.37 Oracleデータベース互換機能のUTL_FILEの削除

非互換

FUJITSU Enterprise Postgres 12では、Oracleデータベース互換機能のUTL_FILEを削除します。

対処方法

ありません。

2.2.38 psqlコマンドの¥conninfoの出力の変更

非互換

FUJITSU Enterprise Postgres 12では、psqlコマンドの¥conninfoの出力にIPアドレスを追加します。

FUJITSU Enterprise Postgres 11以前の場合

ホスト名を指定して、psqlコマンドでサーバに接続した後、¥conninfoで接続情報を表示する時、IPアドレスを表示しません。

[例]

```
=# ¥conninfo
You are connected to database "postgres" as user "postgres" on host "localhost" at port "27500".
```

FUJITSU Enterprise Postgres 12の場合

ホスト名を指定して、psqlコマンドでサーバに接続した後、¥conninfoで接続情報を表示する時、IPアドレスを表示します。

[例]

```
=# ¥conninfo
You are connected to database "postgres" as user "postgres" on host "localhost" (address "::1") at port "27500".
```

対処方法

ありません。

2.2.39 psqlコマンドの¥helpの出力の変更

非互換

FUJITSU Enterprise Postgres 12では、psqlコマンドの¥helpの出力にマニュアルページへのURLを追加します。

FUJITSU Enterprise Postgres 11以前の場合

¥helpの出力にマニュアルページへのURLを追加しません。

[例]

```
=# ¥h checkpoint
Command:     CHECKPOINT
Description: force a write-ahead log checkpoint
Syntax:
CHECKPOINT
```

FUJITSU Enterprise Postgres 12の場合

¥helpの出力にマニュアルページへのURLを追加します。

[例]

```
=# ¥h checkpoint
Command:     CHECKPOINT
Description: force a write-ahead log checkpoint
Syntax:
CHECKPOINT

URL: https://www.postgresql.org/docs/12/sql-checkpoint.html
```

対処方法

ありません。

2.2.40 XML関数のさまざまな不具合の修正

非互換

FUJITSU Enterprise Postgres 12では、XML関数のXMLTABLE、xpath、xmlexistsに対して、下記の不具合を修正します。

- ノードに何も出力されない場合がある不具合
- 予期しないエラーがスローされる不具合
- XML特殊文字の必要なエスケープが行われていなかった不具合

以下に、ノードに何も出力されない場合の不具合の修正例を示します。

FUJITSU Enterprise Postgres 11以前の場合

ノードに何も出力されない場合があります。

[例]

```
=# SELECT * FROM xmltable('/root' passing '<root><element><x>xxx</x>aaa</element></root>' COLUMNS element text);
 element
---------
 aaa
(1 row)
```

FUJITSU Enterprise Postgres 12の場合

ノードに何も出力されない場合がある不具合が修正されました。

[例]

```
=# SELECT * FROM xmltable('/root' passing '<root><element><x>xxx</x>aaa</element></root>' COLUMNS element text);
 element
---------
 xxxaaa
(1 row)
```

対処方法

ありません。

2.2.41 to_timestamp()/to_date()関数の変更

非互換

FUJITSU Enterprise Postgres 12では、to_timestamp()/to_date()関数ではテンプレート書式の不一致を無視します。

FUJITSU Enterprise Postgres 11以前の場合

to_timestamp()/to_date()関数はテンプレート書式の不一致を無視しません。

[例]

```
=# SELECT to_timestamp('2011-12', 'YYYY- MM');
      to_timestamp
------------------------
 2011-02-01 00:00:00+08
(1 row)

=# SELECT to_date('2011-12', 'YYYY- MM');
  to_date
------------
 2011-02-01
(1 row)
```

FUJITSU Enterprise Postgres 12の場合

to_timestamp()/to_date()関数はテンプレート書式の不一致を無視します。

[例]

```
=# SELECT to_timestamp('2011-12', 'YYYY- MM');
      to_timestamp
------------------------
```

```
 2011-12-01 00:00:00+08
(1 row)

=# SELECT to_date('2011-12', 'YYYY- MM');
  to_date
------------
 2011-12-01
(1 row)
```

対処方法

ありません。

2.2.42 SERIALIZABLE分離モードでパラレルクエリが使用できるように変更

非互換

FUJITSU Enterprise Postgres 12では、SERIALIZABLE分離モードでパラレルクエリが使用できるようになります。以前はSERIALIZABLEでは並列処理ができませんでした。

FUJITSU Enterprise Postgres 11以前の場合

SERIALIZABLE分離モードでパラレルクエリが使用できません。

FUJITSU Enterprise Postgres 12の場合

SERIALIZABLE分離モードでパラレルクエリが使用できます。

対処方法

ありません。

2.2.43 単一の子しか持たないパーティションテーブルおよびUNION ALLの問い合わせの最適化の改善

非互換

FUJITSU Enterprise Postgres 12では、単一の子しか持たないパーティションテーブルおよびUNION ALLの問い合わせの最適化が改善されます。

FUJITSU Enterprise Postgres 11以前の場合

Append 処理があります。

[例]

```
=# CREATE TABLE tb(c1 int) PARTITION BY RANGE (c1);
CREATE TABLE
=# CREATE TABLE tb_01 PARTITION OF tb FOR VALUES FROM (0) TO (100);
CREATE TABLE
=# EXPLAIN SELECT * from tb;
                          QUERY PLAN
---------------------------------------------------------------
 Append  (cost=0.00..48.25 rows=2550 width=4)
   ->  Seq Scan on tb_01  (cost=0.00..35.50 rows=2550 width=4)
(2 rows)
```

FUJITSU Enterprise Postgres 12の場合

Append 処理が省略されます。

[例]

```
=# CREATE TABLE tb(c1 int) PARTITION BY RANGE (c1);
CREATE TABLE
=# CREATE TABLE tb_01 PARTITION OF tb FOR VALUES FROM (0) TO (100);
```

```
CREATE TABLE
=# EXPLAIN SELECT * from tb;
                        QUERY PLAN
-----------------------------------------------------------
 Seq Scan on tb_01  (cost=0.00..35.50 rows=2550 width=4)
(1 row)
```

対処方法

ありません。

2.2.44 psqlコマンドの¥dと¥zのパーティションテーブルに対する表示の改善

非互換

FUJITSU Enterprise Postgres 12では、psqlコマンドに¥dと¥zを指定した場合に表示されるパーティションテーブルに対して、パーティションテーブルであることが表示されるようになります。

FUJITSU Enterprise Postgres 11以前の場合

パーティションテーブルであることが表示されません。

[例]

```
=# CREATE TABLE tb(c1 int) PARTITION BY RANGE (c1) tablespace test_tsp;
CREATE TABLE
postgres=# ¥d
         List of relations
 Schema |  Name   | Type  | Owner
--------+---------+-------+--------
 public | tb      | table | postgres
 (1 rows)

postgres=# ¥z
                             Access privileges
 Schema |  Name   | Type  | Access privileges | Column privileges | Policies
--------+---------+-------+-------------------+-------------------+----------
 public | tb      | table |                   |                   |
 (1 rows)
```

FUJITSU Enterprise Postgres 12の場合

パーティションテーブルであることが表示されます。

[例]

```
=# CREATE TABLE tb(c1 int) PARTITION BY RANGE (c1) tablespace test_tsp;
CREATE TABLE
=# ¥d
               List of relations
 Schema |  Name   |       Type        |  Owner
--------+---------+-------------------+----------
 public | tb      | partitioned table | postgres
 (1 rows)

=# ¥z
                                    Access privileges
 Schema |  Name   |       Type        | Access privileges | Column privileges | Policies
--------+---------+-------------------+-------------------+-------------------+----------
 public | tb      | partitioned table |                   |                   |
(1 rows)
```

対処方法

ありません。

2.2.45 システムビューpg_indexesにパーティションインデックスを含むように変更

非互換

FUJITSU Enterprise Postgres 12では、システムビューpg_indexesにパーティションインデックスも含まれるようになります。

対処方法

ありません。

2.2.46 パーティションテーブルにテーブルスペースの指定ができるように変更

非互換

FUJITSU Enterprise Postgres 12では、CREATE TABLEでのテーブルスペース指定がパーティションテーブルに対しても可能になります。この指定は子パーティションにも反映されます。

FUJITSU Enterprise Postgres 11以前の場合

パーティションテーブルにテーブルスペースを指定しても有効になりません。

[例]

```
=# CREATE TABLE tb(c1 int) PARTITION BY RANGE (c1) tablespace test_tsp;
CREATE TABLE
=# CREATE TABLE tb_01 PARTITION OF tb FOR VALUES FROM (0) TO (100);
CREATE TABLE
=# select tablename,tablespace from pg_tables where tablename like 'tb%';
 tablename | tablespace
-----------+------------
 tb        |
 tb_01     |
(2 rows)
```

FUJITSU Enterprise Postgres 12の場合

パーティションテーブルにテーブルスペースを指定できます。この指定は子パーティションにも反映されます。

[例]

```
=# CREATE TABLE tb(c1 int) PARTITION BY RANGE (c1) tablespace test_tsp;
CREATE TABLE
=# CREATE TABLE tb_01 PARTITION OF tb FOR VALUES FROM (0) TO (100);
CREATE TABLE
=# select tablename,tablespace from pg_tables where tablename like 'tb%';
 tablename | tablespace
-----------+------------
 tb        | test_tsp
 tb_01     | test_tsp
(2 rows)
```

対処方法

ありません。

2.2.47 パーティション境界に任意の式を記述できるように変更

非互換

FUJITSU Enterprise Postgres 12では、パーティション境界に任意の式を記述できるようになります。これまでは、パーティション境界に定数しか指定できませんでした。

FUJITSU Enterprise Postgres 11以前の場合

パーティション境界に定数しか指定できません。

[例]

```
=# CREATE TABLE tb(c1 timestamp) PARTITION BY RANGE (c1);
CREATE TABLE
=# CREATE TABLE tb_0 PARTITION OF tb FOR VALUES FROM (CURRENT_DATE::timestamp) TO (CURRENT_DATE::timestamp + '1
year'::interval);
ERROR:  syntax error (10474) at or near "CURRENT_DATE" (10620)
LINE 1: ...REATE TABLE tb_0 PARTITION OF tb FOR VALUES FROM (CURRENT_DA...
                                                             ^
```

FUJITSU Enterprise Postgres 12の場合

パーティション境界に任意の式が指定できます。

[例]

```
=# CREATE TABLE tb(c1 timestamp) PARTITION BY RANGE (c1);
CREATE TABLE
=# CREATE TABLE tb_0 PARTITION OF tb FOR VALUES FROM (CURRENT_DATE::timestamp) TO (CURRENT_DATE::timestamp + '1
year'::interval);
CREATE TABLE
```

対処方法

ありません。

2.2.48 パーティションテーブルを参照する外部キーのサポート

非互換

FUJITSU Enterprise Postgres 12では、パーティションテーブルを参照する外部キーに対応します。

FUJITSU Enterprise Postgres 11以前の場合

パーティションテーブルを参照する外部キーがサポートされません。

[例]

```
=# CREATE TABLE tb(c1 int PRIMARY KEY) PARTITION BY RANGE(c1);
CREATE TABLE
=# CREATE TABLE tb1(c1 int,c2 int, FOREIGN KEY(c2) REFERENCES tb(c1));
ERROR:  cannot reference partitioned table "tb" (19146)
```

FUJITSU Enterprise Postgres 12の場合

パーティションテーブルを参照する外部キーがサポートされます。

[例]

```
=# CREATE TABLE tb(c1 int PRIMARY KEY) PARTITION BY RANGE(c1);
CREATE TABLE
=# CREATE TABLE tb1(c1 int,c2 int, FOREIGN KEY(c2) REFERENCES tb(c1));
CREATE TABLE
```

対処方法

ありません。

2.2.49 動的共有メモリを無効化する設定の廃止

非互換

FUJITSU Enterprise Postgres 12では、dynamic_shared_memory_typeパラメータにnoneが指定できなくなります。

対処方法

ありません。

2.2.50 information_schemaビューで、オブジェクト名の列をvarchar型ではなく、name型として扱うように変更

非互換

SQL標準ではinformation_schemaビューのオブジェクト名の列はドメイン型sql_identifierとして宣言されています。PostgreSQLでの実際の元となるカタログ列はname型です。FUJITSU Enterprise Postgres 12では、ドメイン型sql_identifierの元となる型をvarcahrからnameに変更します。name型の列はデフォルトでデータベースの照合順序ではなく、C照合順序を使うようになります。

対処方法

name型の列に対する正規表現でデータベースの照合順序の振る舞いをさせる場合は、「COLLATE "default"」を加えてください。

2.2.51 テーブルのname型の列がデフォルトでC照合順序を持つように変更

非互換

FUJITSU Enterprise Postgres 12では、nameデータ型に対する比較演算子は、常にCではなく、任意の照合順序が使えるようになります。これまでの問い合わせの意味を維持するため、name型の列は明示的にC照合順序を持つと印付けされるようになります。これにより、name型の列に対する正規表現演算子が、ロケールに依存する正規表現パターン(psqlの¥wで実行されるSQLなど)の振る舞いを決める際のデフォルトの照合順序がデータベースの照合順序からC照合順序に変更となります。

FUJITSU Enterprise Postgres 11以前の場合

name型の列は明示的にC照合順序を持つと印付けされません。

[例]

```
=# SELECT t.typname, t.typcollation, c.collname FROM pg_type t
-# JOIN pg_collation c ON (t.typcollation = c.oid) WHERE typname = 'name';
typname | typcollation | collname
---------+--------------+----------
(0 rows)
```

FUJITSU Enterprise Postgres 12の場合

name型の列は明示的にC照合順序を持つと印付けされます。

[例]

```
=# SELECT t.typname, t.typcollation, c.collname FROM pg_type t
-# JOIN pg_collation c ON (t.typcollation = c.oid) WHERE typname = 'name';
 typname | typcollation | collname
---------+--------------+----------
 name    |          950 | C
(1 row)
```

対処方法

name型の列に対する正規表現でC照合順序以外の振る舞いをさせる場合は、明示的にCOLLATE句を追加してください。

2.2.52 psqlの「¥pset format ..」コマンドでユニークでない略記の禁止

非互換

フォーマットの選択肢は先頭の1文字で指定できますが、これまでは複数候補がある場合にもエラーにならず、いずれかが選択されていました。FUJITSU Enterprise Postgres 12では、「¥pset format ..」コマンドでユニークでない略記が禁止されます。

FUJITSU Enterprise Postgres 11以前の場合

「¥pset format ..」コマンドでユニークでない略記が禁止されません。

[例]

```
=# ¥pset format a
Output format is aligned.
```

FUJITSU Enterprise Postgres 12の場合

「¥pset format ..」コマンドでユニークでない略記が禁止されます。

[例]

```
=# ¥pset format a
¥pset: ambiguous abbreviation "a" matches both "aligned" and "asciidoc"
```

対処方法

ありません。

2.2.53 pg_restoreの動作の変更

非互換

FUJITSU Enterprise Postgres 12では、pg_restoreコマンドでダンプ内容を標準出力に送るために「-f -」の指定が必要になります。

対処方法

ありません。

2.2.54 pg_verify_checksumsの改名

非互換

FUJITSU Enterprise Postgres 12では、コマンドラインツール pg_verify_checksums は pg_checksums に改名されます。

対処方法

ありません。

2.2.55 xpath()関数 や XMLTABLE による値の作成で出力を変更

非互換

FUJITSU Enterprise Postgres 12では、xpath()関数 や XMLTABLE による値の作成で出力を整形しなくなります。

対処方法

ありません。

2.2.56 SQLスタイルの substring(.. from … for ..)関数の動作の変更

非互換

FUJITSU Enterprise Postgres 12では、複数のマッチがありうる場合において、最初の副パターンが最大ではなく最小テキストにマッチするものとして処理されるようになります。

FUJITSU Enterprise Postgres 11以前の場合

マッチングの欲張り動作をしません。

[例]

```
=# SELECT substring('aaaa' from '%#"aa*#"%' for '#');
 substring
-----------
 a
(1 row)
```

FUJITSU Enterprise Postgres 12の場合

マッチングの欲張り動作をします。

[例]

```
=# SELECT substring('aaaa' from '%#"aa*#"%' for '#');
 substring
-----------
 aaaa
(1 row)
```

対処方法

ありません。

2.2.57 random()、setseed()関数がプラットフォーム間で統一

非互換

FUJITSU Enterprise Postgres 12では、random()、setseed()関数がプラットフォームによらず統一した動作をするようになりました。特定のシード値を伴うsetseed()の呼び出しに従って生成されるrandom()関数の値の並びは従来と違ったものになります。一方でこれは再現可能なものになります。以前はサーバ内の他の乱数使用に影響を受けるため、再現が保証されませんでした。SQLのrandom()関数は今後自身のセッション毎の固有の状態を持つようになります。

対処方法

ありません。

2.2.58 新たなアルゴリズムを使用することで、real型、double precision型の値を出力する際の性能を改善

非互換

これまでは出力される浮動小数点の値は、デフォルトでrealなら6桁、double precisionなら15桁に丸められ、extra_float_digits設定値に応じて補正されています。FUJITSU Enterprise Postgres 12では、extra_float_digitsが0より大きい場合にはいつでも、正確なバイナリ値の保持に必要な最小桁数のみが出力されるようになります。つまり、extra_float_digits設定値の1、2、3は同じ動作になります。また、extra_float_digitsのデフォルトは0から1に変更されます。extra_float_digitsに0以下を指定する場合の振る舞いは従来通りです。

また、浮動小数点の指数部の書式がプラットフォームによらず統一されます。3桁が必要ない限り、2桁が使われるようになります。以前はWindowsビルドの場合には常に3桁が出力されていました。

FUJITSU Enterprise Postgres 11以前の場合

Windowsでの出力。

[例]

```
=# SELECT 1.23e99::float;
  float8
-----------
 1.23e+099
(1 row)
```

Linuxでの出力。

[例]

```
=# SELECT 1.23e99::float;
  float8
-----------
 1.23e+99
(1 row)
```

FUJITSU Enterprise Postgres 12の場合

Windows/Linuxでの共通出力。

[例]

```
=# SELECT 1.23e99::float;
  float8
-----------
 1.23e+99
(1 row)
```

対処方法

ありません。

2.2.59 幾何関数と演算子のコードリファクタリング

非互換

FUJITSU Enterprise Postgres 12により、以前のリリースに対して、僅かに異なる結果が生じる可能性があります。特に、NaN、アンダーフロー、オーバーフロー、ゼロ除算に関する場合について、より一貫性のある処理が行われるようになります。

対処方法

ありません。

2.2.60 timetravel拡張の廃止

非互換

FUJITSU Enterprise Postgres 12では、timetravel拡張が廃止されました。

FUJITSU Enterprise Postgres 11以前の場合

timetravel拡張が使用できます。

FUJITSU Enterprise Postgres 12の場合

timetravel拡張が使用できません。

対処方法

ありません。

2.2.61 abstime、reltime、tinterval の各データ型の廃止

非互換

FUJITSU Enterprise Postgres 12では、abstime、reltime、tintervalのデータ型は、timestampのようなSQL標準のデータ型により代替されます。

FUJITSU Enterprise Postgres 11以前の場合

abstime、reltime、tintervalの各データ型が使用できます。

FUJITSU Enterprise Postgres 12の場合

abstime、reltime、tintervalの各データ型が使用できません。

対処方法

timestampのようなSQL標準のデータ型を使用してください。

2.2.62 oid列の特別な振る舞いの廃止

非互換

これまでテーブル作成時にWITH OIDSで隠しoid列を持たせる指定ができましたが、FUJITSU Enterprise Postgres 12ではこの機能が廃止されました。なお、引き続き、テーブルに明示的にoid型の列を含めることは可能です。

隠しoid列を持っていたシステムカタログは、通常のoid列を持つようになります。その結果「SELECT＊…」で、これまで明示的に選択しないと出力されなかったoid列が出力されるようになります。

FUJITSU Enterprise Postgres 11以前の場合

テーブルを作成するときに「WITH OIDS」が指定できます。

[例]

```
=# CREATE TABLE tb1(id int) WITH OIDS;
CREATE TABLE
```

oid列を持っていたシステムカタログは、oid列が出力されません。

[例]

```
=# SELECT * FROM pg_tablespace;
  spcname   | spcowner | spcacl | spcoptions
------------+----------+--------+------------
 pg_default |       10 |        |
 pg_global  |       10 |        |
```

FUJITSU Enterprise Postgres 12の場合

テーブルを作成するときに「WITH OIDS」が指定できません。

[例]

```
=# CREATE TABLE tb1(id int) WITH OIDS;
ERROR:  syntax error at or near "OIDS"
LINE 1: CREATE TABLE tb1(id int) WITH OIDS;
                                      ^
```

oid列を持っていたシステムカタログは、oid列が出力されます。

```
[例]
=# SELECT * FROM pg_tablespace;
 oid  |  spcname   | spcowner | spcacl | spcoptions
------+------------+----------+--------+------------
 1663 | pg_default |       10 |        |
 1664 | pg_global  |       10 |        |
```

対処方法

ありません。

2.2.63 Python言語のサポートの変更

非互換

FUJITSU Enterprise Postgres 12では、Python言語のサポートを以下のように変更します。

サポート言語	FUJITSU Enterprise Postgres 11以前	FUJITSU Enterprise Postgres 12
Python 2言語	○	×
Python 3言語	○	○

対処方法

ありません。

2.2.64 power関数のPOSIX準拠

非互換

FUJITSU Enterprise Postgres 11では、power関数において、NaNをPOSIX標準どおりに扱い、NaN^0 = 1、1^NaN = 1とするように変更します。

FUJITSU Enterprise Postgres 10以前の場合

power(numeric, numeric)のいずれかの引数がNaNである場合、NaNを返します。

[例]

```
=# SELECT power(1.0,'NaN');
   NaN

=# SELECT power('NaN',0.0);
   NaN
```

FUJITSU Enterprise Postgres 11の場合

POSIX標準どおりの結果を返します。

[例]

```
=# SELECT power(1.0,'NaN');
 power
-------
1
(1 行)

=# SELECT power('NaN',0.0);
 power
-------
1
(1 行)
```

対処方法

ありません。

2.2.65 to_number関数のテンプレート区切り文字が一致しない場合の動作変更

非互換

FUJITSU Enterprise Postgres 11では、テンプレート区切り文字が一致しない場合に、to_number関数が文字を消費しないように変更します。また、数値書式用のテンプレートパターンのL(通貨記号)とTH(序数接尾辞)は、数値、正負符号、小数点またはカンマではない文字のみを消費するように変更します。

FUJITSU Enterprise Postgres 10以前の場合

テンプレート区切り文字が一致しない場合でも文字が消費されます。

[例]

```
=# SELECT to_number('1234', 'L9,999');
 to_number
-----------
         4
(1 行)
```

FUJITSU Enterprise Postgres 11の場合

テンプレート区切り文字が一致しない場合は文字が消費されません。

[例]

```
=# SELECT to_number('1234', 'L9,999');
 to_number
-----------
      1234
(1 行)
```

対処方法

ありません。

2.2.66 to_date、to_number、to_timestamp関数でマルチバイトテンプレート文字列を指定した場合の動作変更

非互換

FUJITSU Enterprise Postgres 11では、to_date、to_number、to_timestamp関数において、マルチバイトテンプレート文字列を指定された場合、文字単位で読み飛ばすように変更します。

FUJITSU Enterprise Postgres 10以前の場合

テンプレート文字のバイト数分読み飛ばします。

[例]

```
=# SELECT to_number('$12.34', '￥99D99');
 to_number
-----------
      0.34
(1 行)
```

FUJITSU Enterprise Postgres 11の場合

テンプレート文字を文字単位で読み飛ばします。

[例]

```
=# SELECT to_number('$12.34', '￥99D99');
 to_number
-----------
     12.34
(1 行)
```

対処方法

ありません。

2.2.67 to_char、to_number、to_timestamp関数のダブルクォート内でバックスラッシュを指定した場合の動作変更

非互換

FUJITSU Enterprise Postgres 11では、to_char、to_number、to_timestamp関数において、ダブルクォート内でバックスラッシュが指定された場合、バックスラッシュの後のダブルクォート、またはバックスラッシュのみをエスケープするように変更します。

FUJITSU Enterprise Postgres 10以前の場合

ダブルクォート内でバックスラッシュを指定し、後に続く文字列がダブルクォート、またはバックスラッシュの場合、バックスラッシュの後のすべての文字列がエスケープされます。

[例]

```
=# SELECT to_char(100, 'h"oge¥¥"999');
 to_char
----------
 hoge¥"999
(1 行)
```

FUJITSU Enterprise Postgres 11の場合

ダブルクォート内でバックスラッシュを指定し、後に続く文字列がダブルクォート、またはバックスラッシュの場合、ダブルクォート、またはバックスラッシュのみがエスケープされます。

[例]

```
=# SELECT to_char(100, 'h"oge¥¥"999');
 to_char
----------
 hoge¥ 100
(1 行)
```

対処方法

ありません。

2.2.68 XML関数における相対パスの扱いの変更

非互換

FUJITSU Enterprise Postgres 11では、xpath、xmltableおよびその他のXML関数において、相対パスの開始位置を、入力としたドキュメントのルートノードからドキュメントノードに変更します。

FUJITSU Enterprise Postgres 10以前の場合

相対パスを入力としたドキュメントのルートノードから開始します。

[例]

```
=# SELECT xpath('xml1/text()', '<xml|root><xml|1>xml|1 text</xml|1></xml|root>');
     xpath
---------------
 {"xml|1 text"}
(1 行)

=# SELECT xpath('xml|root/xml|1/text()', '<xml|root><xml|1>xml|1 text</xml|1></xml|root>');
 xpath
-------
 {}
(1 行)
```

FUJITSU Enterprise Postgres 11の場合

相対パスを入力としたドキュメントのドキュメントノードから開始します。

[例]

```
=# SELECT xpath('xml1/text()', '<xml|root><xml|1>xml|1 text</xml|1></xml|root>');
 xpath
```

```
-------
 {}
(1 行)

=# SELECT xpath('xml|root/xml1/text()', '<xml|root><xml1>xml1 text</xml1></xml|root>');
     xpath
---------------
 {"xml1 text"}
(1 行)
```

対処方法

ありません。

2.2.69 拡張問い合わせプロトコルにおけるstatement_timeoutの適用範囲の変更

非互換

FUJITSU Enterprise Postgres 11では、拡張問い合わせプロトコルでstatement_timeoutの設定がクエリ毎に適用されるように変更します。

FUJITSU Enterprise Postgres 10以前の場合

statement_timeoutの設定をSync前のすべてのクエリに適用します。

FUJITSU Enterprise Postgres 11の場合

statement_timeoutの設定をクエリ毎に適用します。

対処方法

ありません。

2.2.70 ラージオブジェクトの権限チェックタイミングの変更

非互換

FUJITSU Enterprise Postgres 11では、ラージオブジェクトの権限チェックをオープン時に行うように変更します。

FUJITSU Enterprise Postgres 10以前の場合

ラージオブジェクトの読み込み、または書き込み時に権限チェックが行われます。

FUJITSU Enterprise Postgres 11の場合

ラージオブジェクトのオープン時に権限チェックが行われます。

対処方法

ありません。

2.2.71 adminpackのpg_file_read、pg_file_length、pg_logfile_rotate関数の削除

非互換

FUJITSU Enterprise Postgres 11では、adminpackのpg_file_read、pg_file_length、pg_logfile_rotate関数を削除します。今後はシステム管理関数を利用してください。

FUJITSU Enterprise Postgres 10以前の場合

- テキストファイルの内容を取得する場合
  adminpackのpg_file_read関数を使用します。
- ファイルサイズを取得する場合
  adminpackのpg_file_length関数を使用します。

- サーバログファイルを循環させる場合
  adminpackのpg_logfile_rotate関数を使用します。

FUJITSU Enterprise Postgres 11の場合

- テキストファイルの内容を取得する場合
  システム管理関数のpg_read_file関数を使用します。
- ファイルサイズを取得する場合
  システム管理関数のpg_stat_file関数で得られるsize列を参照します。
- サーバログファイルを循環させる場合
  システム管理関数のpg_rotate_logfile関数を使用します。
  復帰値はintegerで0または1を返します。

対処方法

ありません。

2.2.72 拡張機能におけるGUC_LIST_QUOTE指定のカスタムパラメータ定義時の動作の変更

非互換

FUJITSU Enterprise Postgres 11では、拡張機能において、GUC_LIST_QUOTEを指定したカスタムパラメータが定義されていた場合、拡張機能読み込み時にエラーとなるように変更します。

対処方法

ありません。

2.2.73 red-blackツリーサポート関数の名前の変更

非互換

FUJITSU Enterprise Postgres 11では、red-blackツリーサポート関数の名前をrb接頭辞からrbt接頭辞に変更します。

対処方法

ありません。

2.2.74 chkpass拡張の削除

非互換

FUJITSU Enterprise Postgres 11では、chkpass拡張を削除します。

対処方法

pgcrypto拡張を使用してください。

2.2.75 ダブルクォートで括ったコマンドオプションの小文字への自動変換の廃止

非互換

FUJITSU Enterprise Postgres11では、ダブルクォートで括られているSQLコマンドのオプション名がシステム内部で小文字に強制変換されないように変更します。

今後は上記の変換処理がなくなり、エラーとなります。

FUJITSU Enterprise Postgres 10以前の場合

オプション名に大文字を含めて指定した場合、正常終了します。

[例]

```
=# ALTER TABLE t1 SET ("FillFactor" = 50);
ALTER TABLE
```

FUJITSU Enterprise Postgres 11の場合

オプション名に大文字を含めて指定した場合、エラーとなります。

[例]

```
=# ALTER TABLE t1 SET ("FillFactor" = 50);
ERROR:  認識できないラメータ "FillFactor"
```

対処方法

オプション名をダブルクォートで括っているロジックについて、ダブルクォートを削除するか、小文字に統一してください。

2.2.76 psqlコマンドの¥d+の出力情報の追加

非互換

FUJITSU Enterprise Postgres 11では、psqlコマンドの¥d+をパーティションテーブルに対して実行した場合、子テーブルのパーティションに関する情報も表示するよう変更します。

FUJITSU Enterprise Postgres 10以前の場合

子テーブルのパーティション情報は表示されません。

[例]

```
=# ¥d+ t1
                                テーブル "schema.t1"
  列  |  型  | 照合順序 | Null 値を許容 | デフォルト | ストレージ | 統計の対象 | 説明
------+------+----------+---------------+------------+------------+------------+------
 c1   | text |          |               |            | extended   |            |
 c2   | text |          |               |            | extended   |            |
 c3   | text |          |               |            | extended   |            |
パーティションキー: LIST (c1)
パーティション: t2 FOR VALUES IN ('aaa'),
                t3 FOR VALUES IN ('bbb'),
                t4 FOR VALUES IN ('ccc')
```

FUJITSU Enterprise Postgres 11の場合

子テーブルのパーティション情報が表示されます。

[例]

```
=# ¥d+ t1
                                テーブル "schema.t1"
  列  |  型  | 照合順序 | Null 値を許容 | デフォルト | ストレージ | 統計の対象 | 説明
------+------+----------+---------------+------------+------------+------------+------
 c1   | text |          |               |            | extended   |            |
 c2   | text |          |               |            | extended   |            |
 c3   | text |          |               |            | extended   |            |
パーティションキー: LIST (c1)
パーティション: t2 FOR VALUES IN ('aaa'), PARTITIONED,
                t3 FOR VALUES IN ('bbb'),
                t4 FOR VALUES IN ('ccc')
```

対処方法

ありません。

2.2.77 pg_basebackupの動作の変更

非互換

FUJITSU Enterprise Postgres 11では、pg_basebackup のベースバックアップから、UNLOGGEDテーブル、一時テーブル、pg_internal.init ファイルを除外するように変更します。

対処方法

ありません。

2.2.78 いくつかの関数で変動性区分を変更

非互換

FUJITSU Enterprise Postgres 11では、以下の関数の変動性区分をVOLATILEに変更します。

- query_to_xml
- cursor_to_xml
- cursor_to_xmlschema
- query_to_xmlschema
- query_to_xml_and_xmlschema

対処方法

必要に応じてALTER FUNCTION文で変動性区分を変更してください。

ただし、変動制区分を変えることで最適な実行計画とならない場合があるため、非推奨です。

2.2.79 いくつかの関数をPARALLEL UNSAFEに変更

非互換

FUJITSU Enterprise Postgres 11では、以下の関数をPARALLEL UNSAFEに変更します。

- brin_summarize_new_values
- brin_summarize_range
- brin_desummarize_range
- gin_clean_pending_list
- cursor_to_xml
- cursor_to_xmlschema
- ts_rewrite
- ts_stat
- binary_upgrade_create_empty_extension
- pg_import_system_collations

対処方法

必要に応じてALTER FUNCTION文で並列モードを変更してください。

ただし、並列モードを変えることで最適な実行計画とならない場合があるため、非推奨です。

2.2.80 アプリケーションの接続先切り替え機能の指定方法の変更

非互換

アプリケーションの接続先切り替え機能で指定する“ターゲットサーバ”を、OSS PostgreSQLと同じ名称に変更します。

対処方法

アプリケーションの接続先切り替え機能としてターゲットサーバを指定しているアプリケーションは、以下のようにターゲットサーバと指定値を変更して再コンパイル後、FUJITSU Enterprise Postgres 10以降のクライアントを利用して実行してください。

ターゲットサーバの指定名称

利用対象	修正前	修正後
JDBCドライバを利用	変更なし	
ODBCドライバを利用	TargetServer	target_session_attrs
.NET Data Providerを利用	target_server	TargetServerType
接続サービスファイルを利用	target_server	target_session_attrs
C言語用ライブラリ(libpq)を利用	target_server 環境変数 (PGXTARGETSERVER)	target_session_attrs 環境変数 (PGTARGETSESSIONATTRS)
埋め込みSQLを利用	target_server	target_session_attrs
psqlを利用	環境変数 (PGXTARGETSERVER)	環境変数 (PGTARGETSESSIONATTRS)

ターゲットサーバへの指定値

サーバの選択順	JDBCドライバ	.NET Data Provider		他のドライバ	
		修正前	修正後	修正前	修正後
プライマリサーバ	変更なし	primary	master	primary	read-write
スタンバイサーバ	変更なし	－	slave	－	－
スタンバイサーバの優先	変更なし	prefer_standby	preferSlave	prefer_standby	prefer-read
任意	変更なし	－	any	－	any

2.2.81 C言語およびCOBOL言語による埋め込みSQLアプリケーションのOSS PostgreSQL互換

非互換

FUJITSU Enterprise postgres 9.6以前のバージョンでコンパイルしたC言語およびCOBOL言語による埋め込みSQLアプリケーションを、FUJITSU Enterprise Postgres 10のクライアントで利用することができません。

対処方法

FUJITSU Enterprise Postgres 10以降のクライアントでアプリケーションの再コンパイルのみを実施後、FUJITSU Enterprise Postgres 10以降のクライアントを利用して実行してください。

2.2.82 Vertical Clustered Index(VCI)の選択条件変更

非互換

VCIが利用されない関数に以下の関数が追加されます。

分類	関数
日付/時刻関数	age(timestamp) current_date current_timestamp current_time localtime localtimestamp
セション情報関数	current_user current_role

対処方法

VCIを利用する場合には、以下のように、age関数は第2引数を指定し、その他の関数は、サブクエリとして指定してください。

[例]

元の指定: select age(列A), current_date from テーブル

変更後: select age(列A, **now()**), **(select current_date)** from テーブル

2.2.83 システム管理関数の名称変更

非互換

FUJITSU Enterprise Postgres 10では、以下のとおり、システム管理関数の名称を変更します。

FUJITSU Enterprise Postgres 9.6以前	FUJITSU Enterprise Postgres 10
pg_current_xlog_flush_location	pg_current_wal_flush_lsn
pg_current_xlog_insert_location	pg_current_wal_insert_lsn
pg_current_xlog_location	pg_current_wal_lsn
pg_switch_xlog	pg_switch_wal
pg_xlogfile_name	pg_walfile_name
pg_xlogfile_name_offset	pg_walfile_name_offset
pg_xlog_location_diff	pg_wal_lsn_diff
pg_last_xlog_receive_location	pg_last_wal_receive_lsn
pg_last_xlog_replay_location	pg_last_wal_replay_lsn
pg_is_xlog_replay_paused	pg_is_wal_replay_paused
pg_xlog_replay_pause	pg_wal_replay_pause
pg_xlog_replay_resume	pg_wal_replay_resume

対処方法

ありません。

2.2.84 システム情報関数の返却項目の名称変更

非互換

FUJITSU Enterprise Postgres 10では、以下のとおり、システム情報関数の返却項目の名称を変更します。

システム情報関数	FUJITSU Enterprise Postgres 9.6以前	FUJITSU Enterprise Postgres 10
pg_control_checkpoint	checkpoint_location	checkpoint_lsn
	prior_location	prior_lsn
	redo_location	redo_lsn
pg_control_recovery	min_recovery_end_location	min_recovery_end_lsn
	backup_start_location	backup_start_lsn
	backup_end_location	backup_end_lsn

対処方法

ありません。

2.2.85 FROM句にシーケンスを指定した場合の返却列の変更

非互換

FUJITSU Enterprise Postgres 10では、FROM句にシーケンスを指定した場合の返却列を変更し、nextvalで更新されるlast_value、log_cnt、is_calledのみを返却します。シーケンス名、開始値、増分量などの情報はpg_sequenceカタログに格納され、pg_sequenceカタログ、またはpg_sequencesシステムビューを通じて参照することができます。

FUJITSU Enterprise Postgres 9.6以前の場合

nextvalで更新されないシーケンス名、開始値なども返却します。

[例]

```
=# SELECT * FROM seq1;
sequence_name | last_value | start_value | increment_by |      max_value      | min_value | cache_value | log_cnt |
is_cycled | is_called
--------------+------------+-------------+--------------+---------------------+-----------+-------------+-----------
+-----------+-----------
 seq1         |          1 |           1 |            1 | 9223372036854775807 |         1 | 1           |       0 | f         | f
(1 行)
```

FUJITSU Enterprise Postgres 10以降の場合

nextvalで更新されるlast_value、log_cnt、is_calledのみ返却します。

[例]

```
=# SELECT * FROM seq1;
 last_value | log_cnt | is_called
------------+---------+-----------
          1 |       0 | f
(1 行)
```

対処方法

nextvalで更新されないシーケンス名、開始値、増分量などの情報はpg_sequenceカタログ、またはpg_sequencesシステムビューを参照してください。以下にFUJITSU Enterprise Postgres 9.6との対応を示します。

FUJITSU Enterprise Postgres 9.6以前	FUJITSU Enterprise Postgres 10以降		
	FROM句に指定	pg_sequenceカタログ	pg_sequencesシステムビュー
sequence_name		sequencename	
last_value	last_value	last_value	
start_value		start_value	seqstart
increment_by		increment_by	seqincrement
max_value		max_value	seqmax
min_value		min_value	seqmin
cache_value		cache_size	seqcache
log_cnt	log_cnt		
is_cycled		cycle	seqcycle
is_called	is_called		

2.2.86 範囲外の値を入力した場合のto_date関数とto_timestamp関数の動作変更

非互換

FUJITSU Enterprise Postgres 10では、to_date関数とto_timestamp関数において、範囲外の値を入力した場合はエラーを返すように変更します。

FUJITSU Enterprise Postgres 9.6以前の場合

整形されたデータを返します。

[例]

```
=# SELECT to_date('2000-4-50', 'YYYY-MM-DD');
  to_date
------------
2000-05-20
(1 行)
```

FUJITSU Enterprise Postgres 10以降の場合

エラーを返します。

[例]

```
=# SELECT to_date('2000-4-50', 'YYYY-MM-DD');
ERROR:  日付時刻のフィールドが範囲外です: "2000-4-50" (11521)
```

対処方法

ありません。

2.2.87 選択リストに集合を返す関数を指定した場合の動作変更

非互換

FUJITSU Enterprise Postgres 10では、選択リストに集合を返す関数を複数指定し、これらが異なる行数を返す場合、行数が少ない結果はNULLを加えて最も長い結果に一致するよう変更します。また、集合を返す関数をCASE式やCOALESCE式に指定した場合はエラーにするよう変更します。

FUJITSU Enterprise Postgres 9.6以前の場合

集合を返す関数を同時に終了するまで実行し、関数の最小公倍数の周期に等しい行数を返します。また、CASE式やCOALESCE式にも指定できます。

[例]

```
=# SELECT generate_series(1,2), generate_series(1,3);
generate_series  | generate_series
-----------------+-----------------
               1 |               1
               2 |               2
               1 |               3
               2 |               1
               1 |               2
               2 |               3
(6 行)


=# select case when true then generate_series(1,2) else null end;
case
------
    1
    2
(2 行)
```

FUJITSU Enterprise Postgres 10以降の場合

行数が少ない場合はNULLを加えた結果を返します。また、集合を返す関数はCASE式やCOALESCE式には指定できません。

[例]

```
=# SELECT generate_series(1,2), generate_series(1,3);
generate_series  | generate_series
-----------------+-----------------
               1 |               1
               2 |               2
                 |               3

(3 行)


=# select case when true then generate_series(1,2) else null end;

ERROR:  set-returning functions are not allowed in CASE

行 1: select case when true then generate_series(1,2) else null en...

                                  ^

HINT:  You might be able to move the set-returning function into a LATERAL FROM item.
```

対処方法

ありません。

2.2.88 同一テーブルが複数回更新される場合の文単位のトリガの動作変更

非互換

FUJITSU Enterprise Postgres 10では、以下のように、1つのSQL文で同一テーブルが複数回更新される場合は、文単位のトリガを1回だけ起動するよう変更します。

- 共通テーブル式で同一テーブルを複数回更新する場合
- 外部キー制約の被参照列が更新され、それに伴い、外部キー制約を付加したテーブルが更新される場合

FUJITSU Enterprise Postgres 9.6以前の場合

1つのSQL文で文単位のトリガが複数回起動されることがあります。

[例]

```
=# CREATE FUNCTION func1() RETURNS TRIGGER LANGUAGE plpgsql AS $$
$#   BEGIN
$#     RAISE NOTICE 'Trigger % called.', TG_NAME;
$#     RETURN NULL;
$#   END;
$# $$;
CREATE FUNCTION
=# CREATE TABLE tbl1(c1 int);
CREATE TABLE
=# CREATE TRIGGER trg1 AFTER INSERT ON tbl1 FOR EACH STATEMENT EXECUTE PROCEDURE func1();
CREATE TRIGGER
=# WITH wcte AS (INSERT INTO tbl1 VALUES(1)) INSERT INTO tbl1 VALUES(2);
NOTICE:  Trigger trg1 called.
NOTICE:  Trigger trg1 called.
INSERT 0 1
```

FUJITSU Enterprise Postgres 10以降の場合

1つのSQL文で文単位のトリガは1回しか起動されません。

[例]

```
=# CREATE FUNCTION func1() RETURNS TRIGGER LANGUAGE plpgsql AS $$
$#   BEGIN
$#     RAISE NOTICE 'Trigger % called.', TG_NAME;
$#     RETURN NULL;
$#   END;
$# $$;
CREATE FUNCTION
=# CREATE TABLE tbl1(c1 int);
CREATE TABLE
=# CREATE TRIGGER trg1 AFTER INSERT ON tbl1 FOR EACH STATEMENT EXECUTE PROCEDURE func1();
CREATE TRIGGER
=# WITH wcte AS (INSERT INTO tbl1 VALUES(1)) INSERT INTO tbl1 VALUES(2);
NOTICE:  Trigger trg1 called.
INSERT 0 1
```

対処方法

ありません。

2.2.89 C言語関数におけるVersion 0呼び出し規約の削除

非互換

FUJITSU Enterprise Postgres 10では、C言語関数において、Version 0の呼び出し規約を削除します。

対処方法

C言語関数をVersion 0呼び出し規約で作成している場合は、Version 1の呼び出し規約に沿うように修正してください。

2.2.90 Oracleデータベース互換機能のrpad関数にマルチバイト文字を入力した場合の返却結果の変更

非互換

FUJITSU Enterprise Postgres 10では、Oracle互換機能のrpad関数において、マルチバイト文字の泣き別れにより第2引数に指定した結果長に満たない場合は、文字列の最後に空白を埋め込むよう変更します。

FUJITSU Enterprise Postgres 9.6以前の場合

マルチバイト文字の泣き別れにより第2引数に指定した結果長に満たない場合は、文字列の先頭に空白を埋め込みます。

[例]

```
=# SELECT '[' || oracle.rpad('あいう', 10, '+えお') || ']';
   ?column?
--------------
[ あいう+え]
(1 行)
```

FUJITSU Enterprise Postgres 10以降の場合

マルチバイト文字の泣き別れにより第2引数に指定した結果長に満たない場合は、文字列の最後に空白を埋め込みます。

[例]

```
=# SELECT '[' || oracle.rpad('あいう', 10, '+えお') || ']';
   ?column?
--------------
［あいう+え ］
(1 行)
```

対処方法

ありません。

2.2.91 money型をbigint型で除算する場合の演算方法の変更

非互換

FUJITSU Enterprise Postgres 10では、money型をbigint型で除算する場合の演算方法を変更し、小数点以下を切り捨てるようにします。

FUJITSU Enterprise Postgres 9.6以前の場合

小数点以下は四捨五入します。

[例]

```
=# SELECT 11::money / 2::bigint;
 ?column?
----------
```

```
      ￥6
(1 行)
```

FUJITSU Enterprise Postgres 10以降の場合

小数点以下は切り捨てます。

[例]

```
=# SELECT 11::money / 2::bigint;
 ?column?
----------
      ￥5
(1 行)
```

対処方法

小数点以下を四捨五入したい場合は、整数以外の数値データ型で除算してください。

2.2.92 cubeモジュールの~>演算子の振る舞いの変更

非互換

FUJITSU Enterprise Postgres 10では、cubeモジュールの~>演算子の振る舞いを変更し、左下－右上の座標において、1次元から順に並び替えるという正規化をしたn次座標を取得します。

FUJITSU Enterprise Postgres 9.6以前の場合

座標を左下－右上の順に並び替えるという正規化をしたn次座標を取得します。

[例]

```
=# SELECT cb, cb~>1, cb~>2, cb~>3, cb~>4 FROM cube(ARRAY[1,2], ARRAY[3,4]) AS cb;
      cb       | ?column? | ?column? | ?column? | ?column?
---------------+----------+----------+----------+----------
 (1, 2),(3, 4) |        1 |        2 |        3 |        4
(1 行)
```

FUJITSU Enterprise Postgres 10以降の場合

左下－右上の座標において、1次元から順に並び替えるという正規化をしたn次座標を取得します。

[例]

```
=# SELECT cb, cb~>1, cb~>2, cb~>3, cb~>4 FROM cube(ARRAY[1,2], ARRAY[3,4]) AS cb;
      cb       | ?column? | ?column? | ?column? | ?column?
---------------+----------+----------+----------+----------
 (1, 2),(3, 4) |        1 |        3 |        2 |        4
(1 行)
```

対処方法

ありません。

2.2.93 tsearch2モジュールの削除

非互換

FUJITSU Enterprise Postgres 10では、PostgreSQL 8.3以前のテキスト検索と互換性のあるtsearch2モジュールを削除します。

対処方法

ありません。

2.2.94 psqlコマンドの特別変数に不当な値を指定した場合の動作変更

非互換

FUJITSU Enterprise Postgres 10では、psqlコマンドの特別変数に不当な値を指定した場合にエラーにするよう変更します。

FUJITSU Enterprise Postgres 9.6以前の場合

特別変数に不当な値を指定した場合、デフォルト値が設定されます。

[例]

```
=# ¥set AUTOCOMMIT abc

"AUTOCOMMIT" の不明な値 "abc"。"on"と仮定します (16201)
```

FUJITSU Enterprise Postgres 10以降の場合

特別変数に不当な値を指定した場合はエラーになります。

[例]

```
=# ¥set AUTOCOMMIT abc

"AUTOCOMMIT" の値 "abc" が認識できません：真偽値を指定してください。
```

対処方法

ありません。

2.2.95 psqlコマンドの¥dおよび¥dDの出力形式の変更

非互換

FUJITSU Enterprise Postgres 10では、psqlコマンドの¥dおよび¥dDの出力形式を変更し、照合順序、NOT NULL制約、およびデフォルト値を個々の列で出力するようにします。

FUJITSU Enterprise Postgres 9.6以前の場合

照合順序、NOT NULL制約、およびデフォルト値を1つの列で出力します。

[例]

```
=# ¥d t1
                    テーブル "public.t1"
 列 |            型             |           修飾語
----+---------------------------+-----------------------------
 c1 | text                      | 照合順序 ja_JP.utf8 not null
 c2 | timestamp without time zone | default now()
 c3 | integer                   | not null default 1
```

FUJITSU Enterprise Postgres 10以降の場合

照合順序、NOT NULL制約、およびデフォルト値を個々の列で出力します。

[例]

```
=# ¥d t1
                              テーブル "public.t1"
 列 |            型             |  照合順序  | Null 値を許容 |    デフォルト
----+---------------------------+------------+---------------+-------------------
 c1 | text                      | ja_JP.utf8 | not null      |
 c2 | timestamp without time zone |            |               | CURRENT_TIMESTAMP
 c3 | integer                   |            | not null      | 1
```

対処方法

ありません。

2.2.96 PL/Tclの自動ロード機能の削除

非互換

FUJITSU Enterprise Postgres 10では、PL/Tclの自動ロード機能を削除します。これに伴い、自動ロード機能で使用するテーブル管理用のサポートスクリプトpltcl_loadmod、pltcl_listmod、pltcl_delmod、およびunknownモジュールのソースunknown.pltclも削除します。

対処方法

Tclセッションで初期化コードを実行する場合は、postgresql.confファイルのpltcl.start_procパラメータおよびpltclu.start_procパラメータを使用してください。

2.2.97 PL/Pythonにおける複合型配列の扱いの変更

非互換

FUJITSU Enterprise Postgres 10では、PL/Pythonで複合型を多次元配列内で使用し、かつ、リストで指定した場合はエラーにするよう変更します。

FUJITSU Enterprise Postgres 9.6以前の場合

複合型をリストで指定することができます。

[例]

```
=# CREATE TYPE mytype AS (c1 int, c2 int);
CREATE TYPE
=# CREATE FUNCTION list() RETURNS SETOF mytype[] AS $$ return [[1, 1], [1, 2]], [[2, 1], [2, 2]] $$ LANGUAGE plpythonu;
CREATE FUNCTION
=# SELECT list();
       list
-------------------
 {"(1,1)","(1,2)"}
 {"(2,1)","(2,2)"}
(2 行)
```

FUJITSU Enterprise Postgres 10以降の場合

複合型をリストで指定した場合はエラーにします。

[例]

```
=# CREATE TYPE mytype AS (c1 int, c2 int);
CREATE TYPE
=# CREATE FUNCTION list() RETURNS SETOF mytype[] AS $$ return [[1, 1], [1, 2]], [[2, 1], [2, 2]] $$ LANGUAGE plpythonu;
CREATE FUNCTION
=# SELECT list();
ERROR:  属性 "c1" が Python オブジェクト中に存在しません (12648)
HINT:  複合型を配列に入れて返したい場合、 "[('foo',)]" のように複合型を Pythonのタプルとして返すようにしてください。
CONTEXT:  戻り値を生成する際に (14845)
PL/Python 関数 "list" (14842)
```

対処方法

複合型を多次元配列内で使用する場合はリストではなく、タプルで指定するように変更してください。

2.2.98 DHパラメータのサイズ変更

非互換

FUJITSU Enterprise Postgres 10では、OpenSSLの短期のDH暗号に使用するデフォルトのDHパラメータのサイズを2048ビットに変更します。本変更により、JDK 6またはJRE 6を使用したアプリケーションからのSSL接続がエラーになります。

対処方法

512ビット以上1024ビット以下、かつ64の倍数のサイズでDHパラメータファイルを作成し、postgresql.confファイルのssl_dh_params_fileパラメータに指定してください。

2.2.99 to_char関数におけるマイナス符号の扱いの変更

非互換

FUJITSU Enterprise Postgres 9.6以降では、to_char関数の時間関連のフィールドにおいて、マイナス符号をフィールド幅の一部としてカウントしないように変更します。

FUJITSU Enterprise Postgres 9.5 の場合

マイナス符号をフィールド幅の一部としてカウントします。

[例]

```
=# SELECT to_char('-4 years'::interval, 'YY');
 to_char
---------
 -4
(1 row)
```

FUJITSU Enterprise Postgres 9.6以降の場合

マイナス符号はフィールド幅の一部としてカウントしません。

[例]

```
=# SELECT to_char('-4 years'::interval, 'YY');
 to_char
---------
 -04
(1 row)
```

対処方法

ありません。

2.2.100 無限大のタイムスタンプを入力した場合のextract関数の戻り値の変更

非互換

FUJITSU Enterprise Postgres 9.6以降では、extract関数において、無限大のタイムスタンプを入力した場合の戻り値を変更します。

FUJITSU Enterprise Postgres 9.5 の場合

抽出するフィールドの種別にかかわらず、0を返却します。

[例]

```
=# SELECT extract(year from TIMESTAMP 'infinity');
 date_part
-----------
         0
(1 row)


=# SELECT extract(month from TIMESTAMP 'infinity');
 date_part
-----------
         0
(1 row)
```

FUJITSU Enterprise Postgres 9.6以降の場合

抽出するフィールドが単調増加するもの(year、century、decade、epoch、isoyear、millennium)であれば無限大を返却し、そうでないものはNULLを返却します。

[例]

```
=# SELECT extract(year from TIMESTAMP 'infinity');
 date_part
-----------
  Infinity
(1 row)


=# ¥pset null NULL
=# SELECT extract(month from TIMESTAMP 'infinity');
 date_part
-----------
      NULL
(1 row)
```

対処方法

ありません。

2.2.101 三角関数のPOSIX準拠

非互換

FUJITSU Enterprise Postgres 9.6以降では、三角関数において、NaNおよびinfinityをPOSIX標準どおりに扱い、NaNを入力した場合はNaNを返し、infinityを入力した場合はエラーにするよう変更します。

FUJITSU Enterprise Postgres 9.5 の場合

NaN、またはinfinityを入力した場合の振る舞いはプラットフォームによって様々であり、POSIX標準とは異なる結果を返す場合があります。

[例]

```
=# SELECT cos('infinity');
 cos
-----
 NaN
(1 row)
```

FUJITSU Enterprise Postgres 9.6以降の場合

POSIX標準どおりの結果を返します。

[例]

```
=# SELECT cos('infinity');
ERROR:  input is out of range
```

対処方法

ありません。

2.2.102 行集合を返す関数がNULLを返せるように変更

非互換

FUJITSU Enterprise Postgres 9.6以降では、“SELECT FROM 関数名”の形式で関数を実行する場合、行集合を返す関数がNULLを返せるように変更します。

FUJITSU Enterprise Postgres 9.5 の場合

NULLを返すことはできません。

[例]

```
=# CREATE TYPE typ1 AS (a int, b text);

CREATE TYPE


=# SELECT * FROM unnest(array[(1,'foo')::typ1, null::typ1]);

ERROR:  function returning set of rows cannot return null value
```

FUJITSU Enterprise Postgres 9.6以降の場合

NULLを返すことができます。

[例]

```
=# CREATE TYPE typ1 AS (a int, b text);

CREATE TYPE


=# ¥pset null NULL

=# SELECT * FROM unnest(array[(1,'foo')::typ1, null::typ1]);

  a   |  b
------+------
    1 | foo
 NULL | NULL
(2 rows)
```

対処方法

ありません。

2.2.103 PL/pgSQLのRAISEコマンドにおけるCONTEXTメッセージを出力するように変更

非互換

FUJITSU Enterprise Postgres 9.6以降では、PL/pgSQLのRAISEコマンドでCONTEXTメッセージを出力するように変更します。

FUJITSU Enterprise Postgres 9.5 の場合

CONTEXTメッセージは出力しません。

[例]

```
=# CREATE FUNCTION func1() RETURNS void AS $$

$# BEGIN

$#     RAISE 'dummy exception';

$#     RETURN;

$# END;

$# $$ LANGUAGE plpgsql;

CREATE FUNCTION
```

```
=# SELECT func1();

ERROR:  dummy exception
```

FUJITSU Enterprise Postgres 9.6以降の場合

CONTEXTメッセージを出力します。

[例]

```
=# CREATE FUNCTION func1() RETURNS void AS $$

$# BEGIN

$#     RAISE 'dummy exception';

$#     RETURN;

$# END;

$# $$ LANGUAGE plpgsql;

CREATE FUNCTION


=# SELECT func1();

ERROR:  dummy exception

CONTEXT:  PL/pgSQL function func1() line 3 at RAISE
```

対処方法

ありません。

2.2.104 テキスト検索パーサにおける数字から始まるメールアドレスやホスト名の解析方法の変更

非互換

FUJITSU Enterprise Postgres 9.6以降では、テキスト検索パーサにおいて、数字から始まるメールアドレスやホスト名を1つのトークンとして扱うように変更します。

FUJITSU Enterprise Postgres 9.5 の場合

数字から始まるメールアドレスやホスト名は数字とそれ以外のトークンに分解します。

[例]

```
=# SELECT to_tsvector('http://123-example.com');

       to_tsvector
--------------------------
 '123':1 'example.com':2
(1 row)
```

FUJITSU Enterprise Postgres 9.6以降の場合

数字から始まるメールアドレスやホスト名は1つのトークンとして扱います。

[例]

```
=# SELECT to_tsvector('http://123-example.com');
```

```
     to_tsvector
----------------------
 '123-example.com':1
(1 row)
```

対処方法

ありません。

2.2.105 関数をORDER BYソートの順序で実行するように変更

非互換

FUJITSU Enterprise Postgres 9.6以降では、SELECT文の出力リスト内の揮発性のある、もしくは高コストの関数をORDER BYソートの順序で実行するように変更します。また、この変更により、LIMIT句を指定している場合は、必要とされる回数以上に関数が実行されなくなります。

FUJITSU Enterprise Postgres 9.5 の場合

関数はORDER BYソートの前に実行します。

[例]

```
=# CREATE SEQUENCE seq1;

CREATE SEQUENCE

=# SELECT c1,c2,nextval('seq1') FROM t1 ORDER BY c1 LIMIT 3;

 c1 | c2 | nextval
----+----+---------
  1 |  1 |       4
  2 |  2 |       6
  3 |  3 |       2
(3 rows)

=# SELECT curval('seq1')

currval
---------
      10
(1 row)
```

FUJITSU Enterprise Postgres 9.6以降の場合

関数はORDER BYソートの順序で実行します。

[例]

```
=# SELECT c1,c2,nextval('seq1') FROM t1 ORDER BY c1 LIMIT 3;

 c1 | c2 | nextval
----+----+---------
  1 |  1 |       1
  2 |  2 |       2
  3 |  3 |       3
(3 rows)

=# SELECT curval('seq1')

currval
---------
       3
(1 row)
```

対処方法

ありません。

2.2.106 日付やタイムスタンプの限界値チェックの強化

非互換

FUJITSU Enterprise Postgres 9.6以降では、日付やタイムスタンプの限界値のチェック処理において、timestamp with time zoneの限界値に極めて近い値を適切に処理するように変更します。

FUJITSU Enterprise Postgres 9.5 の場合

予期せぬ範囲外エラーが発生する場合があります。

[例]

```
=# SELECT '4714-11-23 16:00:00-08 BC'::timestamptz;

ERROR:  timestamp out of range: "4714-11-23 16:00:00-08 BC"
```

FUJITSU Enterprise Postgres 9.6以降の場合

範囲内の値であれば正しく処理します。

[例]

```
=# SELECT '4714-11-23 16:00:00-08 BC'::timestamptz;

           timestamptz
--------------------------------
 4714-11-24 09:18:59+09:18:59 BC
(1 row)
```

対処方法

ありません。

2.2.107 OIDを取得するシステムカタログ情報関数における引数のデータ型の変更

非互換

FUJITSU Enterprise Postgres 9.6以降では、OIDを取得する以下のシステムカタログ情報関数において、引数のデータ型をcstring型からtext型に変更します。

- to_regclass
- to_regproc
- to_regprocedure
- to_regoper
- to_regoperator
- to_regtype
- to_regnamespace
- to_regrole

FUJITSU Enterprise Postgres 9.5 の場合

引数のデータ型がcstring型のため、キャストが必要な場合があります。

[例]

```
=# SELECT to_regclass('t'||'1');
```

```
ERROR:  function to_regclass(text) does not exist

LINE 1: SELECT to_regclass('t'||'1');

               ^

HINT:  No function matches the given name and argument types. You might need to add explicit type casts.


=# SELECT to_regclass(('t'||'1')::cstring);

 to_regclass
-------------
 t1
(1 row)
```

FUJITSU Enterprise Postgres 9.6以降の場合

引数のデータ型がtext型のため、キャストが不要になります。cstring型にキャストする場合はエラーになります。

[例]

```
=# SELECT to_regclass('t'||'1');

 to_regclass
-------------
 t1
(1 row)


=# SELECT to_regclass(('t'||'1')::cstring);

ERROR:  function to_regclass(cstring) does not exist

LINE 1: SELECT to_regclass(('t'||'1')::cstring);

                  ^

HINT:  No function matches the given name and argument types. You might need to add explicit type casts.
```

対処方法

引数をcstring型にキャストしている場合は、キャストを削除してください。

2.2.108 不当な値を入力した場合のシステムカタログ情報関数の動作変更

非互換

FUJITSU Enterprise Postgres 9.6以降では、システムカタログ情報関数において、不当な値を入力した場合はエラーにせず、NULLを返すように変更します。

FUJITSU Enterprise Postgres 9.5 の場合

「cache lookup failed」のエラーになります。

[例]

```
=# SELECT pg_get_indexdef(0);

ERROR:  cache lookup failed for index 0
```

FUJITSU Enterprise Postgres 9.6以降の場合

NULLを返します。

[例]

```
=# ¥pset null NULL

=# SELECT pg_get_indexdef(0);

 pg_get_indexdef
-----------------
NULL
(1 row)
```

対処方法

ありません。

2.2.109 レプリケーションSQL関数pg_replication_origin_xact_resetの引数の削除

非互換

FUJITSU Enterprise Postgres 9.6以降では、レプリケーションSQL関数pg_replication_origin_xact_resetにおいて、使用しない2つの引数を削除します。

FUJITSU Enterprise Postgres 9.5 の場合

pg_replication_origin_xact_reset関数は2つの引数を持ちます。

[例]

```
=# SELECT pg_replication_origin_xact_reset('0/1F419B0', current_timestamp);

 pg_replication_origin_xact_reset
----------------------------------

(1 row)


=# SELECT pg_replication_origin_xact_reset();

ERROR:  function pg_replication_origin_xact_reset() does not exist

LINE 1: SELECT pg_replication_origin_xact_reset();

               ^

HINT:  No function matches the given name and argument types. You might need to add explicit type casts.
```

FUJITSU Enterprise Postgres 9.6以降の場合

pg_replication_origin_xact_reset関数は引数を持ちません。

[例]

```
=# SELECT pg_replication_origin_xact_reset('0/1F419B0', current_timestamp);

ERROR:  function pg_replication_origin_xact_reset(unknown, timestamp with time zone) does not exist

LINE 1: SELECT pg_replication_origin_xact_reset('0/1F419B0', current...

               ^

HINT:  No function matches the given name and argument types. You might need to add explicit type casts.


=# SELECT pg_replication_origin_xact_reset();

 pg_replication_origin_xact_reset
----------------------------------
```

(1 row)

対処方法

pg_replication_origin_xact_reset関数に指定している引数を削除してください。

2.2.110 psqlコマンドの-cオプションを指定した場合に起動用ファイルを読み込むように変更

非互換

FUJITSU Enterprise Postgres 9.6以降では、psqlコマンドの-cオプションを指定した場合、起動用ファイルを読み込むように変更します。

FUJITSU Enterprise Postgres 9.5 の場合

-cオプションが指定された場合、-Xオプション、または--no-psqlrcオプションが暗黙的に指定されたとし、起動用ファイルを読み込みません。

FUJITSU Enterprise Postgres 9.6以降の場合

-cオプションが指定された場合、起動用ファイルを読み込みます。

対処方法

起動用ファイルを読み込ませたくない場合は、-Xオプション、または--no-psqlrcオプションを指定してください。

2.2.111 画面表示モードが自動設定の場合のpsqlコマンドの表示方法の変更

非互換

FUJITSU Enterprise Postgres 9.6以降では、¥pset expanded autoでpsqlコマンドの画面表示モードを自動設定にした場合、問い合わせの出力が1列の場合は拡張モードではなく、通常モードで表示するように変更します。

FUJITSU Enterprise Postgres 9.5 の場合

問い合わせの出力が画面幅より広い場合に拡張モードで表示します。

FUJITSU Enterprise Postgres 9.6以降の場合

問い合わせの出力が1列の場合、出力が画面幅より広くても通常モードで表示します。

対処方法

ありません。

2.2.112 psqlコマンドの¥df+の出力情報の追加

非互換

FUJITSU Enterprise Postgres 9.6以降では、psqlコマンドの¥df+の出力情報として、並列処理での安全性とアクセス権限を新たに表示するよう変更します。

対処方法

ありません。

2.2.113 環境変数PAGERに空文字列または全て空白文字の文字列を指定した場合のpsqlコマンドの表示方法の変更

非互換

FUJITSU Enterprise Postgres 9.6以降では、psqlコマンドにおいて、環境変数PAGERに空文字列または全て空白文字の文字列を指定した場合、ページャの使用を無効にするよう変更します。

FUJITSU Enterprise Postgres 9.5 の場合

コマンドの結果を表示しません。

[例]

```
$ export PAGER=

$ psql postgres -c "SELECT generate_series(1,100)"

$
```

FUJITSU Enterprise Postgres 9.6以降の場合

ページャを使用せずに結果を表示します。

[例]

```
$ export PAGER=

$ psql postgres -c "SELECT generate_series(1,100)"

 generate_series
-----------------
               1
               2
               3
  .  .  .  .  .  .  .  .  .
              98
              99
             100
(100 rows)
```

対処方法

ありません。

2.2.114 pgbenchコマンドの独自スクリプトの書式変更

非互換

FUJITSU Enterprise Postgres 9.6以降では、pgbenchコマンドの独自スクリプトの書式を変更し、1つのSQL文は改行までではなく、セミコロンまでとするようにします。本変更により、複数行に跨るSQL文を記述することが可能となります。

FUJITSU Enterprise Postgres 9.5 の場合

SQL文の末尾にセミコロンを記述しない場合でも、1行を1つのSQL文として解釈します。

FUJITSU Enterprise Postgres 9.6以降の場合

SQL文の末尾にセミコロンを記述しない場合、次行以降を含めセミコロンまでを1つのSQL文として解釈します。

対処方法

独自スクリプトにセミコロンを付けずにSQL文を複数記述している場合は、各SQL文の行末にセミコロンを付け加えてください。ただし、SQL文の後にメタコマンドが続く行では、セミコロンを記述する必要ありません。

2.2.115 pgbenchコマンドのスクリプトファイルメタコマンド¥setrandomの削除

非互換

FUJITSU Enterprise Postgres 9.6以降では、pgbenchコマンドのスクリプトファイルメタコマンドの¥setrandomを削除します。

FUJITSU Enterprise Postgres 9.5 の場合

スクリプトファイルメタコマンドの¥setrandomを使用してランダムな数値を生成します。

[例]

```
¥setrandom r1 1 10

¥setrandom r2 1 10 exponential 3.0

update t1 set c2 = :r2 where c1 = :r1;
```

FUJITSU Enterprise Postgres 9.6以降の場合

組み込み関数を使用してランダムな数値を生成します。

[例]

```
¥set r1 random(1, 10)

¥set r2 random_exponential(1, 10, 3.0)

update t1 set c2 = :r2 where c1 = :r1;
```

対処方法

組み込み関数random、random_exponential、もしくはrandom_gaussianを使用してください。

2.2.116 hstoreのhstore_to_jsonb_loose関数の数値判別の厳格化

非互換

FUJITSU Enterprise Postgres 9.6以降では、hstoreのhstore_to_jsonb_loose関数において、JSON構文の数値の仕様に厳密に一致した場合にのみJSON数値に変換するように変更します。

FUJITSU Enterprise Postgres 9.5 の場合

JSON構文の数値の仕様に一致しない場合でもJSON数値に変換する場合があります。

[例]

```
=# SELECT hstore_to_jsonb_loose('key=>+1');

 hstore_to_jsonb_loose
-----------------------
 {"key": 1}
(1 row)


=#  SELECT hstore_to_json_loose('key=>+1');

 hstore_to_json_loose
----------------------
 {"key": "+1"}
(1 row)
```

FUJITSU Enterprise Postgres 9.6以降の場合

JSON構文の数値の仕様に一致する場合にのみJSON数値に変換します。

[例]

```
=# SELECT hstore_to_jsonb_loose('key=>+1');

 hstore_to_jsonb_loose
-----------------------
```

```
 {"key": "+1"}
(1 row)


=# SELECT hstore_to_json_loose('key=>+1');

 hstore_to_json_loose
----------------------
 {"key": "+1"}
(1 row)
```

対処方法

ありません。

2.2.117 pageinspectのheap_page_items関数の返却情報の追加

非互換

FUJITSU Enterprise Postgres 9.6以降では、pageinspectのheap_page_items関数の返却情報として、各タプルの生データをt_data列として新たに返すように変更します。

対処方法

ありません。

2.2.118 Oracleデータベース互換機能利用時の変更

非互換

FUJITSU Enterprise Postgres 9.6以降では、Oracleデータベース互換機能を利用するためには、新規インスタンス作成後、データベース“postgres”および“template1”に対して以下を実行してください。

```
CREATE EXTENSION oracle_compatible
```

対処方法

ありません。

2.3 運用の移行に関する非互換

項目	移行前のバージョン				
	9.5	9.6	10	11	12
pg_settingsビューのcategory列の値の変更	－	◎	◎	◎	◎
統計情報ビューpgx_stat_lwlockの変更	－	◎	◎	◎	◎
pgx_rcvallの動作の変更	◎	◎	◎	◎	◎
postgresql.confファイルのmax_files_per_processパラメータの指定値の範囲を変更	○	○	○	○	○
REINDEX CONCURRENTLYの前後でのプランナ統計情報の変更	－	－	－	－	○
ALTER TABLE ONLY ... ENABLE/DISABLE TRIGGERが子テーブルを再帰的に処理しないように変更	－	－	－	○	○
認証設定ファイルにUnixドメインソケットの認証設定の追加	○	○	○	○	○
統計情報ビューpg_stat_statementsの変更	○	○	○	○	○
統計情報ビューpg_stat_activityの変更	○	○	○	○	○

項目	移行前のバージョン				
	9.5	9.6	10	11	12
createuserコマンドのオプション「-a/--adduser」と「-A/--no-adduser」の削除	○	○	○	○	○
pg_waldumpの解析結果に「PREPARE TRANSACTION」に関連する情報の追加	○	○	○	○	○
EXPLAIN文にプラン作成時のバッファ使用量の追加	○	○	○	○	○
指定するリカバリ目標に到達しない場合の動作の変更	○	○	○	○	○
最小 TLS バージョンのデフォルト値の変更	－	－	－	－	○
EXPLAIN文に子テーブルに対応する列別名の追加	○	○	○	○	○
ログ出力先はcsvlogの場合のフォーマットの変更	○	○	○	○	○
postgresql.confファイルのwal_keep_segmentsパラメータ名をwal_keep_sizeに変更	○	○	○	○	○
postgresql.confファイルのeffective_io_concurrencyパラメータの設定方法の変更	○	○	○	○	○
Mirroring Controllerによるデータベースプロセスのダウン検知時の監視リトライの廃止	◎	◎	◎	◎	－
Mirroring Controllerの昇格後コマンドの名称とパラメータ名の変更	－	－	◎	◎	－
Mirroring Controllerのユーザーコマンドの入力値変更	－	－	◎	◎	－
postgresql.confファイルのdefault_with_oidsパラメータの削除	○	○	○	○	－
postgresql.confファイルのjitパラメータのデフォルト値の変更	－	－	－	○	－
pg_rewindの「--dry-run」オプションの動作の変更	○	○	○	○	－
pg_upgradeコマンドの動作の変更	○	○	○	○	－
外部キーのデフォルトの制約名の変更	○	○	○	○	－
max_connectionsの計算方法の変更	○	○	○	○	－
設定パラメータのデータ型の変更	○	○	○	○	－
autovacuum_vacuum_cost_delayのデフォルト値の変更	○	○	○	○	－
共通テーブル式(CTE)をインライン化するように変更	○	○	○	○	－
サーバの起動時にサーバのバージョン番号を出力するように変更	○	○	○	○	－
権限のないユーザにpg_stat_sslビューの行の表示を制限するように変更	○	○	○	○	－
pg_stat_sslシステムビューのclientdn列の改名	○	○	○	○	－
pg_stat_sslシステムビューにクライアント認証に関する情報を追加	○	○	○	○	－
グローバルオブジェクトをシステムビューpg_stat_databaseで追跡できるように変更	○	○	○	○	－
システムビューpg_stat_databaseへのchecksum_failures列とchecksum_last_failure列を追加	○	○	○	○	－
pg_attrdefシステムテーブルからadsrc列を削除	○	○	○	○	－
pg_constraintシステムテーブルからconsrc列を削除	○	○	○	○	－
pg_statistic_extシステムテーブルの分割	－	－	○	○	－

項目	移行前のバージョン				
	9.5	9.6	10	11	12
DROP IF EXISTS FUNCTION/PROCEDURE/ AGGREGATE/ROUTINEのメッセージの変更	―	―	○	○	―
Btreeインデックスの最大エントリ長の変更	○	○	○	○	―
postgresql.confファイルのrecovery_target_timelineパラメータのデフォルト値の変更	○	○	○	○	―
recovery_target_*パラメータの複数指定の不可	○	○	○	○	―
recovery.confの廃止	○	○	○	○	―
pg_procカタログのprotransform列をprosupport列に変更	○	○	○	○	―
コマンドの出力メッセージのフォーマットの変更	○	○	○	○	―
pg_dumpおよびpg_restoreコマンドのデフォルト動作の変更	○	○	○	―	―
pg_dumpallコマンドのデフォルト動作の変更	○	○	○	―	―
pg_dumpallの--globals-onlyオプションの動作の変更	○	○	○	―	―
非一時テーブルに対する一時テーブルのパーティション設定不可	○	○	○	―	―
関数と列参照のデフォルト動作の変更	○	○	○	―	―
テーブルとドメイン制約の命名規約を変更	○	○	○	―	―
pg_classカタログからrelhaspkey列を削除	○	○	○	―	―
情報スキーマのtablesビューのtable_type列のテーブル種類名の変更	○	○	○	―	―
pg_procカタログのproisagg列とproiswindow列をprokind列に統合	○	○	○	―	―
非スーパーユーザーによる共有カタログのインデックス再作成の防止	○	○	○	―	―
psコマンドで出力されるプロセス名をpg_stat_activityに同期	○	○	○	―	―
ロジカルレプリケーション時のTRUNCATE文の動作の変更	○	○	○	―	―
情報スキーマtriggersのaction_order列、action_reference_old_table列、action_reference_new_table列に格納する値の変更	○	○	○	―	―
情報スキーマtable_constraintsへのenforced列の追加	○	○	○	―	―
CREATE TABLE文 のLIKE句オプション INCLUDING ALL 指定の動作変更	○	○	○	―	―
postgresql.confファイルのreplacement_sort_tuplesパラメータの削除	○	○	○	―	―
CREATE FUCNTION 文のWITH句の廃止	○	○	○	―	―
EXPLAIN ANALYZE 文でのパラレルワーカーのソート処理情報の追加	○	○	○	―	―
プランナが抽出条件「カラム名 = カラム名」を「IS NOT NULL」へ置換する処理に変更	○	○	○	―	―
pg_rewindコマンドがrootユーザーで実行不可に変更	○	○	○	―	―
pg_stat_statements_resetの実行がスーパーユーザー以外で禁止に変更	○	○	○	―	―
pg_stat_statementsのqueryid列を64ビットに拡張	○	○	○	―	―

項目	移行前のバージョン				
	9.5	9.6	10	11	12
ビューに対する SELECT FOR UPDATE の動作変更	○	○	○	－	－
VACUUM のオプションに列リストを指定する場合の動作変更	○	○	○	－	－
システムビューpg_stat_wal_receiverへのsender_host列、sender_port列の追加	○	○	○	－	－
pg_dumpが出力するCREATE TRIGGER ... REFERENCINGコマンドの遷移テーブル名をダブルクォートで括るように変更	○	○	○	－	－
パーティションテーブルに対するCOPYコマンドの動作変更	○	○	○	－	－
論理レプリケーションスロットが存在する場合のサーバ起動可否判定の変更	○	○	○	－	－
1サーバ内での最大接続数の変更	◎	◎	－	－	－
ALTER TABLESPACE文による暗号化設定の変更	◎	◎	－	－	－
データ格納先ディレクトリのサブディレクトリの名称変更	○	○	－	－	－
コマンドおよびオプションの名称変更	○	○	－	－	－
pg_ctlコマンドの操作完了待ちのデフォルト動作の変更	○	○	－	－	－
pg_ctlコマンドのstartモードとrestartモードにおけるタイムアウト時の終了ステータスの変更	○	○	－	－	－
pg_basebackupコマンドのデフォルト動作の変更と-xオプションの削除	○	○	－	－	－
createlangコマンドとdroplangコマンドの削除	○	○	－	－	－
クライアントコマンドのsearch_path設定の変更	○	○	－	－	－
postgresql.confファイルのパラメータのデフォルト値の変更	○	○	－	－	－
postgresql.confファイルのmin_parallel_relation_sizeパラメータの削除とmin_parallel_table_scan_sizeパラメータおよびmin_parallel_index_scan_sizeパラメータの追加	○	○	－	－	－
postgresql.confファイルのsql_inheritanceパラメータの削除	○	○	－	－	－
postgresql.confファイルのpassword_encryptionパラメータの指定値の変更	○	○	－	－	－
postgresql.confファイルのshared_preload_librariesパラメータと関連するパラメータ群の大文字小文字区別の変更	○	○	－	－	－
RADIUS認証の設定オプションの名称変更	○	○	－	－	－
統計情報ビューpg_stat_activityの変更	○	○	－	－	－
統計情報ビューpg_stat_replicationの変更	○	○	－	－	－
システムビューpg_user_mappingsのumoptions列の可視性の厳格化	○	○	－	－	－
pg_stat_statementsビューに格納するSQL文の定数の置換方法の変更	○	○	－	－	－
CREATE ROLE文と同種のコマンド群のUNENCRYPTEDオプションの削除	○	○	－	－	－
CREATE ROLE文と同種のコマンド群における空パスワード指定時の動作変更	○	○	－	－	－

項目	移行前のバージョン				
	9.5	9.6	10	11	12
親テーブルにPRIMARY KEY制約を追加する場合の動作変更	○	○	－	－	－
レプリケーション接続認証のデフォルト設定の変更	○	○	－	－	－
論理レプリケーションのクライアント認証の設定方法の変更	○	○	－	－	－
データベース多重化機能によるクラスタシステムのデフォルト構成の変更	◎	◎	－	－	
mc_ctlコマンドのオプション省略時のデフォルト動作の変更	◎	◎	－	－	－
Mirroring Controllerがインスタンスに接続する際の接続設定の変更	◎	◎	－	－	－
Mirroring Controllerのサーバの状態表示の変更	◎	◎	－	－	－
データベース多重化運用中のsynchronous_standby_namesパラメータ変更時の動作の変更	◎	◎	－	－	－
CREATE ROLE文と同種のコマンド群のCREATEUSERオプションとNOCREATEUSERオプションの削除	○	－	－	－	－
pg_で始まるロール名を使用不可にする変更	○	－	－	－	－
pg_controldataおよび関連箇所におけるNextXIDの表示形式の変更	○	－	－	－	－
統計情報ビューpg_stat_activityのwaiting列の削除とwait_event_type列とwait_event列の追加	○	－	－	－	－
pg_amカタログの列の変更	○	－	－	－	－
システムビューpg_replication_slotsへのconfirmed_flush_lsn列の追加	○	－	－	－	－
システムビューpg_settingsのunit列に格納する値の変更	○	－	－	－	－
情報スキーマのroutinesビューのresult_cast_character_set_name列名の変更	○	－	－	－	－
update_process_titleパラメータのデフォルト値の変更	○	－	－	－	－
未サポートデータ型に対する秘匿化ポリシー定義の変更	◎	－	－	－	－

◎:FUJITSU Enterprise Postgresによる非互換あり

○:OSSによる非互換あり

－:非互換なし

2.3.1 pg_settingsビューのcategory列の値の変更

非互換

FUJITSU Enterprise Postgres 13では、pg_settingsビューのcategory列の値を変更します。

FUJITSU Enterprise Postgres 12以前の場合

本来の値	誤った値
Preset Options	Fujitsu Enterprise Postgres Parameters
Customized Options	Preset Options
Developer Options	Customized Options

これは、本来の値と別の値であるため、誤った結果です。

FUJITSU Enterprise Postgres 13の場合

pg_settingsビューのcategory列が本来の値となります。

対処方法

pg_settingsビューのcategory列を、移行前の値に置換することで、移行前と同一の実行結果を出力できるようにします。

2.3.2 統計情報ビューpgx_stat_lwlockの変更

非互換

FUJITSU Enterprise Postgres 13では、統計情報ビューpgx_stat_lwlockのlwlock_name列で表示される待機イベント名を変更します。

待機イベント名

FUJITSU Enterprise Postgres 12以前	FUJITSU Enterprise Postgres 13
clog (注1)	XactBuffer
commit_timestamp (注1)	CommitTSBuffer
subtrans (注1)	SubtransBuffer
multixact_offset (注1)	MultiXactOffsetBuffer
multixact_member (注1)	MultiXactMemberBuffer
async (注1)	NotifyBuffer
oldserxid (注1)	SerialBuffer
wal_insert (注1)	WALInsert
buffer_content (注1)	BufferContent
buffer_io (注1)	BufferIO
replication_origin (注1)	ReplicationOriginState
replication_slot_io (注1)	ReplicationSlotIO
proc (注1)	LockFastPath
buffer_mapping (注1)	BufferMapping
lock_manager (注1)	LockManager
predicate_lock_manager (注1)	PredicateLockManager
parallel_hash_join (注3)	ParallelHashJoin
parallel_query_dsa (注2)	ParallelQueryDSA
session_dsa (注3)	PerSessionDSA
session_record_table (注3)	PerSessionRecordType
session_typmod_table (注3)	PerSessionRecordTypmod
shared_tuplestore (注3)	SharedTupleStore
tbm (注2)	SharedTidBitmap
parallel_append (注3)	ParallelAppend
serializable_xact (注4)	PerXactPredicateList
shared_mcxt (注4)	SharedMcxt
meta_cache_map (注4)	MetaCacheMap
global_metacache (注4)	GlobalCatcache
cached_buf_tranche_id (注4)	CachedBufTranche

注1）FUJITSU Enterprise Postgres 9.6で追加したイベントです。

注2）FUJITSU Enterprise Postgres 10で追加したイベントです。

注3）FUJITSU Enterprise Postgres 11で追加したイベントです。

注4）FUJITSU Enterprise Postgres 12で追加したイベントです。

対処方法

ありません。

2.3.3 pgx_rcvallの動作の変更

非互換

FUJITSU Enterprise Postgres 13では、pgx_rcvallコマンドの-eオプションに未来の時間を指定、または、-nオプションに存在しないリストアポイントを指定した場合、pgx_rcvallコマンドが失敗するように変更します。

FUJITSU Enterprise Postgres 12以前の場合

pgx_rcvallコマンドの-eオプションに未来の時間を指定、または、-nオプションに存在しないリストアポイントを指定した場合、すべてのアーカイブ済みWALを適用してリカバリが実施されます。

FUJITSU Enterprise Postgres 13の場合

pgx_rcvallコマンドの-eオプションに未来の時間を指定、または、-nオプションに存在しないリストアポイントを指定した場合、pgx_rcvallコマンドが失敗します。

対処方法

必要に応じて、リカバリ目標を正しく指定してください。

2.3.4 postgresql.confファイルのmax_files_per_processパラメータの指定値の範囲を変更

非互換

FUJITSU Enterprise Postgres 13では、postgresql.confファイルのmax_files_per_processパラメータに設定する値の範囲を変更します。

FUJITSU Enterprise Postgres 12以前の場合

25～2147483647までの値が指定できます。

FUJITSU Enterprise Postgres 13の場合

64～2147483647までの値が指定できます。

対処方法

必要に応じて、max_files_per_processに64以上の値を設定してください。

2.3.5 REINDEX CONCURRENTLYの前後でのプランナ統計情報の変更

非互換

FUJITSU Enterprise Postgres 13では、REINDEX CONCURRENTLYの前後でインデックス式のプランナ統計情報が維持されるようになります。

FUJITSU Enterprise Postgres 12以前の場合

REINDEX CONCURRENTLY の前後でインデックス式のプランナ統計情報が維持しません。

[例]

```
=# CREATE TABLE concur_exprs_tab (c1 int , c2 boolean);
=# INSERT INTO concur_exprs_tab (c1, c2) VALUES (1369652450, FALSE);
```

```
=# CREATE UNIQUE INDEX concur_exprs_index_expr
-#   ON concur_exprs_tab ((c1::text COLLATE "C"));
=# ANALYZE concur_exprs_tab;
=# SELECT starelid::regclass, count(*) FROM pg_statistic WHERE starelid IN (
(#   'concur_exprs_index_expr'::regclass)
-#   GROUP BY starelid ORDER BY starelid::regclass::text;
        starelid         | count
-------------------------+-------
 concur_exprs_index_expr |     1
(1 row)
=# REINDEX TABLE CONCURRENTLY concur_exprs_tab;
=# SELECT starelid::regclass, count(*) FROM pg_statistic WHERE starelid IN (
(#   'concur_exprs_index_expr'::regclass)
-#   GROUP BY starelid ORDER BY starelid::regclass::text;
 starelid | count
----------+-------
(0 row)
```

FUJITSU Enterprise Postgres 13の場合

REINDEX CONCURRENTLY の前後でインデックス式のプランナ統計情報が維持されます。

[例]

```
=# CREATE TABLE concur_exprs_tab (c1 int , c2 boolean);
=# INSERT INTO concur_exprs_tab (c1, c2) VALUES (1369652450, FALSE);
=# CREATE UNIQUE INDEX concur_exprs_index_expr
-#   ON concur_exprs_tab ((c1::text COLLATE "C"));
=# ANALYZE concur_exprs_tab;
=# SELECT starelid::regclass, count(*) FROM pg_statistic WHERE starelid IN (
(#   'concur_exprs_index_expr'::regclass)
-#   GROUP BY starelid ORDER BY starelid::regclass::text;
        starelid         | count
-------------------------+-------
 concur_exprs_index_expr |     1
(1 row)
=# REINDEX TABLE CONCURRENTLY concur_exprs_tab;
=# SELECT starelid::regclass, count(*) FROM pg_statistic WHERE starelid IN (
(#   'concur_exprs_index_expr'::regclass)
-#   GROUP BY starelid ORDER BY starelid::regclass::text;
        starelid         | count
-------------------------+-------
 concur_exprs_index_expr |     1
(1 row)
```

対処方法

ありません。

2.3.6 ALTER TABLE ONLY ... ENABLE/DISABLE TRIGGERが子テーブルを再帰的に処理しないように変更

非互換

FUJITSU Enterprise Postgres 13では、ALTER TABLE ONLY ... ENABLE/DISABLE TRIGGER が子テーブルを再帰的に処理しないように変更しました。

FUJITSU Enterprise Postgres 12以前の場合

「ONLY」が無視されて、子テーブルまで再帰的に処理されます。

[例]

```
=# create table parent (a int) partition by list (a);
=# create table child1 partition of parent for values in (1);
```

```
=# create trigger tg after insert on parent
-#   for each row execute procedure trig_nothing();
=# select tgrelid::regclass, tgname, tgenabled from pg_trigger
-#   where tgrelid in ('parent'::regclass, 'child1'::regclass)
-#  order by tgrelid::regclass::text;
 tgrelid | tgname | tgenabled
---------+--------+-----------
 child1  | tg     | O
 parent  | tg     | O
(2 row)
=# alter table only parent enable always trigger tg;
=# select tgrelid::regclass, tgname, tgenabled from pg_trigger
-#  where tgrelid in ('parent'::regclass, 'child1'::regclass)
-#  order by tgrelid::regclass::text;
 tgrelid | tgname | tgenabled
---------+--------+-----------
 child1  | tg     | A
 parent  | tg     | A
(2 row)
```

FUJITSU Enterprise Postgres 13の場合

子テーブルを再帰的に処理しません。

[例]

```
=# create table parent (a int) partition by list (a);
=# create table child1 partition of parent for values in (1);
=# create trigger tg after insert on parent
-#   for each row execute procedure trig_nothing();
=# select tgrelid::regclass, tgname, tgenabled from pg_trigger
-#   where tgrelid in ('parent'::regclass, 'child1'::regclass)
-#  order by tgrelid::regclass::text;
 tgrelid | tgname | tgenabled
---------+--------+-----------
 child1  | tg     | O
 parent  | tg     | O
(2 row)
=# alter table only parent enable always trigger tg;
=# select tgrelid::regclass, tgname, tgenabled from pg_trigger
-#  where tgrelid in ('parent'::regclass, 'child1'::regclass)
-#  order by tgrelid::regclass::text;
 tgrelid | tgname | tgenabled
---------+--------+-----------
 child1  | tg     | O
 parent  | tg     | A
(2 rows)
```

対処方法

必要に応じて、「only」を削除してください。

2.3.7 認証設定ファイルにUnixドメインソケットの認証設定の追加

非互換

FUJITSU Enterprise Postgres 13では、Unixドメインソケットをサポートするように変更します。

pg_hba.confファイルのデフォルト設定が変更されます。

FUJITSU Enterprise Postgres 12以前の場合

Unixドメインソケットの認証設定のレコードがありません。

[例]

```
# IPv4 local connections:
host    all             all             127.0.0.1/32            trust
# Allow replication connections from localhost, by a user with the
# replication privilege.
host    replication     all             127.0.0.1/32            trust
```

FUJITSU Enterprise Postgres 13の場合

Unixドメインソケットの認証設定のレコードが追加されます。

[例]

```
# "local" is for Unix domain socket connections only
local   all             all                                     trust
# IPv4 local connections:
host    all             all             127.0.0.1/32            trust
# Allow replication connections from localhost, by a user with the
# replication privilege.
local   replication     all                                     trust
host    replication     all             127.0.0.1/32            trust
```

対処方法

Unixドメインソケットの認証を使用しない場合は、Unixドメインソケットの認証設定の行をコメントアウトしてください。

2.3.8 統計情報ビューpg_stat_statementsの変更

非互換

FUJITSU Enterprise Postgres 13では、以下のとおり、統計情報ビューpg_stat_statementsの列名を変更します。また、WAL使用状況と実行計画の作成時間の列を追加します。

列名の変更

FUJITSU Enterprise Postgres 12以前	FUJITSU Enterprise Postgres 13
total_time	total_exec_time
min_time	total_exec_time
max_time	max_exec_time
mean_time	mean_exec_time
stddev_time	stddev_exec_time

列の追加

列名
plans
total_plan_time
min_plan_time
max_plan_time
mean_plan_time
stddev_plan_time
wal_records
wal_fpi
wal_bytes

対処方法

ありません。

2.3.9 統計情報ビューpg_stat_activityの変更

非互換

FUJITSU Enterprise Postgres 13では、以下のとおり、統計情報ビューpg_stat_activityにおける待機イベント名の変更、追加、削除をします。

待機イベント名の変更

待機イベントタイプ	待機イベント名	
	FUJITSU Enterprise Postgres 12 以前	FUJITSU Enterprise Postgres 13
LWLock	ShmemIndexLock	ShmemIndex
	OidGenLock	OidGen
	XidGenLock	XidGen
	ProcArrayLock	ProcArray
	SInvalReadLock	SInvalRead
	SInvalWriteLock	SInvalWrite
	WALBufMappingLock	WALBufMapping
	WALWriteLock	WALWrite
	ControlFileLock	ControlFile
	CheckpointLock	Checkpoint
	CLogControlLock	XactSLRU
	SubtransControlLock	SubtransSLRU
	MultiXactGenLock	MultiXactGen
	MultiXactOffsetControlLock	MultiXactOffsetSLRU
	MultiXactMemberControlLock	MultiXactMemberSLRU
	RelCacheInitLock	RelCacheInit
	CheckpointerCommLock	CheckpointerComm
	TwoPhaseStateLock	TwoPhaseState
	TablespaceCreateLock	TablespaceCreate
	BtreeVacuumLock	BtreeVacuum
	AddinShmemInitLock	AddinShmemInit
	AutovacuumLock	Autovacuum
	AutovacuumScheduleLock	AutovacuumSchedule
	SyncScanLock	SyncScan
	RelationMappingLock	RelationMapping
	AsyncCtlLock	NotifySLRU
	AsyncQueueLock	NotifyQueue
	SerializableXactHashLock	SerializableXactHash
	SerializableFinishedListLock	SerializableFinishedList
	SerializablePredicateLockListLock	SerializablePredicateList

待機イベントタイプ	待機イベント名	
	FUJITSU Enterprise Postgres 12 以前	FUJITSU Enterprise Postgres 13
	OldSerXidLock	SerialSLRU
	SyncRepLock	SyncRep
	BackgroundWorkerLock	BackgroundWorker
	DynamicSharedMemoryControlLock	DynamicSharedMemoryControl
	AutoFileLock	AutoFile
	ReplicationSlotAllocationLock	ReplicationSlotAllocation
	ReplicationSlotControlLock	ReplicationSlotControl
	CommitTsControlLock	CommitTsSLRU
	CommitTsLock	CommitTs
	ReplicationOriginLock	ReplicationOrigin
	MultiXactTruncationLock	MultiXactTruncation
	OldSnapshotTimeMapLock	OldSnapshotTimeMap
	LogicalRepWorkerLock	LogicalRepWorker
	CLogTruncationLock	XactTruncation
	clog	XactBuffer
	commit_timestamp	CommitTSBuffer
	subtrans	SubtransBuffer
	multixact_offset	MultiXactOffsetBuffer
	multixact_member	MultiXactMemberBuffer
	async	NotifyBuffer
	oldserxid	SerialBuffer
	wal_insert	WALInsert
	buffer_content	BufferContent
	buffer_io	BufferIO
	replication_origin	ReplicationOriginState
	replication_slot_io	ReplicationSlotIO
	proc	LockFastPath
	buffer_mapping	BufferMapping
	lock_manager	LockManager
	predicate_lock_manager	PredicateLockManager
	serializable_xact	PerXactPredicateList
	parallel_query_dsa	ParallelQueryDSA
	tbm	SharedTidBitmap
	parallel_append	ParallelAppend
	parallel_hash_join	ParallelHashJoin
	session_dsa	PerSessionDSA
	session_record_table	PerSessionRecordType
	session_typmod_table	PerSessionRecordTypmod

待機イベントタイプ	待機イベント名	
	FUJITSU Enterprise Postgres 12 以前	FUJITSU Enterprise Postgres 13
	shared_tuplestore	SharedTupleStore
Lock	speculative token	spectoken
IPC	Hash/Batch/Allocating	HashBatchAllocate
	Hash/Batch/Electing	HashBatchElect
	Hash/Batch/Loading	HashBatchLoad
	Hash/Build/Allocating	HashBuildAllocate
	Hash/Build/Electing	HashBuildElect
	Hash/Build/HashingInner	HashBuildHashInner
	Hash/Build/HashingOuter	HashBuildHashOuter
	Hash/GrowBatches/Allocating	HashGrowBatchesAllocate
	Hash/GrowBatches/Deciding	HashGrowBatchesDecide
	Hash/GrowBatches/Electing	HashGrowBatchesElect
	Hash/GrowBatches/Finishing	HashGrowBatchesFinish
	Hash/GrowBatches/Repartitioning	HashGrowBatchesRepartition
	Hash/GrowBuckets/Allocating	HashGrowBucketsAllocate
	Hash/GrowBuckets/Electing	HashGrowBucketsElect
	Hash/GrowBuckets/Reinserting	HashGrowBucketsReinsert

待機イベント名の追加

待機イベントタイプ	待機イベント名
LWLock	WrapLimitsVacuum
	NotifyQueueTail
IPC	BackupWaitWalArchive
	ProcSignalBarrier
	RecoveryConflictSnapshot
	RecoveryConflictTablespace
	RecoveryPause
	XactGroupUpdate
Timeout	RecoveryRetrieveRetryInterval
	VacuumDelay

待機イベント名の削除

待機イベントタイプ	待機イベント名
Activity	RecoveryWalAll
IPC	ClogGroupUpdate

対処方法

ありません。

2.3.10 createuserコマンドのオプション「-a/--adduser」と「-A/--no-adduser」の削除

非互換

FUJITSU Enterprise Postgres 13では、createuserコマンドのオプション「-a/--adduser」と「-A/--no-adduser」を削除します。

「-a/--adduser」は「-s/--superuser」に置き換えられて、「-A/--no-adduser」は「-S/--no-superuser」に置き換えられます。これまでは互換のために「-a/--adduser」と「-A/--no-adduser」を残していました。

対処方法

「-s/--superuser」と「-S/--no-superuser」を使用してください。

2.3.11 pg_waldumpの解析結果に「PREPARE TRANSACTION」に関連する情報を追加

非互換

FUJITSU Enterprise Postgres 13では、pg_waldumpの解析結果に「PREPARE TRANSACTION」に関連する情報が増えます。グローバルトランザクション識別子などの情報も含むように修正します。

FUJITSU Enterprise Postgres 12以前の場合

グローバルトランザクション識別子などの情報がありません。

[例]

```
$ pg_waldump 000000010000000000000002
rmgr: Transaction len (rec/tot):    242/   242, tx:        665, lsn: 0/02040FB0, prev 0/02040F70, desc: PREPARE
```

FUJITSU Enterprise Postgres 13の場合

グローバルトランザクション識別子などの情報が表示されます。

[例]

```
$ pg_waldump 000000010000000000000002
rmgr: Transaction len (rec/tot):    242/   242, tx:        624, lsn: 0/020190D0, prev 0/02019090, desc: PREPARE gid
test_prepared#1: 2020-09-16 05:55:46.787031 EDT
```

対処方法

ありません。

2.3.12 EXPLAIN文にプラン作成時のバッファ使用量を追加

非互換

FUJITSU Enterprise Postgres 13では、EXPLAINコマンドのBUFFERオプションを指定する場合、EXPLAIN文にプラン作成時のバッファ使用量が追加されます。

FUJITSU Enterprise Postgres 12以前の場合

プラン作成時のバッファ使用量の情報がありません。

[例]

```
=# explain (analyze, buffers, format yaml) select * from int8_tbl i8;
           QUERY PLAN
--------------------------------
 - Plan:                       +
     Node Type: "Seq Scan"     +
     ...
   Planning Time: 0.052        +
   Triggers:                   +
```

```
   Execution Time: 0.020
(1 row)
```

FUJITSU Enterprise Postgres 13の場合

プラン作成時のバッファ使用量の情報が追加されます。

[例]

```
=# explain (analyze, buffers, format yaml) select * from int8_tbl i8;
           QUERY PLAN
--------------------------------
 - Plan:                       +
     Node Type: "Seq Scan"     +
     ...
   Planning:                   +
     Shared Hit Blocks: 13     +
     ...
     Temp Written Blocks: 0    +
   Planning Time: 0.148        +
   Triggers:                   +
   Execution Time: 0.013
(1 row)
```

対処方法

ありません。

2.3.13 指定するリカバリ目標に到達しない場合の動作を変更

非互換

FUJITSU Enterprise Postgres 13では、指定するリカバリ目標に到達しない場合、リカバリ処理はエラーになります。

FUJITSU Enterprise Postgres 12以前の場合

指定するリカバリ目標に到達しなくても、WALの最後までリカバリされます。

[例]

```
$ vim postgresql.conf
restore_command = 'cp /home/postgres/archivedir/%f %p'
recovery_target_name = 'does_not_exist'
:wq
$ touch recovery.signal

$ pg_ctl -D pgdata -l logfile start
waiting for server to start.... done
server started
```

FUJITSU Enterprise Postgres 13の場合

指定するリカバリ目標に到達しない場合、リカバリ処理はエラーになります。

[例]

```
$ vim postgresql.conf
restore_command = 'cp /home/postgres/archivedir/%f %p'
recovery_target_name = 'does_not_exist'
:wq
$ touch recovery.signal
$ pg_ctl -D pgdata -l logfile start
waiting for server to start.... stopped waiting
pg_ctl: could not start server
Examine the log output.
```

また、ログファイルに「FATAL: recovery ended before configured recovery target was reached」というメッセージが出力されます。

対処方法

必要に応じて、リカバリ目標を正しく指定してください。

2.3.14 最小TLSバージョンのデフォルト値の変更

非互換

FUJITSU Enterprise Postgres 13では、接続プロトコルの最小TLS バージョンのデフォルト値を「1.0」から「1.2」に変更します。

FUJITSU Enterprise Postgres 12以前の場合

最小TLS バージョンのデフォルト値はv1.0です。

[例]

```
=# select name,boot_val from pg_settings where name='ssl_min_protocol_version';
           name           | boot_val
--------------------------+----------
 ssl_min_protocol_version | TLSv1
(1 row)
```

デフォルト設定の場合、TLSv1またはTLSv1.1のクライアントから接続できます。

FUJITSU Enterprise Postgres 13の場合

最小TLS バージョンのデフォルト値はv1.2です。

[例]

```
=# select name,boot_val from pg_settings where name='ssl_min_protocol_version';
           name           | boot_val
--------------------------+----------
 ssl_min_protocol_version | TLSv1.2
(1 row)
```

デフォルト設定の場合、TLSv1またはTLSv1.1のクライアントからの接続に失敗します。サーバ側で下記のようなエラーが出力されます。

[エラーメッセージの例]

```
「This may indicate that the client does not support any SSL protocol version between TLSv1.2 and TLSv1.2.」
```

対処方法

必要に応じて、ssl_min_protocol_versionパラメータの設定値を変更してください。

2.3.15 EXPLAIN文に子テーブルに対応する列別名の追加

非互換

FUJITSU Enterprise Postgres 13では、EXPLAIN文に子テーブルに対応する列別名の情報が追加されます。

FUJITSU Enterprise Postgres 12以前の場合

EXPLAIN文に子テーブル名だけを出力します。

[例]

```
=# create temp table patest0 (id int, x int);
CREATE TABLE
=# create temp table patest1() inherits (patest0);
CREATE TABLE
=# explain (costs off) select * from patest0;
        QUERY PLAN
---------------------------
```

```
 Append
   ->  Seq Scan on patest0
   ->  Seq Scan on patest1
(3 rows)
```

FUJITSU Enterprise Postgres 13の場合

EXPLAIN文に子テーブルに対応する列別名を出力します。

[例]

```
=# create temp table patest0 (id int, x int);
CREATE TABLE
=# create temp table patest1() inherits (patest0);
CREATE TABLE
=# explain (costs off) select * from patest0;
             QUERY PLAN
------------------------------------
 Append
   ->  Seq Scan on patest0 patest0_1
   ->  Seq Scan on patest1 patest0_2
(3 rows)
```

対処方法

ありません。

2.3.16 ログ出力先がcsvlogの場合のフォーマットを変更

非互換

FUJITSU Enterprise Postgres 13では、ログ出力先としてcsvlogを設定する場合、ログの最後に「バックエンドタイプ」が追加されます。

FUJITSU Enterprise Postgres 12以前の場合

CSVファイルに出力するログに、「バックエンドタイプ」を出力しません。

[例]

```
…,"postgres","postgres",…,"psql"
```

FUJITSU Enterprise Postgres 13の場合

CSVファイルに出力するログに、「バックエンドタイプ」を出力します。

[例]

```
…,"postgres","postgres",…,"psql","client backend"
```

対処方法

ありません。

2.3.17 postgresql.confファイルのwal_keep_segmentsパラメータ名をwal_keep_sizeに変更

非互換

FUJITSU Enterprise Postgres 13では、postgresql.confファイルのwal_keep_segmentsパラメータ名をwal_keep_sizeに変更します。設定値の計算方法も変更します。

対処方法

以前のwal_keep_segments値より、下記の方法でwal_keep_sizeの値を計算して、wal_keep_sizeに設定してください。

wal_keep_size = wal_keep_segments * wal_segment_size（デフォルトは16MB）

2.3.18 postgresql.confファイルのeffective_io_concurrencyパラメータの設定方法の変更

非互換

FUJITSU Enterprise Postgres 13では、effective_io_concurrencyパラメータに設定する数値に対して、計算方法が変更されます。

対処方法

以前の値(OLDVALUE)より、下記の方法で新しい値を計算して、effective_io_concurrencyに設定してください。

```
SELECT round(sum(OLDVALUE / n::float)) FROM generate_series(1, OLDVALUE) s(n);
```

2.3.19 Mirroring Controllerによるデータベースプロセスのダウン検知時の監視リトライの廃止

非互換

FUJITSU Enterprise Postgres 12のMirroring Controllerでは、データベースプロセスの生死監視でダウンを検知した場合に、監視のリトライを行わないように変更します。

対処方法

ありません。

2.3.20 Mirroring Controllerの昇格後コマンドの名称とパラメータ名の変更

非互換

FUJITSU Enterprise Postgres 12のMirroring Controllerでは、状態遷移コマンドである昇格後コマンドの名称、および昇格後コマンドを指定するサーバ定義ファイルのパラメータ名を変更します。

FUJITSU Enterprise Postgres 11以前の場合

- コマンド名

  昇格後コマンド

- 昇格後コマンドを指定するサーバ定義ファイルのパラメータ名

  post_promote_command

FUJITSU Enterprise Postgres 12の場合

- コマンド名

  切り替え後コマンド

- 昇格後コマンドを指定するサーバ定義ファイルのパラメータ名

  post_switch_command

対処方法

サーバ定義ファイルのpost_promote_commandパラメータは、FUJITSU Enterprise Postgres 12以降も継続して使用可能です。サーバ定義ファイルに指定した場合、切り替え後コマンドとして動作します。post_promote_commandパラメータとpost_switch_commandパラメータは同時に指定することはできません。

2.3.21 Mirroring Controllerのユーザーコマンドの入力値変更

非互換

Mirroring Controllerの災害対策運用において、以下のユーザーコマンドに「センタ種別」の入力値を追加します。

- データベースサーバのフェンシングコマンド
- 裁定コマンド
- 切り替え後コマンド
- 切り離し前コマンド
- 組み込み後コマンド

対処方法

ユーザーコマンドで引数の数を確認している場合は、引数の数が1増加する対処を実施してください。

2.3.22 postgresql.confファイルのdefault_with_oidsパラメータの削除

非互換

FUJITSU Enterprise Postgres 12では、「oid」列を制御するdefault_with_oidsパラメータをpostgresql.confから削除します。

対処方法

ありません。

2.3.23 postgresql.confファイルのjitパラメータのデフォルト値の変更

非互換

FUJITSU Enterprise Postgres 12では、実行時コンパイル(just-in-time compilation)の使用を制御するjitパラメータのデフォルト値を「off」から「on」に変更します。

対処方法

必要に応じて、パラメータの設定を変更してください。

2.3.24 pg_rewindの「--dry-run」オプションの動作の変更

非互換

FUJITSU Enterprise Postgres 12では、pg_rewindの「--dry-run」オプションでpg_controlファイルの更新をしません。

FUJITSU Enterprise Postgres 11以前の場合

pg_rewindの「--dry-run」オプションでpg_controlファイルを更新します。

FUJITSU Enterprise Postgres 12の場合

pg_rewindの「--dry-run」オプションでpg_controlファイルを更新しません。

対処方法

ありません。

2.3.25 pg_upgradeコマンドの動作の変更

非互換

FUJITSU Enterprise Postgres 12では、pg_upgradeコマンドでsql_identifier型の列を持つテーブルを拒否します。

対処方法

pg_upgradeコマンドを実行する前に、sql_identifier型をvarchar型に置き換えてください。

2.3.26 外部キーのデフォルトの制約名の変更

非互換

FUJITSU Enterprise Postgres 12では、外部キーのデフォルトの制約名を命名するときに、すべてのキー列の名前を使用します。

FUJITSU Enterprise Postgres 11以前の場合

外部キーのデフォルトの制約名に最初の列名のみが含まれます。

[例]

```
=# CREATE TABLE tb1(c1 int,c2 int,PRIMARY KEY(c1,c2));
CREATE TABLE
=# CREATE TABLE tb2(id int,c1 int,c2 int,FOREIGN KEY(c1,c2) REFERENCES tb1);
CREATE TABLE
=# ¥d tb2
               Table "public.tb2"
 Column |  Type   | Collation | Nullable | Default
--------+---------+-----------+----------+---------
 id     | integer |           |          |
 c1     | integer |           |          |
 c2     | integer |           |          |
Foreign-key constraints:
    "tb2_c1_fkey" FOREIGN KEY (c1, c2) REFERENCES tb1(c1, c2)
```

FUJITSU Enterprise Postgres 12の場合

外部キーのデフォルトの制約名にすべてのキー列の名前を使用します。

[例]

```
=# CREATE TABLE tb1(c1 int,c2 int,PRIMARY KEY(c1,c2));
CREATE TABLE
=# CREATE TABLE tb2(id int,c1 int,c2 int,FOREIGN KEY(c1,c2) REFERENCES tb1);
CREATE TABLE
=# ¥d tb2
               Table "public.tb2"
 Column |  Type   | Collation | Nullable | Default
--------+---------+-----------+----------+---------
 id     | integer |           |          |
 c1     | integer |           |          |
 c2     | integer |           |          |
Foreign-key constraints:
    "tb2_c1_c2_fkey" FOREIGN KEY (c1, c2) REFERENCES tb1(c1, c2)
```

対処方法

ありません。

2.3.27 max_connectionsの計算方法の変更

非互換

FUJITSU Enterprise Postgres 12では、max_wal_sendersはmax_connectionsの一部としてカウントします。

対処方法

ありません。

2.3.28 設定パラメータのデータ型の変更

非互換

FUJITSU Enterprise Postgres 12では、autovacuum_vacuum_cost_delayとvacuum_cost_delayのデータ型をintegerからrealに変更します。

対処方法

ありません。

2.3.29 autovacuum_vacuum_cost_delayのデフォルト値の変更

非互換

FUJITSU Enterprise Postgres 12では、autovacuum_vacuum_cost_delayのデフォルト値を20msから2msに変更します。

対処方法

ありません。

2.3.30 共通テーブル式(CTE)をインライン化するように変更

非互換

FUJITSU Enterprise Postgres 12では、共通テーブル式(CTE)を外部の問い合わせにインライン化できるようになります。

FUJITSU Enterprise Postgres 11以前の場合

CTEはインライン化されません。

[例]

```
=# EXPLAIN WITH t AS(SELECT * FROM test_tb) SELECT * FROM t;
                            QUERY PLAN
------------------------------------------------------------------
 CTE Scan on t  (cost=18.80..36.40 rows=880 width=64)
   CTE t
     ->  Seq Scan on test_tb  (cost=0.00..18.80 rows=880 width=64)
(3 rows)
```

FUJITSU Enterprise Postgres 12の場合

CTEはインライン化されます。

[例]

```
=# EXPLAIN WITH t AS(SELECT * FROM test_tb) SELECT * FROM t;
                          QUERY PLAN
-------------------------------------------------------------
 Seq Scan on test_tb  (cost=0.00..7213.00 rows=500000 width=4)
(1 row)
```

対処方法

ありません。

2.3.31 サーバの起動時にサーバのバージョン番号を出力するように変更

非互換

FUJITSU Enterprise Postgres 12では、サーバの起動時に、サーバのバージョン番号を含むログメッセージを出力するようになります。

対処方法

ありません。

2.3.32 権限のないユーザーにpg_stat_sslビューの行の表示を制限するように変更

非互換

FUJITSU Enterprise Postgres 12では、権限のないユーザーに pg_stat_sslビューの行の表示が制限されます。

対処方法

ありません。

2.3.33 pg_stat_sslシステムビューのclientdn列の改名

非互換

FUJITSU Enterprise Postgres 12では、clientdn列がclient_dnに改名されます。

対処方法

ありません。

2.3.34 pg_stat_sslシステムビューにクライアント認証に関する情報を追加

非互換

FUJITSU Enterprise Postgres 12では、pg_stat_sslシステムビューにclient_serial列とissuer_dn列が追加されます。

対処方法

ありません。

2.3.35 グローバルオブジェクトをシステムビューpg_stat_databaseで追跡できるように変更

非互換

FUJITSU Enterprise Postgres 12では、グローバルオブジェクトをシステムビューpg_stat_databaseで追跡できるようになります。

FUJITSU Enterprise Postgres 11以前の場合

グローバルオブジェクトをシステムビューpg_stat_databaseで追跡できません。

[例]

```
=# SELECT datid,datname from pg_stat_database;
 datid |  datname
-------+-----------
 13972 | postgres
     1 | template1
 13971 | template0
(3 rows)
```

FUJITSU Enterprise Postgres 12の場合

グローバルオブジェクトをシステムビューpg_stat_databaseで追跡できます。

[例]

```
=# SELECT datid,datname from pg_stat_database;
 datid |  datname
-------+-----------
```

```
     0 |
 14205 | postgres
     1 | template1
 14204 | template0
(4 rows)
```

対処方法

ありません。

2.3.36 システムビューpg_stat_databaseにchecksum_failures列とchecksum_last_failure列を追加

非互換

FUJITSU Enterprise Postgres 12では、pg_stat_database にチェックサムエラーのカウンタが加わります。

対処方法

ありません。

2.3.37 pg_attrdefシステムテーブルからadsrc列を削除

非互換

FUJITSU Enterprise Postgres 12では、pg_attrdefシステムテーブルのadsrc列が削除されます。列デフォルト値の定義はadbin列に格納されています。

対処方法

ありません。

2.3.38 pg_constraintシステムテーブルからconsrc列を削除

非互換

FUJITSU Enterprise Postgres 12では、pg_constraintシステムテーブルのconsrc列が削除されます。制約の定義はconbin列に格納されています。

対処方法

ありません。

2.3.39 pg_statistic_extシステムテーブルの分割

非互換

FUJITSU Enterprise Postgres 12では、pg_statistic_extシステムテーブルが2つのシステムテーブル（pg_statistic_extとpg_statistic_ext_data）に分割され、それらのビューであるpg_stats_extが追加されます。

対処方法

ありません。

2.3.40 DROP IF EXISTS FUNCTION/PROCEDURE/AGGREGATE/ROUTINEのメッセージの変更

非互換

FUJITSU Enterprise Postgres 12では、DROP IF EXISTS FUNCTION/PROCEDURE/AGGREGATE/ROUTINEが、引数リストを省略していて一致するオブジェクトが複数あるときに、エラーを出力すようになります。

FUJITSU Enterprise Postgres 11以前の場合

引数リストを省略していて一致するオブジェクトが複数あるときに、「NOTICE」を出力します。

[例]

```
=# CREATE FUNCTION f1() RETURNS int LANGUAGE sql AS $$ SELECT 1 $$;
CREATE FUNCTION
=# CREATE FUNCTION f1(p int) RETURNS int LANGUAGE sql AS $$ SELECT 1 $$;
CREATE FUNCTION
=# DROP FUNCTION IF EXISTS f1;
NOTICE:  function f1() does not exist, skipping (11891)
DROP FUNCTION
```

FUJITSU Enterprise Postgres 12の場合

引数リストを省略していて一致するオブジェクトが複数あるときに、「ERROR」を出力します。

[例]

```
=# CREATE FUNCTION f1() RETURNS int LANGUAGE sql AS $$ SELECT 1 $$;
CREATE FUNCTION
=# CREATE FUNCTION f1(p int) RETURNS int LANGUAGE sql AS $$ SELECT 1 $$;
CREATE FUNCTION
=# DROP FUNCTION IF EXISTS f1;
ERROR:  function name "f1" is not unique
HINT:  Specify the argument list to select the function unambiguously.
```

対処方法

ありません。

2.3.41 Btreeインデックスの最大エントリ長の変更

非互換

FUJITSU Enterprise Postgres 12では、新たなBtreeインデックスでインデックスの最大エントリ長が8バイト減ります。pg_upgradeコマンドでアップグレードしたデータベースクラスタでREINDEX操作をしたときに失敗する可能性があります。

対処方法

ありません。

2.3.42 postgresql.confファイルのrecovery_target_timelineパラメータのデフォルト値の変更

非互換

FUJITSU Enterprise Postgres 12では、recovery_target_timelineはlatestがデフォルトになりました。これまではcurrentがデフォルトでした。

対処方法

必要に応じてrecovery_target_timelineパラメータの設定を変更してください。

2.3.43 recovery_target_*パラメータの複数指定の不可

非互換

FUJITSU Enterprise Postgres 12では、recovery_target、recovery_target_lsn、recovery_target_name、recovery_target_time、recovery_target_xidのうち1つだけが指定できるようになりました。これまでは、これらの複数指定が可能で、最後に指定されたパラメータの設定に従って動作していました。FUJITSU Enterprise Postgres 12では、同一パラメータを複数指定することは可能で、その場合は最後に指定された値が適用されます。

対処方法

ありません。

2.3.44 recovery.confの廃止

非互換

FUJITSU Enterprise Postgres 12では、recovery.confが廃止され、recovery.confのパラメータはpostgresql.confファイルに統合されました。さらに、以下の変更点があります。

- recovery.confファイルを配置するとインスタンスは起動できません。
- 非プライマリモードへの切り替えは、recovery.signalファイル、standby.signalファイルが使われます。
- postgresql.confファイルのstandby_modeが廃止されました。
- trigger_fileはpromote_trigger_fileに名前が変更されました。

対処方法

recovery.confに設定していた内容は、postgresql.confに設定してください。

2.3.45 pg_procカタログのprotransform列をprosupport列に変更

非互換

FUJITSU Enterprise Postgres 12では、pg_procカタログのprotransform列をprosupport列に変更します。

対処方法

ありません。

2.3.46 コマンドの出力メッセージのフォーマットの変更

非互換

FUJITSU Enterprise Postgres 12では、コマンド実行時に出力メッセージのフォーマットを変更します。

FUJITSU Enterprise Postgres 11以前の場合

メッセージのエラーラベルを出力しません。

[例]

```
$ pg_basebackup -F invalid
pg_basebackup: invalid output format "invalid", must be "plain" or "tar"
```

FUJITSU Enterprise Postgres 12の場合

メッセージのエラーラベルを出力します。

[例]

```
$ pg_basebackup -F invalid
pg_basebackup: error: invalid output format "invalid", must be "plain" or "tar"
```

対処方法

ありません。

2.3.47 pg_dumpおよびpg_restoreコマンドのデフォルト動作の変更

非互換

FUJITSU Enterprise Postgres 11では、pg_dumpコマンドとpg_restoreコマンドは--createオプション指定時にGRANT/REVOKEやALTER DATABASE SETなどのデータベース属性をダンプ/リストアするようになります。
また、データベースレベルのコメントとセキュリティラベルはデータベース属性として扱われるようになり、--createオプションを指定しない場合はダンプ/リストアされません。

FUJITSU Enterprise Postgres 10以前の場合

- --createオプションを指定する場合
  データベース属性はダンプ/リストアされません。

- --createオプションを指定しない場合
  コメントとセキュリティラベルがダンプ/リストアされます。
  [例]
  $ pg_dump db1
  ・・・
  --
  -- Name: EXTENSION plpgsql; Type: COMMENT; Schema: -; Owner:
  --
  ・・・

FUJITSU Enterprise Postgres 11の場合

- --createオプションを指定する場合
  データベース属性をダンプ/リストアします。
  [例]
  $ pg_dump db1 --create
  ・・・
  ALTER DATABASE db1 SET enable_indexscan TO 'off';
  ・・・

- --createオプションを指定しない場合
  コメントとセキュリティラベルはダンプ/リストアされません。

対処方法

必要に応じて出力された内容を修正してください。

2.3.48 pg_dumpallコマンドのデフォルト動作の変更

非互換

FUJITSU Enterprise Postgres 11では、pg_dumpallコマンドでダンプ取得元の環境のロケールとエンコーディングを指定したCREATE DATBASEコマンド文を出力するようになります。そのため、リストア先の環境にダンプ取得元と同様のロケールやエンコーディングが利用できない場合、リストアに失敗します。

FUJITSU Enterprise Postgres 10以前の場合

ロケールとエンコーディング情報をダンプしません。

[例]

```
$ pg_dumpall
・・・
CREATE DATABASE db1 WITH TEMPLATE = template0 OWNER = foo;
・・・
```

FUJITSU Enterprise Postgres 11の場合

ロケールとエンコーディング情報をダンプします。

[例]

```
$ pg_dumpall
・・・
CREATE DATABASE db1 WITH TEMPLATE = template0 ENCODING = 'UTF8' LC_COLLATE = 'ja_JP.UTF-8' LC_CTYPE = 'ja_JP.UTF-8';
・・・
```

対処方法

必要に応じて出力された内容を修正してください。

2.3.49 pg_dumpallの--globals-onlyオプションの動作の変更

非互換

FUJITSU Enterprise Postgres 11では、pg_dumpallコマンドの--globals-onlyオプションを指定した場合、データベース属性を出力しないように変更します。

FUJITSU Enterprise Postgres 10以前の場合

データベース属性を出力します。

[例]

```
$ pg_dumpall --globals-only
・・・
--
-- Per-Database Role Settings
--

ALTER ROLE foo IN DATABASE db1 SET client_min_messages TO 'debug';
・・・
```

FUJITSU Enterprise Postgres 11の場合

データベース属性を出力しません。

対処方法

データベース属性を出力する場合、“pg_dump --create”コマンドを使用してください。

2.3.50 非一時テーブルに対する一時テーブルのパーティション設定不可

非互換

FUJITSU Enterprise Postgres 11では、非一時テーブルに対して一時テーブルをパーティションとして設定できなくするように変更します。

対処方法

ありません。

2.3.51 関数と列参照のデフォルト動作の変更

非互換

FUJITSU Enterprise Postgres 11では、関数と列参照において、構文形式を優先するように変更します。

FUJITSU Enterprise Postgres 10以前の場合

関数名と同じ名前の列を定義することで、関数が実行されているかのように振る舞うことができます。

[例]

```
=# CREATE TABLE t1 (c1 int);
CREATE TABLE
=# CREATE FUNCTION f1(t1) RETURNS TEXT AS 'BEGIN RETURN ''HELLO, WORLD'';  END; ' LANGUAGE 'plpgsql';
CREATE FUNCTION
=# ALTER TABLE t1 ADD COLUMN f1 int;
ALTER TABLE
=# INSERT INTO t1 VALUES (10,10);
INSERT 0 1
=# SELECT f1(t1) FROM t1;
 f1
----
10
(1 行)
```

FUJITSU Enterprise Postgres 11の場合

関数と同じ名前の列が定義されている場合、構文形式に一致する方を優先します。

[例]

```
=# CREATE TABLE t1 (c1 int);
CREATE TABLE
=# CREATE FUNCTION f1(t1) RETURNS TEXT AS 'BEGIN RETURN ''HELLO, WORLD'';  END; ' LANGUAGE 'plpgsql';
CREATE FUNCTION
=# ALTER TABLE t1 ADD COLUMN f1 int;
ALTER TABLE
=# INSERT INTO t1 VALUES (10,10);
INSERT 0 1
=# SELECT f1(t1) FROM t1;
      f1
--------------
 HELLO, WORLD
(1 行)
```

対処方法

ありません。

2.3.52 テーブルとドメイン制約の命名規約を変更

非互換

FUJITSU Enterprise Postgres 11では、テーブルとドメイン制約において、同じ制約名の指定を許可しないように変更します。

FUJITSU Enterprise Postgres 10以前の場合

同じ制約名の指定が許可される場合があります。

[例]

```
=# CREATE TABLE t1(c1 integer);
CREATE TABLE
=# ALTER TABLE t1 ADD CONSTRAINT const1 CHECK(c1 > 1);
ALTER TABLE
=# ALTER TABLE t1 ADD CONSTRAINT const1 UNIQUE(c1);
ALTER TABLE
```

FUJITSU Enterprise Postgres 11の場合

同じ制約名の指定は許可されません。

[例]

```
=# CREATE TABLE t1(c1 integer);
CREATE TABLE
=# ALTER TABLE t1 ADD CONSTRAINT const1 CHECK(c1 > 1);
```

```
ALTER TABLE
=# ALTER TABLE t1 ADD CONSTRAINT const1 UNIQUE(c1);
ERROR:  すでに制約"const1"はリレーション"t1"に存在します
```

対処方法

ありません。

2.3.53 pg_classカタログからrelhaspkey列を削除

非互換

FUJITSU Enterprise Postgres 11では、pg_classカタログからrelhaspkey列を削除します。

対処方法

プライマリキーであることの確認はpg_indexカタログのindisprimary列を参照してください。

2.3.54 情報スキーマのtablesビューのtable_type列のテーブル種類名の変更

非互換

FUJITSU Enterprise Postgres 11では、情報スキーマのtablesビューのtable_type列が表示する外部テーブルの種類名を“FOREIGN TABLE”から“FOREIGN”に変更します。

FUJITSU Enterprise Postgres 10以前の場合

外部テーブルを“FOREIGN TABLE”と表示します。

[例]

```
=# SELECT table_name, table_type FROM information_schema.tables;
 table_name |  table_type
------------+---------------
 t1         | FOREIGN TABLE
・・・
```

FUJITSU Enterprise Postgres 11の場合

外部テーブルを“FOREIGN”と表示します。

[例]

```
=# SELECT table_name, table_type FROM information_schema.tables;
 table_name | table_type
------------+------------
 t1         | FOREIGN
・・・
```

対処方法

ありません。

2.3.55 pg_procカタログのproisagg列とproiswindow列をprokind列に統合

非互換

FUJITSU Enterprise Postgres 11では、pg_procカタログのproisagg列とproiswindow列を、prokind列に統合します。

対処方法

集約関数、ウィンドウ関数の判断はprokind列を参照してください。

prokind列は各関数を以下のように表します。

- f:通常関数
- p:プロシージャ
- a:集約関数
- w:ウィンドウ関数

2.3.56 非スーパーユーザーによる共有カタログのインデックス再作成の防止

非互換

FUJITSU Enterprise Postgres 11では、非スーパーユーザーはREINDEX DATABASE、REINDEX SCHEMAまたはREINDEX SYSTEMを実行したとき、非スーパーユーザーがカタログの所有権を持っていない場合、共有カタログのインデックス再作成はスキップされます。

FUJITSU Enterprise Postgres 10以前の場合

非スーパーユーザーがREINDEX DATABASE、REINDEX SCHEMAまたはREINDEX SYSTEMを実行したとき、共有カタログのインデックス再作成が行われます。

FUJITSU Enterprise Postgres 11の場合

非スーパーユーザーがREINDEX DATABASE、REINDEX SCHEMAまたはREINDEX SYSTEMを実行したとき、非スーパーユーザーがカタログの所有権を持っていない場合、共有カタログのインデックス再作成はスキップされます。

対処方法

共有カタログのインデックス再作成を行う場合、非スーパーユーザーにカタログの所有権を付与するか、スーパーユーザーでインデックス再作成を実施してください。

2.3.57 psコマンドで出力されるプロセス名をpg_stat_activityに同期

非互換

FUJITSU Enterprise Postgres 11では、psコマンドで出力されるプロセス名をpg_stat_activityで表示される名称に同期するように変更します。

psコマンドで出力されるプロセス名

FUJITSU Enterprise Postgres 10以前	FUJITSU Enterprise Postgres 11
checkpointer process	checkpointer
writer process	background writer
wal writer process	walwriter
autovacuum launcher process	autovacuum launcher
stats collector process	stats collector
bgworker: logical replication launcher	logical replication launcher

対処方法

ありません。

2.3.58 ロジカルレプリケーション時のTRUNCATE文の動作の変更

非互換

FUJITSU Enterprise Postgres 11では、ロジカルレプリケーション時にTRUNCATE文を伝播するように変更します。

FUJITSU Enterprise Postgres 10以前の場合

ロジカルレプリケーション時にTRUNCATE文は伝播しません。

[例]

```
パブリッシャ側で実行
=# TRUNCATE t1;
TRUNCATE TABLE
=# SELECT * FROM t1;
 c1
----
(0 行)

サブスクライバ側で実行
=# SELECT * FROM t1;
 c1
----
1
(1 行)
```

FUJITSU Enterprise Postgres 11の場合

ロジカルレプリケーション時にTRUNCATE文を伝播します。

[例]

```
パブリッシャ側で実行
=# TRUNCATE t1;
TRUNCATE TABLE
=# SELECT * FROM t1;
 c1
----
(0 行)

サブスクライバ側で実行
=# SELECT * FROM t1;
 c1
----
(0 行)
```

対処方法

ALTER PUBLICATION文でTRUNCATE以外の権限を付与するように変更してください。

2.3.59 情報スキーマtriggersのaction_order列、action_reference_old_table列、action_reference_new_table列に格納する値の変更

非互換

FUJITSU Enterprise Postgres 11では、情報スキーマtriggersのaction_order列、action_reference_old_table列およびaction_reference_new_table列に値を格納するように変更します。

対処方法

ありません。

2.3.60 情報スキーマtable_constraintsへのenforced列の追加

非互換

FUJITSU Enterprise Postgres 11では、情報スキーマtable_constraintsにenforced列を追加します。

なお、この値は常にYESを設定します。

対処方法

ありません。

2.3.61 CREATE TABLE文 のLIKE句オプション INCLUDING ALL 指定の動作変更

非互換

FUJITSU Enterprise Postgres 11では、CREATE TABLE文のLIKE句オプションにてINCLUDING ALLを指定した場合、コピー元テーブルの拡張統計情報(STATISTICS)もコピーするように変更します。

FUJITSU Enterprise Postgres 10以前の場合

INCLUDING ALL を指定しても、コピー元テーブルに設定された拡張統計情報(STATISTICS)はコピーしません。

[例]

```
=# ¥d t1
                テーブル "public.t1"
列  |   型    | 照合順序 | Null 値を許容 | デフォルト
-----+---------+----------+---------------+------------
 c1 | integer |          | not null      |
 c2 | integer |          |               |
インデックス:
    "t1_pkey" PRIMARY KEY, btree (c1)
統計オブジェクト:
    "public"."statistics_t1" (ndistinct, dependencies) ON c1, c2 FROM t1
=#
=# CREATE TABLE t2 (LIKE t1 INCLUDING ALL);
CREATE TABLE
=# ¥d t2
                テーブル "public.t2"
列  |   型    | 照合順序 | Null 値を許容 | デフォルト
-----+---------+----------+---------------+------------
 c1 | integer |          | not null      |
 c2 | integer |          |               |
インデックス:
    "t2_pkey" PRIMARY KEY, btree (c1)
```

FUJITSU Enterprise Postgres 11の場合

INCLUDING ALL を指定した場合、コピー元テーブルに設定された拡張統計情報(STATISTICS)もコピーします。

[例]

```
=# ¥d t1
                テーブル "public.t1"
列  |   型    | 照合順序 | Null 値を許容 | デフォルト
-----+---------+----------+---------------+------------
 c1 | integer |          | not null      |
 c2 | integer |          |               |
インデックス:
    "t1_pkey" PRIMARY KEY, btree (c1)
統計オブジェクト:
    "public"."statistics_t1" (ndistinct, dependencies) ON c1, c2 FROM t1
=#
=# CREATE TABLE t2 (LIKE t1 INCLUDING ALL);
CREATE TABLE
=# ¥d t2
                テーブル "public.t2"
列  |   型    | 照合順序 | Null 値を許容 | デフォルト
-----+---------+----------+---------------+------------
 c1 | integer |          | not null      |
 c2 | integer |          |               |
インデックス:
    "t2_pkey" PRIMARY KEY, btree (c1)
統計オブジェクト:
    "public"."t2_c1_c2_stat" (ndistinct, dependencies) ON c1, c2 FROM t2
```

対処方法

テーブルコピー時に拡張統計情報をコピーしたくない場合は、LIKE句オプションにてINCLUDING ALLを指定せずにコピーする属性を個別に指定してください。

2.3.62 postgresql.confファイルのreplacement_sort_tuplesパラメータの削除

非互換

FUJITSU Enterprise Postgres 11では、replacement_sort_tuplesパラメータをpostgresql.confファイルから削除します。

対処方法

ありません。

2.3.63 CREATE FUCNTION 文のWITH句の廃止

非互換

FUJITSU Enterprise Postgres 11では、CREATE FUNCTION文のWITH句を廃止します。

対処方法

FUJITSU Enterprise Postgres 10以前にWITH句で指定していた以下の属性はCREATE FUNCTION 文の以下の既存オプションで対応可能です。

対応オプション

WITH句に指定していた属性	対応オプション
isStrict	STRICT または、 RETURNS NULL ON NULL INPUT
isCachable	IMMUTABLE

2.3.64 EXPLAIN ANALYZE 文でのパラレルワーカーのソート処理情報の追加

非互換

FUJITSU Enterprise Postgres 11では、パラレルクエリが有効の場合、EXPLAIN ANALYZE 文で出力する計測データにパラレルワーカーのソート処理情報を追加します。

FUJITSU Enterprise Postgres 10以前の場合

EXPLAIN ANALYZE 文の出力情報にパラレルワーカーのソート処理情報は出力しません。

[例]

```
=# EXPLAIN ANALYZE SELECT * FROM t1 ORDER BY c1;

                            QUERY PLAN
------------------------------------------------------------------------
Gather Merge  (cost=9204049.69..19925528.06 rows=91891970 width=12) (actual time=112404.558..154479.236 rows=100000000
loops=1)
   Workers Planned: 2
   Workers Launched: 2
   ->  Sort  (cost=9203049.67..9317914.63 rows=45945985 width=12) (actual time=111671.406..115815.382 rows=33333333
loops=3)
         Sort Key: c1
         Sort Method: external sort  Disk: 856432kB
```

```
         ->  Parallel Seq Scan on test3  (cost=0.00..1000000.85 rows=45945985 width=12) (actual time=2.710..93100.395
rows=33

333333 loops=3)
Planning time: 0.160 ms
Execution time: 158725.714 ms
(9 行)

=#
```

FUJITSU Enterprise Postgres 11の場合

EXPLAIN ANALYZE 文の出力情報にパラレルワーカーのソート処理情報(ソート方法、処理サイズ)を出力します。

[例]

```
=# EXPLAIN ANALYZE SELECT * FROM t1 ORDER BY c1;
                           QUERY PLAN
-----------------------------------------------------------------------
Gather Merge  (cost=8367863.07..18090772.59 rows=83333406 width=12) (actual time=212648.534..234706.544 rows=100000000
loops=1)
   Workers Planned: 2
   Workers Launched: 2
   ->  Sort  (cost=8366863.05..8471029.81 rows=41666703 width=12) (actual time=204899.404..208541.511 rows=33333333
loops=3)
         Sort Key: c1
         Sort Method: external merge  Disk: 867128kB
         Worker 0:  Sort Method: external merge  Disk: 853328kB
         Worker 1:  Sort Method: external merge  Disk: 823776kB
         ->  Parallel Seq Scan on test8  (cost=0.00..957208.03 rows=41666703 width=12) (actual time=8.744..135696.844
rows=33333333

 loops=3)
Planning Time: 0.720 ms
Execution Time: 238852.702 ms
(11 行)

=#
```

対処方法

ありません。

2.3.65 プランナが抽出条件「カラム名 = カラム名」を「IS NOT NULL」へ置換する処理に変更

非互換

FUJITSU Enterprise Postgres 11では、EXPLAINに指定するSQL文の条件句に、「id = id」のような「カラム名 = カラム名」を指定した場合、プランナは同一条件句である「IS NOT NOLL」に置換して問い合わせ計画を作成します。

FUJITSU Enterprise Postgres 10以前の場合

EXPLAIN に指定するSQL文の抽出条件に「カラム名 = カラム名」を指定されていても、その条件のまま実行計画を作成します。

[例]

```
=# EXPLAIN SELECT * FROM t1 WHERE c1=c1;
                        QUERY PLAN
-----------------------------------------------------------
Seq Scan on t1  (cost=0.00..38.25 rows=11 width=8)
   Filter: (c1 = c1)
```

FUJITSU Enterprise Postgres 11の場合

EXPLAINに指定するSQL文の抽出条件に「カラム名＝カラム名」を指定されていた場合、「IS NOT NOLL」に置換し、実行計画を作成します。

[例]

```
=# EXPLAIN SELECT * FROM t1 WHERE c1=c1;
                       QUERY PLAN
-------------------------------------------------------
Seq Scan on t1  (cost=0.00..32.60 rows=2249 width=8)
   Filter: (id1 IS NOT NULL)
```

対処方法

ありません。

2.3.66 pg_rewindコマンドがOSのスーパーユーザー(root)で実行不可に変更

非互換

FUJITSU Enterprise Postgres 11では、pg_rewindコマンドをOSのスーパーユーザー(root)で実行できないように変更します。

対処方法

データベースのスーパーユーザー権限を持つユーザーでpg_rewindコマンドを実行してください。

2.3.67 pg_stat_statements_resetの実行がデータベースのスーパーユーザー権限を持つユーザー以外で禁止に変更

非互換

FUJITSU Enterprise Postgres 11では、関数 pg_stat_statements_reset はデータベースのスーパーユーザー権限を持つユーザー以外で実行できないように変更します。

対処方法

ありません。

2.3.68 pg_stat_statementsのqueryid列を64ビットに拡張

非互換

FUJITSU Enterprise Postgres 11では、pg_stat_statementsのqueryid列を32ビットから64ビットに拡張します。

FUJITSU Enterprise Postgres 10以前の場合

整数で表示されます。

[例]

```
=# SELECT queryid, calls, total_time
-# FROM pg_stat_statements
-# ORDER BY total_time DESC LIMIT 5;
  queryid   | calls | total_time
------------+-------+------------
 3317055812 |     1 |  41.729585
 3042908371 |     8 |   13.06031
  650390579 |     1 |  12.098997
 1386568356 |     4 |   5.734614
 3651065577 |     1 |   3.766022
(5 行)
```

FUJITSU Enterprise Postgres 11の場合

queryid列の桁数が増加し、負の整数でも表示されます。

[例]

```
=# SELECT queryid, calls, total_time
-# FROM pg_stat_statements
-# ORDER BY total_time DESC LIMIT 5;
       queryid        | calls | total_time
----------------------+-------+------------
   841746699074173216 |     1 |   1.097363
  4687651466509215193 |     1 |   0.656679
 -7036387361585959152 |     2 |   0.451125
 -7337170263053070648 |     4 |   0.378247
  1077204437590680632 |     1 |   0.152526
(5 行)
```

対処方法

ありません。

2.3.69 ビューに対する SELECT FOR UPDATE の動作変更

非互換

FUJITSU Enterprise Postgres 11では、入れ子になったビューにおいて、親のビューに対してユーザーにUPDATE権限がない場合、子のビューにUPDATE権限があっても、子のビューを通したSELECT FOR UPDATEを実行できないように変更します。

FUJITSU Enterprise Postgres 10以前の場合

正常終了し、SELECT結果が返却されます。

[例]

```
=> SELECT * FROM v1 FOR UPDATE;
 c1 | c2 | c3 | c4
----+----+----+----
  1 | aaa|  2 |  1
(1 行)
```

FUJITSU Enterprise Postgres 11の場合

UPDATE権限がないビューに関して、アクセスが拒否され、エラーとなります。

[例]

```
=> SELECT * FROM v1 FOR UPDATE;
ERROR:  ビュー t1 へのアクセスが拒否されました
```

対処方法

更新対象の全てのビューに対するUPDATE権限をユーザーに付与してください。

・

2.3.70 VACUUM のオプションに列リストを指定する場合の動作変更

非互換

FUJITSU Enterprise Postgres 11では、VACUUMでオプションに列リストを指定する場合、ANALYZEキーワードの指定が必須に変更となります。FUJITSU Enterprise Postgres 10以前では、そのような場合は暗黙にANALYZEが含まれました。

FUJITSU Enterprise Postgres 11では、ANALYZEキーワードを指定しない場合、エラーとなります。

FUJITSU Enterprise Postgres 10以前の場合

正常終了します。

[例]

```
=# VACUUM (FULL) t1(c1);

VACUUM
```

FUJITSU Enterprise Postgres 11の場合

エラーを返します。

[例]

```
=# VACUUM (FULL) t1(c1);

ERROR:  ANALYZE オプションは列リストが与えられているときのみ指定できます
```

対処方法

オプションに列リストを指定する場合、ANALYZEキーワードも指定してください。

2.3.71 システムビューpg_stat_wal_receiverへのsender_host列、sender_port列の追加

非互換

FUJITSU Enterprise Postgres 11では、システムビューpg_stat_wal_receiverにおいて、接続情報のホスト名を示すsender_host列とポート番号を示すsender_port列を追加するよう変更します。

対処方法

ありません。

2.3.72 pg_dumpが出力するCREATE TRIGGER ... REFERENCINGコマンドの遷移テーブル名をダブルクォートで括るように変更

非互換

FUJITSU Enterprise Postgres 11では、pg_dumpコマンドが出力するCREATE TRIGGER ... REFERENCINGコマンドの遷移テーブル名に大文字・小文字が含まれている場合、ダブルクォートで括るように変更します。

FUJITSU Enterprise Postgres 10以前の場合

大文字・小文字を含む遷移テーブル名はダブルクォートで括られません。

[例]

```
$ pg_dump db1

・・・

CREATE TRIGGER tg1 AFTER INSERT ON public."t1_T1" REFERENCING NEW TABLE AS t1_T1 FOR EACH STATEMENT EXECUTE PROCEDURE
public.f1();

・・・
```

FUJITSU Enterprise Postgres 11の場合

大文字・小文字を含む遷移テーブル名はダブルクォートで括られます。

[例]

```
$ pg_dump db1
```

```
・・・

CREATE TRIGGER tg1 AFTER INSERT ON public."t1_T1" REFERENCING NEW TABLE AS "t1_T1" FOR EACH STATEMENT EXECUTE PROCEDURE
public.f1();

・・・
```

対処方法

ありません。

2.3.73 パーティションテーブルに対するCOPYコマンドの動作変更

非互換

FUJITSU Enterprise Postgres 11では、パーティションテーブルに対してCOPYコマンドを実行した場合、パーティションに付与されたCHECK制約を適用するように変更します。

FUJITSU Enterprise Postgres 10以前の場合

パーティションに付与されたCHECK制約は適用されません。

[例]

0を含むデータt1_dataをCOPYする場合

```
=# CREATE TABLE t1 ( c1 int) PARTITION BY RANGE (c1);
CREATE TABLE
=# CREATE TABLE t1_p1 PARTITION OF t1 ( CHECK ( c1 > 1 )) FOR VALUES FROM (0) TO (10);
CREATE TABLE
=# COPY t1 FROM '/tmp/t1_data';
COPY 2
```

FUJITSU Enterprise Postgres 11の場合

パーティションに付与されたCHECK制約を適用します。

[例]

0を含むデータt1_dataをCOPYする場合

```
=# CREATE TABLE t1 ( c1 int) PARTITION BY RANGE (c1);
CREATE TABLE
=# CREATE TABLE t1_p1 PARTITION OF t1 ( CHECK ( c1 > 1 )) FOR VALUES FROM (0) TO (10);
CREATE TABLE
=# COPY t1 FROM '/tmp/t1_data';
ERROR:  リレーション"t1_p1"の新しい行は検査制約"t1_p1_c1_check"に違反しています
DETAIL:  失敗した行は(0)を含みます
CONTEXT:  t1のCOPY、行 1: "0"
```

対処方法

ありません。

2.3.74 論理レプリケーションスロットが存在する場合のサーバ起動可否判定の変更

非互換

FUJITSU Enterprise Postgres 11では、論理レプリケーションスロットが存在する場合、wal_levelにlogical未満の値を指定すると起動できないように変更します。

対処方法

ありません。

2.3.75 1サーバ内での最大接続数の変更

非互換

1サーバでの最大接続数上限を“262,143”から“65,535”に変更します。

対処方法

ありません。

2.3.76 ALTER TABLESPACE文による暗号化設定の変更

非互換

テーブル空間内にデータが格納されていない場合に、ALTER TABLESPACE文により、テーブル空間の暗号化設定を変更できるようになります。

対処方法

ありません。

2.3.77 データ格納先ディレクトリのサブディレクトリの名称変更

非互換

FUJITSU Enterprise Postgres 10では、以下のとおり、データ格納先ディレクトリのサブディレクトリの名称を変更します。

FUJITSU Enterprise Postgres 9.6以前	FUJITSU Enterprise Postgres 10
pg_xlog	pg_wal
pg_clog	pg_xact

対処方法

ありません。

2.3.78 コマンドおよびオプションの名称変更

非互換

FUJITSU Enterprise Postgres 10では、以下のとおり、コマンドおよびコマンドオプションの名称を変更します。

コマンドの名称変更

FUJITSU Enterprise Postgres 9.6以前	FUJITSU Enterprise Postgres 10
pg_receivexlog	pg_receivewal
pg_resetxlog	pg_resetwal
pg_xlogdump	pg_waldump
pgx_xlogcopy.cmd	pgx_walcopy.cmd

コマンドオプションの名称変更

コマンド	FUJITSU Enterprise Postgres 9.6以前	FUJITSU Enterprise Postgres 10
pg_basebackup	--xlogdir	--waldir
	--xlog-method	--wal-method

コマンド	FUJITSU Enterprise Postgres 9.6以前	FUJITSU Enterprise Postgres 10
initdb	--xlogdir	--waldir

対処方法

ありません。

2.3.79 pg_ctlコマンドの操作完了待ちのデフォルト動作の変更

非互換

FUJITSU Enterprise Postgres 10では、pg_ctlコマンドのstartモード、restartモード、promoteモード、およびregisterモードにおいて、デフォルトで操作が完了するのを待つように変更します。

FUJITSU Enterprise Postgres 9.6以前の場合

stopモードは操作の完了を待ちます。startモード、restartモード、promoteモード、およびregisterモードは操作の完了を待ちません。

FUJITSU Enterprise Postgres 10以降の場合

startモード、stopモード、restartモード、promoteモード、およびregisterモードで操作の完了を待ちます。

対処方法

操作の完了を待ちたくない場合は、-Wオプションを指定してpg_ctlコマンドを実行してください。

2.3.80 pg_ctlコマンドのstartモードとrestartモードにおけるタイムアウト時の終了ステータスの変更

非互換

FUJITSU Enterprise Postgres 10では、pg_ctlコマンドのstartモードとrestartモードにおいて、タイムアウトした場合は終了ステータス1を返すように変更します。

FUJITSU Enterprise Postgres 9.6以前の場合

startモード、restartモードでタイムアウトした場合は0を返します。

[例]

```
$ pg_ctl start -w -t 0

サーバの起動完了を待っています... 待機処理が停止されました

サーバは依然起動中です。

$ echo $?

0
```

FUJITSU Enterprise Postgres 10以降の場合

startモード、restartモードでタイムアウトした場合は1を返します。

[例]

```
$ pg_ctl start -w -t 0

サーバの起動完了を待っています... 待機処理が停止されました

pg_ctl: server did not start in time

$ echo $?
```

1

対処方法

ありません。

2.3.81 pg_basebackupコマンドのデフォルト動作の変更と-xオプションの削除

非互換

FUJITSU Enterprise Postgres 10では、デフォルトでWALファイルをバックアップに含めるように変更します。また、-x/--xlogオプションを削除します。

FUJITSU Enterprise Postgres 9.6以前の場合

デフォルトではWALファイルをバックアップに含めません。

FUJITSU Enterprise Postgres 10以降の場合

デフォルトでWALファイルをバックアップに含めます。-X/--wal-methodオプションのデフォルトはstreamであり、WALファイルをストリーミングで収集します。

対処方法

WALファイルをバックアップに含めない場合は、-X/--wal-methodオプションにn/noneを指定してください。-x/--xlogを指定していた場合は、-X/--wal-methodオプションにf/fetchを指定してください。

参考

- pg_basebackupコマンドのオプションの名称が一部変更されています。詳細は“2.3.2 コマンドおよびオプション名の名称変更”を参照してください。

2.3.82 createlangコマンドとdroplangコマンドの削除

非互換

FUJITSU Enterprise Postgres 10では、createlangコマンドとdroplangコマンドを削除します。

対処方法

手続き言語をインストール、またはアンインストールする場合は、CREATE EXTENSION、またはDROP EXTENSIONを使用してください。

2.3.83 クライアントコマンドのsearch_path設定の変更

非互換

FUJITSU Enterprise Postgres 10では、pg_dump、pg_upgrade、vacuumdb、および他のクライアントコマンドにおいて、search_path設定にpg_catalogスキーマのみを含むように変更します。本変更により、クライアントコマンドがエラーになる場合があります。

FUJITSU Enterprise Postgres 9.6以前の場合

クライアントコマンドのスキーマの検索順序はsearch_pathパラメータで設定した順となります。

[例]

```
=# CREATE FUNCTION f1(int) RETURNS int LANGUAGE SQL AS 'SELECT $1 + 1';

CREATE FUNCTION

=# CREATE FUNCTION f2(int) RETURNS int LANGUAGE SQL AS 'SELECT f1($1) + 1';

CREATE FUNCTION
```

```
=# CREATE TABLE t1(c1 int, c2 int, c3 int);
CREATE TABLE
=# CREATE INDEX ix1 ON t1(f2(c1));
CREATE INDEX

$ reindexdb db
```

FUJITSU Enterprise Postgres 10以降の場合

クライアントコマンドのスキーマの検索対象はpg_catalogスキーマのみです。

[例]

```
=# CREATE FUNCTION f1(int) RETURNS int LANGUAGE SQL AS 'SELECT $1 + 1';
CREATE FUNCTION
=# CREATE FUNCTION f2(int) RETURNS int LANGUAGE SQL AS 'SELECT f1($1) + 1';
CREATE FUNCTION
=# CREATE TABLE t1(c1 int, c2 int, c3 int);
CREATE TABLE
=# CREATE INDEX ix1 ON t1(f2(c1));
CREATE INDEX

$ reindex db
reindexdb: データベース"db"の再インデックス化に失敗しました: ERROR:  関数f1(integer) は存在しません (10494)
行 1: SELECT f1($1) + 1
              ^
HINT:  指定名称、指定引数型に合う関数がありません。明示的な型キャストが必要かもしれません (14431)
QUERY:  SELECT f1($1) + 1
CONTEXT:  SQL関数"f2"がインラインになっています (14744)
```

対処方法

クライアントコマンドがエラーになった場合、エラーの原因となったオブジェクトをスキーマ修飾してください。

2.3.84 postgresql.confファイルのパラメータのデフォルト値の変更

非互換

FUJITSU Enterprise Postgres 10では、以下のとおり、postgresql.confファイルのパラメータのデフォルト値を変更します。

パラメータ	FUJITSU Enterprise Postgres 9.6以前	FUJITSU Enterprise Postgres 10
password_encryption	on	md5
max_parallel_workers_per_gather	0	2

パラメータ	FUJITSU Enterprise Postgres 9.6以前	FUJITSU Enterprise Postgres 10
wal_level	minimal	replica
max_wal_senders	0	10
max_replication_slots	0	10
hot_standby	off	on
log_directory	'pg_log'	'log'
log_line_prefix	''	'%m [%p] '

対処方法

必要に応じて、パラメータの設定を変更してください。

2.3.85 postgresql.confファイルのmin_parallel_relation_sizeパラメータの削除とmin_parallel_table_scan_sizeパラメータおよびmin_parallel_index_scan_sizeパラメータの追加

非互換

FUJITSU Enterprise Postgres 10では、postgresql.confファイルから並列検索を制御するmin_parallel_relation_sizeパラメータを削除し、代わりにmin_parallel_table_scan_sizeパラメータとmin_parallel_index_scan_sizeパラメータを追加します。これにより、テーブルとインデックスに分けてパラレルスキャンをチューニングすることができます。

対処方法

min_parallel_relation_sizeパラメータを使用している場合は、代わりにparallel_table_scan_sizeパラメータやmin_parallel_index_scan_sizeパラメータを使用してください。

2.3.86 postgresql.confファイルのsql_inheritanceパラメータの削除

非互換

FUJITSU Enterprise Postgres 10では、テーブル参照が継承された子テーブルを含むか否かを制御するsql_inheritanceパラメータをpostgresql.confファイルから削除します。

対処方法

sql_inheritanceパラメータにoffを指定している場合は、FROM句のテーブル名の前にONLYを指定してください。

2.3.87 postgresql.confファイルのpassword_encryptionパラメータの指定値の変更

非互換

FUJITSU Enterprise Postgres 10では、postgresql.confファイルのpassword_encryptionパラメータに指定可能な値を変更します。

FUJITSU Enterprise Postgres 9.6以前の場合

password_encryptionパラメータに指定可能な値はon、offです。

FUJITSU Enterprise Postgres 10以降の場合

password_encryptionパラメータに指定可能な値はmd5、scram-sha-256、およびonです。

対処方法

ありません。

2.3.88 postgresql.confファイルのshared_preload_librariesパラメータと関連するパラメータ群の大文字小文字区別の変更

非互換

FUJITSU Enterprise Postgres 10では、postgresql.confファイルの以下のパラメータにおいて、ダブルクォーテーションで括られていない文字列を小文字に変換せず、大文字小文字を区別するように変更します。

- local_preload_librariesパラメータ
- session_preload_librariesパラメータ
- shared_preload_librariesパラメータ

FUJITSU Enterprise Postgres 9.6以前の場合

文字列をダブルクォーテーションで括らない場合、大文字を小文字に変換します。このため、大文字小文字は区別されません。

FUJITSU Enterprise Postgres 10以降の場合

文字列をダブルクォーテーションで括らない場合でも、大文字を小文字に変換しません。このため、大文字小文字は区別されます。

対処方法

ファイルパス名が小文字であるにもかかわらず、パラメータを大文字で指定している場合はエラーになります。このため、パラメータはファイルパス名のとおり、小文字で指定してください。

2.3.89 RADIUS認証の設定オプションの名称変更

非互換

FUJITSU Enterprise Postgres 10では、複数のRADIUSサーバに対応するため、以下のとおりRADIUS認証の設定オプションの名称を変更します。

FUJITSU Enterprise Postgres 9.6以前	FUJITSU Enterprise Postgres 10
radiusserver	radiusservers
Radiussecret	radiussecrets
Radiusport	radiusports
Radiusidentifier	radiusidentifiers

対処方法

ありません。

2.3.90 統計情報ビューpg_stat_activityの変更

非互換

FUJITSU Enterprise Postgres 10では、統計情報ビューpg_stat_activityを以下のとおりに変更します。

- backend_type列の追加
- wait_event_type列に格納する待機イベントの追加と変更
- 待機イベントが軽量ロック時のwait_event列に格納される待機イベント名の追加
- 表示対象プロセスの追加

backend_type列の追加

バックエンドの種別を示すbackend_type列を追加します。

wait_event_type列に格納する待機イベントの追加と変更

LWLockNamedとLWLockTrancheをLWLockに変更します。また、新しい待機イベントを追加します。

FUJITSU Enterprise Postgres 9.6以前	FUJITSU Enterprise Postgres 10
LWLockNamed	LWLock
LWLockTranche	
Lock	Lock
BufferPin	BufferPin
	Activity
	Extension
	Client
	IPC
	Timeout
	IO

待機イベントが軽量ロック時のwait_event列に格納される待機イベント名の追加

待機イベントが軽量ロック(LWLock)の場合に、wait_event列に新しい待機イベント名を追加します。

wait_event_type		wait_event列に追加する待機イベント名
FUJITSU Enterprise Postgres 9.6以前	FUJITSU Enterprise Postgres 10	
LWLockNamed	LWLock	BackendRandomLock
		LogicalRepWorkerLock
		CLogTruncationLock
LWLockTranche		parallel_query_dsa
		tbm

表示対象プロセスの追加

補助プロセスやパラレルワーカーなど、クエリを実行するバックエンド以外のプロセスの情報も表示するよう変更します。プロセスの種別はbackend_type列で判別することができます。

[例]

```
=# SELECT pid, state, query, backend_type FROM pg_stat_activity WHERE pid <> pg_backend_pid();

  pid | state  |          query          |    backend_type
------+--------+-------------------------+---------------------
12526 |        |                         | autovacuum launcher
12528 |        |                         | background worker
11544 | active | SELECT max(c2) FROM t1; | client backend
11603 | active | SELECT max(c2) FROM t1; | background worker
11602 | active | SELECT max(c2) FROM t1; | background worker
12524 |        |                         | background writer
12523 |        |                         | checkpointer
12525 |        |                         | walwriter
(8 行)
```

対処方法

クエリを実行するバックエンドのみ表示したい場合は、backend_type列が“client backend”の行のみを検索するようにしてください。

2.3.91 統計情報ビューpg_stat_replicationの変更

非互換

FUJITSU Enterprise Postgres 10では、以下のとおり、統計情報ビューpg_stat_replicationの列名を変更します。また、レプリケーション遅延時間を示す列を追加します。

列名の変更

FUJITSU Enterprise Postgres 9.6以前	FUJITSU Enterprise Postgres 10
sent_location	sent_lsn
write_location	write_lsn
flush_location	flush_lsn
replay_location	replay_lsn

列の追加

列名
write_lag
flush_lag
replay_lag

対処方法

ありません。

2.3.92 システムビューpg_user_mappingsのumoptions列の可視性の厳格化

非互換

FUJITSU Enterprise Postgres 10では、ユーザマッピングオプションとして格納されたパスワード情報を保護するため、システムビューpg_user_mappingsのumoptions列の可視性を厳格化しました。

FUJITSU Enterprise Postgres 9.6以前の場合

外部サーバの所有者、または外部サーバのUSAGE権限を持っている場合は、すべてのユーザマッピングのオプションを参照することを許していました。

[例]

```
=# SET ROLE foo;

SET

=> SELECT * FROM pg_user_mappings;

umid  | srvid | srvname  | umuser | usename |            umoptions
------+-------+----------+--------+---------+---------------------------------
16393 | 16392 | myserver |  16385 | foo     | {user=foo,password=foo000}
16394 | 16392 | myserver |  16386 | bar     | {user=bar,password=bar000}
(2 行)
```

FUJITSU Enterprise Postgres 10以降の場合

以下のいずれかの場合のみ、ユーザマッピングのオプションが参照できます。それ以外の場合はNULLを返却します。

- カレントユーザーが外部サーバにマッピングされているユーザーであり、外部サーバを所有しているか、外部サーバのUSAGE権限を持っている場合
- カレントユーザーが外部サーバの所有者であり、PUBLICとしてマッピングされている場合

ー カレントユーザーがスーパーユーザーの場合

[例]

```
=# SET ROLE foo;

SET

=> SELECT * FROM pg_user_mappings;

umid  | srvid | srvname  | umuser | usename |         umoptions
------+-------+----------+--------+---------+---------------------------
16392 | 16391 | myserver |  16384 | foo     | {user=foo,password=foo000}
16393 | 16391 | myserver |  16385 | bar     |
(2 行)
```

対処方法

ありません。

2.3.93 pg_stat_statementsビューに格納するSQL文の定数の置換方法の変更

非互換

FUJITSU Enterprise Postgres 10では、pg_stat_statementsモジュールで提供するpg_stat_statementsビューのquery列に格納するSQL文の形式を変更し、定数を$1のようなパラメータ記号に置換するようにします。

FUJITSU Enterprise Postgres 9.6以前の場合

SQL文中の定数を?に置換します。

[例]

```
=# SELECT query FROM pg_stat_statements WHERE mean_time > 1000;

                      query
-------------------------------------------------
UPDATE t1 SET c2 =c2+? WHERE c1 BETWEEN ? AND ?;
(1 行)
```

FUJITSU Enterprise Postgres 10以降の場合

SQL文中の定数を$1のようなパラメータ記号に置換します。

[例]

```
=# SELECT query FROM pg_stat_statements WHERE mean_time > 1000;

                       query
-------------------------------------------------
UPDATE t1 SET c2=c2+$1 WHERE c1 BETWEEN $2 AND $3
(1 行)
```

対処方法

ありません。

2.3.94 CREATE ROLE文と同種のコマンド群のUNENCRYPTEDオプションの削除

非互換

FUJITSU Enterprise Postgres 10では、以下のSQL文からUNENCRYPTEDオプションを削除します。

- ALTER ROLE
- ALTER USER

- CREATE GROUP
- CREATE ROLE
- CREATE USER

また、createuserコマンドの-N/--unencryptedオプションも削除します。

対処方法

ありません。

2.3.95 CREATE ROLE文と同種のコマンド群における空パスワード指定時の動作変更

非互換

FUJITSU Enterprise Postgres 10では、以下のSQL文のパスワード設定において、空の文字列を指定した場合はパスワードにNULLを設定するよう変更します。

- ALTER ROLE
- ALTER USER
- CREATE GROUP
- CREATE ROLE
- CREATE USER

createuserコマンドの-P/--pwpromptオプションで空パスワードを指定した場合も同様です。

FUJITSU Enterprise Postgres 9.6以前の場合

空パスワードが設定されます。

[例]

```
=# CREATE ROLE r1 LOGIN PASSWORD '';

CREATE ROLE

=# SELECT rolpassword FROM pg_authid WHERE rolname='r1';

             rolpassword
-------------------------------------
 md57c92cf1eee8d99cc85f8355a3d6e4b86
(1 行)
```

FUJITSU Enterprise Postgres 10以降の場合

NULLが設定されます。

[例]

```
=# CREATE ROLE r1 LOGIN PASSWORD '';

NOTICE:  empty string is not a valid password, clearing password

CREATE ROLE

=# SELECT rolpassword FROM pg_authid WHERE rolname='r1';

 rolpassword
-------------

(1 行)
```

対処方法

ありません。

2.3.96 親テーブルにPRIMARY KEY制約を追加する場合の動作変更

非互換

FUJITSU Enterprise Postgres 10では、親テーブルにPRIMARY KEY制約を追加し、それに伴い、NOT NULL制約が追加される場合、継承の子テーブルにもNOT NULL制約を追加するよう変更します。

FUJITSU Enterprise Postgres 9.6以前の場合

子テーブルにはNOT NULL制約を追加しません。

[例]

```
=# CREATE TABLE t1(c1 int, c2 int not null);

CREATE TABLE

=# CREATE TABLE t2(c3 int) INHERITS(t1);

CREATE TABLE

=# ALTER TABLE t1 ADD PRIMARY KEY(c1);

ALTER TABLE

db=# ¥d t2

  テーブル "public.t2"

 列 |   型    |  修飾語
----+---------+----------
 c1 | integer |
 c2 | integer | not null
 c3 | integer |
継承: t1
```

FUJITSU Enterprise Postgres 10以降の場合

子テーブルにもNOT NULL制約を追加します。

[例]

```
db=# CREATE TABLE t1(c1 int, c2 int not null);

CREATE TABLE

db=# CREATE TABLE t2(c3 int) INHERITS(t1);

CREATE TABLE

db=# ALTER TABLE t1 ADD PRIMARY KEY(c1);

ALTER TABLE

db=# ¥d t2

                テーブル "public.t2"

 列 |   型    | 照合順序 | Null 値を許容 | デフォルト
----+---------+----------+---------------+------------
 c1 | integer |          | not null      |
 c2 | integer |          | not null      |
```

```
 c3 | integer |           |              |
継承元: t1
```

対処方法

子テーブルにNOT NULL制約を付与したくない場合は、ALTER TABLEでNOT NULL制約を削除してください。

2.3.97 レプリケーション接続認証のデフォルト設定の変更

非互換

FUJITSU Enterprise Postgres 10では、pg_hba.confファイルのデフォルト設定を変更することで、pg_hba.confファイルを編集しなくてもローカルホストからのレプリケーションを利用可能にします。

FUJITSU Enterprise Postgres 9.6以前の場合

デフォルトでレプリケーション接続の行をコメントアウトします。

[例]

```
# Allow replication connections from localhost, by a user with the

# replication privilege.

#local   replication     pguser                                 trust

#host    replication     pguser        127.0.0.1/32              trust

#host    replication     pguser        ::1/128                   trust
```

FUJITSU Enterprise Postgres 10以降の場合

デフォルトでローカルホストからのレプリケーション接続を受け付ける設定にします。

[例]

```
# Allow replication connections from localhost, by a user with the

# replication privilege.

local   replication     all                                     trust

host    replication     all             127.0.0.1/32            trust

host    replication     all             ::1/128                 trust
```

対処方法

ローカルホストからのレプリケーション接続を拒否したい場合は、レプリケーション接続の行をコメントアウトしてください。

2.3.98 論理レプリケーションのクライアント認証の設定方法の変更

非互換

FUJITSU Enterprise Postgres 10では、論理レプリケーションのクライアント認証の設定方法を変更し、pg_hba.confファイルのデータベース名に通常のクライアント認証と同じようにデータベース名やallなどのキーワードを指定するようにします。

FUJITSU Enterprise Postgres 9.6以前の場合

pg_hba.confファイルのデータベース名にはreplicationを指定します。

FUJITSU Enterprise Postgres 10以降の場合

pg_hba.confファイルのデータベース名にはデータベース名やallなどのキーワードを指定します。

対処方法

pg_hba.confファイルにおいて論理レプリケーション接続を許可する設定になっているかを確認してください。許可する設定になっていない場合は、論理レプリケーション接続を許可する行を追加してください。

2.3.99 データベース多重化機能によるクラスタシステムのデフォルト構成の変更

非互換

FUJITSU Enterprise Postgres 10では、管理用ネットワークによるハートビート監視がタイムアウトした場合にスプリットブレインを発生させないため、第三者サーバである裁定サーバが必要なシステム構成をデフォルトに変更します。

FUJITSU Enterprise Postgres 9.6以前の場合

2台のデータベースサーバでクラスタシステムを構成します。

FUJITSU Enterprise Postgres 10以降の場合

デフォルトでは、2台のデータベースサーバと裁定サーバによりクラスタシステムを構成します。

対処方法

FUJITSU Enterprise Postgres 9.6以前と同じシステム構成を選択する場合には、OS/サーバの生死監視においてハートビート異常を検出した場合に無条件で自動縮退を行う以下のパラメータをサーバ定義ファイルに設定してください。

- パラメータ名：heartbeat_error_action
- 指定値：fallback

2.3.100 mc_ctlコマンドのオプション省略時のデフォルト動作の変更

非互換

FUJITSU Enterprise Postgres 10では、mc_ctlコマンドの以下のオプションが省略された場合のデフォルト動作を変更します。

- startモード時の-fオプション
- startモード時の-wオプション

FUJITSU Enterprise Postgres 9.6以前の場合

- -fオプションが省略された場合は、Mirroring Controller起動直後に自動切り替え/切り離しの有効化を行いません。
- -wオプションが省略された場合は、操作の完了を待ち合わせません。

FUJITSU Enterprise Postgres 10以降の場合

- -fオプションが省略された場合でも、Mirroring Controller起動直後に自動切り替え/切り離しの有効化を行います。
- -wオプションが省略された場合でも、操作の完了を待ち合わせます。

対処方法

FUJITSU Enterprise Postgres 9.6以前と同じ動作を選択する場合には、mc_ctlコマンドのstartモードを実行する際に以下の対処方法を行ってください。

- -fオプションを省略している場合には、-Fオプションを指定します。
- -wオプションを省略している場合には、-Wオプションを指定します。

2.3.101 Mirroring Controllerがインスタンスに接続する際の接続設定の変更

非互換

FUJITSU Enterprise Postgres 10では、Mirroring Controllerが各データベース要素の障害を検知するためにインスタンスへ接続する際の以下の接続設定を変更します。

・ アプリケーション名

FUJITSU Enterprise Postgres 9.6以前の場合

ー アプリケーション名は、空文字列です。

FUJITSU Enterprise Postgres 10以降の場合

ー アプリケーション名は、'mc_agent'です。

対処方法

アプリケーション名を変更する方法はありません。

そのため、アプリケーション名が空文字列であることによりMirroring Controllerのデータベース接続のセッションであることを特定するようなアプリケーションが存在する場合には、'mc_agent'によって特定するように処理を修正してください。

2.3.102 Mirroring Controllerのサーバの状態表示の変更

非互換

FUJITSU Enterprise Postgres 10では、Mirroring Controllerのmc_ctlコマンドのstatusモードで行うサーバの状態表示について、トランザクションログを受信するプロセス(WAL受信プロセス)を異常と表示する場合の条件を変更します。

・ 該当するサーバの役割(host_role):スタンバイ(standby)

・ 表示項目:DBMSプロセスの状態(db_proc_status)

・ 表示内容:abnormal(異常プロセス名)
異常プロセス名に‘wal_receiver’が含まれるか否かが非互換の対象となります。

FUJITSU Enterprise Postgres 9.6以前の場合

Mirroring ControllerがWAL送信プロセスの停止を検知した場合に表示します。

FUJITSU Enterprise Postgres 10以降の場合

Mirroring ControllerがWAL受信プロセスによるプライマリサーバとのストリーミングレプリケーションの接続を確立できていない状態を検知した場合に表示します。

対処方法

mc_ctlコマンドのstatusモードにより得られた表示内容から、WAL受信プロセスの状態を判断するようなアプリケーションが存在する場合に、従来と同じ精度で判断を行う場合には、以下のように修正してください。ただし、精度向上に伴う非互換のため、推奨はできません。

・ host_roleが'primary'のdb_proc_status項目の'abnormal'に表示される異常プロセス名に'wal_sender'が含まれる場合には、WAL受信プロセスに異常が発生していると判断します。

・ host_roleが'primary'のdb_proc_status項目の'abnormal'に表示される異常プロセス名に'wal_sender'が含まれない場合には、WAL受信プロセスに異常が発生していないと判断します。

2.3.103 データベース多重化運用中のsynchronous_standby_namesパラメータ変更時の動作の変更

非互換

FUJITSU Enterprise Postgres 9.6以前では、データベース多重化運用中にユーザーが誤ってpostgresql.confのsynchronous_standby_namesパラメータを変更した場合に備え、Mirroring Controllerが定期的にデータベースにアクセスしてsynchronous_standby_namesパラメータの値をSHOWコマンドにより取得し、変更が検出された場合には自動で復旧していました。しかし、本処理はデータベースサーバのCPUを冗長に使用し、またSQL文が高頻度で実行されることから、FUJITSU Enterprise Postgres 10では、デフォルトでこれらの処理を行わないようにします。

FUJITSU Enterprise Postgres 9.6以前の場合

postgresql.confのsynchronous_standby_namesパラメータがユーザーの誤操作により変更されたかを、Mirroring Controllerがチェックします。

FUJITSU Enterprise Postgres 10以降の場合

デフォルトでは、postgresql.confのsynchronous_standby_namesパラメータがユーザーの誤操作により変更されたかを、Mirroring Controllerはチェックしません。

対処方法

FUJITSU Enterprise Postgres 9.6以前と同じ動作を選択する場合には、以下のパラメータをサーバ定義ファイルに設定してください。

- パラメータ名:check_synchronous_standby_names_validation
- 指定値:'on'

2.3.104 CREATE ROLE文と同種のコマンド群のCREATEUSERオプションとNOCREATEUSERオプションの削除

非互換

FUJITSU Enterprise Postgres 9.6以降では、以下のSQL文からCREATEUSERオプションとNOCREATEUSERオプションを削除します。

- ALTER ROLE
- ALTER USER
- CREATE GROUP
- CREATE ROLE
- CREATE USER

対処方法

CREATEUSERオプション、またはNOCREATEUSERオプションを指定している場合は、SUPERUSERオプション、またはNOSUPERUSERオプションを指定してください。

2.3.105 pg_で始まるロール名を使用不可にする変更

非互換

FUJITSU Enterprise Postgres 9.6以降では、pg_で始まるロール名は予約済みとして扱い、使用不可にするように変更します。

FUJITSU Enterprise Postgres 9.5 の場合

pg_で始まるロール名でも使用できます。

[例]

```
=# CREATE ROLE pg_dummy_role;

CREATE ROLE
```

FUJITSU Enterprise Postgres 9.6以降の場合

pg_で始まるロール名は使用できません。

[例]

```
=# CREATE ROLE pg_dummy_role;

ERROR:  role name "pg_dummy_role" is reserved

DETAIL:  Role names starting with "pg_" are reserved.
```

対処方法

ありません。

2.3.106 pg_controldataおよび関連箇所におけるNextXIDの表示形式の変更

非互換

FUJITSU Enterprise Postgres 9.6以降では、pg_controldata、および関連箇所におけるNextXIDの表示形式を変更します。

FUJITSU Enterprise Postgres 9.5 の場合

number/number形式で表示します。

[例]

```
$ pg_controldata

・・・

Latest checkpoint's NextXID:          0/1711

・・・
```

FUJITSU Enterprise Postgres 9.6以降の場合

number:number形式で表示します。

[例]

```
$ pg_controldata

・・・

Latest checkpoint's NextXID:          0:1711

・・・
```

対処方法

ありません。

2.3.107 統計情報ビューpg_stat_activityのwaiting列の削除とwait_event_type列とwait_event列の追加

非互換

FUJITSU Enterprise Postgres 9.6以降では、統計情報ビューpg_stat_activityからロック待ちの有無を示すwaiting列を削除し、代わりにロック種別を示すwait_event_type列とwait_event列を追加します。

対処方法

ロック待ちの有無をwaiting列で参照している場合は、代わりにwait_event_type列やwait_event列を参照してください。

2.3.108 pg_amカタログの列の変更

非互換

FUJITSU Enterprise Postgres 9.6以降では、pg_amカタログの列を大幅に変更し、大部分の列を削除します。

対処方法

pg_amカタログでインデックスアクセスメソッドの情報を取得している場合は、システムカタログ情報関数pg_index_column_has_property、pg_index_has_property、pg_indexam_has_propertyを使用してください。

2.3.109 システムビューpg_replication_slotsへのconfirmed_flush_lsn列の追加

非互換

FUJITSU Enterprise Postgres 9.6以降では、システムビューpg_replication_slotsにおいて、データの受信を確認できている論理スロットのアドレスを示すconfirmed_flush_lsn列を追加します。

対処方法

ありません。

2.3.110 システムビューpg_settingsのunit列に格納する値の変更

非互換

FUJITSU Enterprise Postgres 9.6以降では、システムビューpg_settingsにおいて、パラメータがmin_wal_size、またはmax_wal_sizeの場合はunit列にはNULLではなく、値を格納するように変更します。

FUJITSU Enterprise Postgres 9.5 の場合

unit列にはNULLが格納されます。

[例]

```
=# ¥pset null NULL
=# SELECT name, unit FROM pg_settings WHERE name = 'min_wal_size';
     name     | unit
--------------+------
 min_wal_size | NULL
(1 row)


=# SELECT name, unit FROM pg_settings WHERE name = 'max_wal_size';

     name     | unit
--------------+------
 max_wal_size | NULL
(1 row)
```

FUJITSU Enterprise Postgres 9.6以降の場合

unit列にはNULLではなく値が格納されます。

[例]

```
=# SELECT name, unit FROM pg_settings WHERE name = 'min_wal_size';

     name     | unit
--------------+------
 min_wal_size | 16MB
(1 row)


=# SELECT name, unit FROM pg_settings WHERE name = 'max_wal_size';

     name     | unit
--------------+------
 max_wal_size | 16MB
(1 row)
```

対処方法

ありません。

2.3.111 情報スキーマのroutinesビューのresult_cast_character_set_name列名の変更

非互換

FUJITSU Enterprise Postgres 9.6以降では、情報スキーマのroutinesビューのresult_cast_character_set_nameの列名をresult_cast_char_set_nameに変更します。

対処方法

ありません。

2.3.112 update_process_titleパラメータのデフォルト値の変更

非互換

FUJITSU Enterprise Postgres 9.6以降では、Windowsにおいて、update_process_titleパラメータのデフォルト値をonからoffに変更します。

対処方法

必要に応じてupdate_process_titleパラメータの設定を変更してください。

2.3.113 未サポートデータ型に対する秘匿化ポリシー定義の変更

非互換

FUJITSU Enterprise Postgresのデータ秘匿化機能において、未サポートのデータ型である配列型またはtimestamp with timezone型に対して秘匿化ポリシーが定義された場合、定義時にエラーとなるよう変更します。

FUJITSU Enterprise Postgres 9.5 の場合

配列型またはtimestamp with timezone型に対する秘匿化ポリシーの定義は正常終了しますが、ポリシーが定義された表の列へのアクセス時にエラーとなります。

FUJITSU Enterprise Postgres 9.6以降の場合

配列型またはtimestamp with timezone型に対する秘匿化ポリシーの定義時にエラーとなります。

対処方法

FUJITSU Enterprise Postgres 9.5で作成した秘匿化ポリシー定義をFUJITSU Enterprise Postgres 9.6以降で再作成する場合、かつ、未サポートのデータ型(配列型またはtimestamp with timezone型)に対するポリシーが含まれる場合は、以下のいずれかを実施してください。

- 秘匿化ポリシーの秘匿化対象に当該データ型のみが含まれる場合、エラーを無視してください、または、pgx_drop_confidential_policyシステム管理関数により当該ポリシーを削除してください。
- 秘匿化ポリシーの秘匿化対象にサポート対象のデータ型と当該データ型の列の両方が含まれる場合、当該データ型の列をpgx_alter_confidential_policyシステム関数により削除してください。

2.4 JDBCドライバに関する非互換

項目	移行前のバージョン				
	9.5	9.6	10	11	12
targetServerTypeの値の変更	◎	◎	◎	◎	◎
PgDatabaseMetaDataクラスのgetTypeInfoメソッドの返却値の変更	○	○	○	○	○
Java LocalTimeを変換する場合の動作の変更	○	○	○	○	○
PgDatabaseMetaDataクラスの getIndexInfoメソッドの返却値の変更	○	○	○	○	○

項目	移行前のバージョン				
	9.5	9.6	10	11	12
PgDatabaseMetaDataクラスのgetProceduresメソッドの動作の変更	○	○	○	○	○
DatabaseMetaDataクラスのgetColumnsメソッドの返却値の変更	○	○	○	○	○
PgConnectionクラスのgetTypeMapメソッドのエントリが存在しない場合の返却値の変更	○	○	○	○	○
text配列の引用符で囲まれていない要素の返却値の変更	○	○	○	○	○
Connectionクラスのcreate Arrayメソッドが作成する配列の動作の変更	○	○	○	○	○
データベース列の値がNULL、かつJREアサーションが有効な場合の動作の変更	○	○	○	○	○
PgTokenizerクラスの返却値の変更	○	○	○	○	○
gssEncModeパラメータのデフォルト値の変更	○	○	○	○	○
1582年10月の時間を変換時の返却値の変更	○	○	○	○	○
getNotificationsメソッドの返却値の変更	○	○	○	○	○
PgResultSetクラスのrelativeメソッドの返却値の変更	○	○	○	○	○
PGXAConnectionクラスのcommitPreparedメソッドのException処理の変更	○	○	○	○	○
BaseDataSourceクラスのsetUrlメソッドでエラーのURLを指定した場合の動作の変更	○	○	○	○	○
PgConnectionクラスのabortメソッドで引数にNULLを指定した場合の動作の変更	○	○	○	○	○
Driverクラスのconnectメソッドで1つ目の引数にNULLを指定した場合のException処理の変更	○	○	○	○	○
1970年より前かつ時刻を持つタイムスタンプの解析結果の変更	○	○	○	○	○
紀元前の時間の解析結果の変更	○	○	○	○	○
PGCopyInputStreamクラスのreadメソッドおよびReaderInputStreamクラスのreadメソッドの返却値範囲の変更	○	○	○	○	○
PgDatabaseMetaDataクラスのgetFunctionColumnsメソッドの返却値の変更	○	○	○	○	○
検索結果に浮動小数点数のNaNがある場合の返却値の変更	○	○	○	○	○
リソースバンドル取得時のロケールを変更	○	○	○	－	－
PGObject型のgetStringメソッドの返却値の変更	○	○	○	－	－
sslオプションにtrueを指定した場合のsslmodeのデフォルト値の変更	○	○	○	－	－
接続パラメータの削除	○	○	－	－	－
ログの取得方法の変更	○	○	－	－	－
カレンダを使用した場合のgetTimeメソッドの返却値の変更	○	○	－	－	－
24:00:00以上の時刻を入力した場合のgetTimeメソッドの返却値の変更	○	○	－	－	－

項目	移行前のバージョン				
	9.5	9.6	10	11	12
DriverManagerクラスのgetConnectionメソッドのURLへのデータベース名および接続引数の値の指定方法の変更	○	－	－	－	－
connectTimeoutパラメータのデフォルト値の変更	○	－	－	－	－

◎:FUJITSU Enterprise Postgresによる非互換あり

○:OSSによる非互換あり

－:非互換なし

2.4.1 targetServerTypeの値の変更

非互換

FUJITSU Enterprise Postgres 13では、接続文字列に指定するtargetServerTypeの値を変更しました。そのため、以前使用していた値は使用できなくなります。

対処方法

targetServerTypeの値を指定している場合は、以下のように値を変更してください。

ターゲットサーバへの指定値

サーバの選択順	FUJITSU Enterprise Postgres 12以前	FUJITSU Enterprise Postgres 13
プライマリサーバ	master	primary
スタンバイサーバ	slave	secondary
スタンバイサーバの優先	preferSlave	preferSecondary
任意	any	any

2.4.2 PgDatabaseMetaDataクラスのgetTypeInfoメソッドの返却値の変更

非互換

FUJITSU Enterprise Postgres 13では、PgDatabaseMetaDataクラスのgetTypeInfoメソッドの返却値をDATA_TYPEの順に返却します。DATA_TYPEの順に返却するのが正しい仕様ですが、これまではDATA_TYPEの順に返却しない場合がありました。

FUJITSU Enterprise Postgres 12以前の場合

DATA_TYPEの順に返却しない場合があります。

FUJITSU Enterprise Postgres 13の場合

DATA_TYPEの順に返却します。

対処方法

ありません。

2.4.3 Java LocalTimeを変換する場合の動作の変更

非互換

FUJITSU Enterprise Postgres 13では、Java LocalTimeを変換する場合の丸めの動作を変更します。

FUJITSU Enterprise Postgres 12以前の場合

マイクロ秒で丸めを行います。

FUJITSU Enterprise Postgres 13の場合

ナノ秒で丸めを行います。

対処方法

ありません。

2.4.4 PgDatabaseMetaDataクラスのgetIndexInfoメソッドの返却値の変更

非互換

FUJITSU Enterprise Postgres 13では、PgDatabaseMetaDataクラスのgetIndexInfoメソッドの返却値の順序と数を変更します。

FUJITSU Enterprise Postgres 12以前の場合

下記の値を順に返却します。

```
table_cat, table_schem, table_name, non_unique, index_qualifier, index_name, type, ordinal_position, column_name,
asc_or_desc, cardinality, pages, filter_condition
```

FUJITSU Enterprise Postgres 13の場合

下記の値を順に返却します。

```
table_schem, table_name, non_unique, index_qualifier, index_name, type, ordinal_position, column_name, asc_or_desc,
cardinality, pages, filter_condition
```

対処方法

ありません。

2.4.5 PgDatabaseMetaDataクラスのgetProceduresメソッドの動作の変更

非互換

FUJITSU Enterprise Postgres 13では、PgDatabaseMetaDataクラスのgetProceduresメソッドの動作を変更します。

FUJITSU Enterprise Postgres 12以前の場合

プロシージャとファンクションに関する記述を取得します。

FUJITSU Enterprise Postgres 13の場合

プロシージャに関する記述を取得します。

対処方法

ファンクションに関する記述を取得する場合は、getFunctionsメソッドを使用するようにアプリケーションを変更してください。

2.4.6 DatabaseMetaDataクラスのgetColumnsメソッドの返却値の変更

非互換

FUJITSU Enterprise Postgres 13では、DatabaseMetaDataクラスのgetColumnsメソッドの返却値を変更します。

FUJITSU Enterprise Postgres 12以前の場合

NULL列のデータの場合、0を返却します。

FUJITSU Enterprise Postgres 13の場合

NULL列のデータの場合、NULLを返却します。

対処方法

DatabaseMetaDataクラスのgetColumnsメソッドを使用している場合は、NULLが返却されることを考慮してアプリケーションを修正してください。

2.4.7 PgConnectionクラスのgetTypeMapメソッドでエントリが存在しない場合の返却値の変更

非互換

FUJITSU Enterprise Postgres 13では、PgConnectionクラスのgetTypeMapメソッドでエントリが存在しない場合、空のTypeMapを返却します。空のTypeMapを返却するのが仕様ですが、NULLを返却する場合がありました。

FUJITSU Enterprise Postgres 12以前の場合

NULLを返却します。

FUJITSU Enterprise Postgres 13の場合

空のTypeMapを返却します。

対処方法

PgConnectionクラスのgetTypeMapメソッドを使用している場合は、空のTypeMapが返却されることを考慮してアプリケーションを修正してください。

2.4.8 text配列の引用符で囲まれていない要素の返却値の変更

非互換

FUJITSU Enterprise Postgres 13では、text配列の引用符で囲まれていない要素の返却値を変更します。

[例]下記のSQL文を処理します。

```
insert into test_strings (strings)  values (U&'{a¥2001b, a b}'::varchar[]);
select strings from test_strings;
```

FUJITSU Enterprise Postgres 12以前の場合

[ab, a b]を返却します。

FUJITSU Enterprise Postgres 13の場合

[a　b,""a b""]を返却します。

対処方法

ありません。

2.4.9 ConnectionクラスのcreateArrayメソッドが作成する配列の動作の変更

非互換

FUJITSU Enterprise Postgres 13では、ConnectionクラスのcreateArrayメソッドが作成する配列を、1ベースに修正します。1ベースが正しい仕様ですが、これまでは0ベースで作成していました。

[例]下記を処理します。

```
con.createArray（ "varchar"、new String [] {"a"、 "b"、 "c"}）
```

FUJITSU Enterprise Postgres 12以前の場合

0ベースで使用します。

FUJITSU Enterprise Postgres 13の場合

1ベースで使用します。

対処方法

Connectionクラスの createArrayメソッドを使用して配列を使用している場合は、1ベースの配列として使用するようにアプリケーションを修正してください。

2.4.10 データベース列の値がNULL、かつ、JREアサーションが有効な場合の動作の変更

非互換

FUJITSU Enterprise Postgres 13では、データベース列の値がNULL、かつ、JREアサーションが有効な場合の下記のクラスの動作を変更します。

PGbox、PGcircle、PGline、PGlseg、PGpath、PGpoint、PGpolygon、PGmoney

FUJITSU Enterprise Postgres 12以前の場合

不定な値を返却する場合があります。

FUJITSU Enterprise Postgres 13の場合

NULLを返却します。

対処方法

ありません。

2.4.11 PgTokenizerクラスの返却値の変更

非互換

FUJITSU Enterprise Postgres 13では、最後のトークンがNULL、または、空白の場合のPgTokenizerクラスの返却値を変更します。

[例] (1,2EC1830300027,1,,)の場合

FUJITSU Enterprise Postgres 12以前の場合

```
0 = 1
1 = 2EC1830300027
2 = 1
3 =
```

FUJITSU Enterprise Postgres 13の場合

```
0 = 1
1 = 2EC1830300027
2 = 1
3 =
4 =
```

対処方法

ありません。

2.4.12 gssEncModeパラメータのデフォルト値の変更

非互換

FUJITSU Enterprise Postgres 13以降では、gssEncModeパラメータのデフォルト値をPREFERからALLOWに変更します。

対処方法

必要に応じてgssEncModeパラメータの設定を変更してください。

2.4.13 1582年10月の時間を変換時の返却値の変更

非互換

FUJITSU Enterprise Postgres 12では、1582年10月の時間の変換処理の返却値を変更します。

[例]

下記のSQL文を処理します。

```
SELECT * FROM generate_series('1582-09-30 00:00'::timestamp, '1582-10-2 00:00'::timestamp, '1 day');
```

FUJITSU Enterprise Postgres 11以前の場合

例のSQL文の処理結果:

1582-10-10 00:00:00

1582-10-11 00:00:00

1582-10-12 00:00:00

FUJITSU Enterprise Postgres 12の場合

例のSQL文の処理結果:

1582-09-30 00:00:00

1582-10-01 00:00:00

1582-10-02 00:00:00

対処方法

ありません。

2.4.14 getNotificationsメソッドの返却値の変更

非互換

FUJITSU Enterprise Postgres 12では、getNotificationsメソッドで取得できるnotificationの範囲を変更します。

FUJITSU Enterprise Postgres 11以前の場合

1秒前のnotificationだけを取得します。

FUJITSU Enterprise Postgres 12の場合

すべてのnotificationを取得します。

対処方法

ありません。

2.4.15 PgResultSetクラスのrelativeメソッドの返却値の変更

非互換

FUJITSU Enterprise Postgres 12では、下記の条件を満たす場合、PgResultSetクラスのrelativeメソッドの返却値を変更します。

- 引数に負の値を指定した場合
- 現在の位置から相対位置に移動して、先頭行を超えた場合

FUJITSU Enterprise Postgres 11以前の場合

trueを返却します。

FUJITSU Enterprise Postgres 12の場合

falseを返却します。

対処方法

ありません。

2.4.16 PGXAConnectionクラスのcommitPreparedメソッドのException処理の変更

非互換

FUJITSU Enterprise Postgres 12では、GXAConnectionクラスのcommitPreparedメソッドのException処理の変更があります。

FUJITSU Enterprise Postgres 11以前の場合

IllegalArgumentExceptionを投げます。

［例］

```
java.lang.IllegalArgumentException: Unmatched braces in the pattern.
```

FUJITSU Enterprise Postgres 12の場合

PGXAExceptionを投げます。

［例］

```
PGXAException: Not implemented: 2nd phase commit must be issued using an idle connection. commit xid={0},
currentXid={1}, state={2}, transactionState={3}
```

対処方法

ありません。

2.4.17 BaseDataSourceクラスのsetUrlメソッドでエラーのURLを指定した場合の動作の変更

非互換

FUJITSU Enterprise Postgres 12では、BaseDataSourceクラスのsetUrlメソッドでエラーのURLを指定した場合、動作を変更します。

FUJITSU Enterprise Postgres 11以前の場合

NullPointerExceptionを投げます。

［例］

```
Exception in thread "main" java.lang.NullPointerException
```

FUJITSU Enterprise Postgres 12の場合

IllegalArgumentExceptionを投げます。

［例］

```
Exception in thread "main" java.lang.IllegalArgumentException: URL invalid ur/情報
```

対処方法

ありません。

2.4.18 PgConnectionクラスのabortメソッドで引数にNULLを指定した場合の動作の変更

非互換

FUJITSU Enterprise Postgres 12では、PgConnectionクラスのabortメソッドの引数にNULLを指定した場合、動作を変更します。

FUJITSU Enterprise Postgres 11以前の場合

abortしてコネクションがクローズして、後の操作を実施する時に、Exceptionを投げます。

FUJITSU Enterprise Postgres 12の場合

abortする時点でExceptionを投げます。

対処方法

ありません。

2.4.19 Driverクラスのconnectメソッドで1つ目の引数にNULLを指定した場合のException処理の変更

非互換

FUJITSU Enterprise Postgres 12では、Driverクラスのconnectメソッドの1つ目の引数にNULLを指定した場合、Exception処理を変更します。

FUJITSU Enterprise Postgres 11以前の場合

NullPointerExceptionを投げます。

［例］

```
Exception in thread "main" java.lang.NullPointerException
```

FUJITSU Enterprise Postgres 12の場合

SQLExceptionを投げます。

［例］

```
Exception in thread "main" java.sql.SQLException: url is null
```

対処方法

ありません。

2.4.20 1970年より前かつ時刻を持つタイムスタンプの解析結果の変更

非互換

FUJITSU Enterprise Postgres 12では、1970年より前かつ時刻を持つタイムスタンプの解析結果を変更します。

FUJITSU Enterprise Postgres 11以前の場合

［例］

「1969-01-10 12:00:00」の時間を解析して、「1969-01-11」を返却します。

FUJITSU Enterprise Postgres 12の場合

［例］

「1969-01-10 12:00:00」の時間を解析して、「1969-01-10」を返却します。

対処方法

ありません。

2.4.21 紀元前の時間の解析結果の変更

非互換

FUJITSU Enterprise Postgres 12では、紀元前の時間の解析結果を変更します。

FUJITSU Enterprise Postgres 11以前の場合

［例］

下記のソースより、紀元前1997-06-30の時間を解析して、「1996-06-30」を返却します。

```
LocalDate data = LocalDate.parse ("-1996-06-30");
```

FUJITSU Enterprise Postgres 12の場合

［例］

下記のソースより、紀元前1997-06-30の時間を解析して、「1997-06-30」を返却します。

```
LocalDate data = LocalDate.parse ("-1996-06-30");
```

対処方法

ありません。

2.4.22 PGCopyInputStreamクラスのreadメソッドおよびReaderInputStreamクラスのreadメソッドの返却値範囲の変更

非互換

FUJITSU Enterprise Postgres 12では、PGCopyInputStreamクラスのreadメソッドおよびReaderInputStreamクラスのreadメソッドの返却値範囲を変更します。

FUJITSU Enterprise Postgres 11以前の場合

PGCopyInputStreamクラスのreadメソッドおよびReaderInputStreamクラスのreadメソッドの返却値の範囲は4294967168～4294967295です。

FUJITSU Enterprise Postgres 12の場合

PGCopyInputStreamクラスのreadメソッドおよびReaderInputStreamクラスのreadメソッドの返却値の範囲は128～255です。

対処方法

ありません。

2.4.23 PgDatabaseMetaDataクラスのgetFunctionColumnsメソッドの返却値の変更

非互換

FUJITSU Enterprise Postgres 12では、PgDatabaseMetaDataクラスのgetFunctionColumnsメソッドで取得できるResultSet型の返却値を変更します。

ResultSet中の各行には、フィールド名と取得データの型の変更があります。

変更詳細について、下記の表を参照してください。

FUJITSU Enterprise Postgres 11以前		FUJITSU Enterprise Postgres 12	
フィールド名	取得データの型	フィールド名	取得データの型
PROCEDURE_CAT	Oid.VARCHAR	FUNCTION_CAT	Oid.VARCHAR
PROCEDURE_SCHEM	Oid.VARCHAR	FUNCTION_SCHEM	Oid.VARCHAR
PROCEDURE_NAME	Oid.VARCHAR	FUNCTION_NAME	Oid.VARCHAR
PRECISION	Oid.INT4	PRECISION	Oid.INT2
COLUMN_DEF	Oid.VARCHAR	削除	－
SQL_DATA_TYPE	Oid.INT4	削除	－

FUJITSU Enterprise Postgres 11以前		FUJITSU Enterprise Postgres 12	
SQL_DATETIME_SUB	Oid.INT4	削除	－

対処方法

ありません。

2.4.24 検索結果に浮動小数点数のNaNがある場合の返却値の変更

非互換

FUJITSU Enterprise Postgres 12では、下記の浮動小数点数のNaNを検索できる場合、返却値を変更します。

FUJITSU Enterprise Postgres 11以前	FUJITSU Enterprise Postgres 12
'NaN'::real	Float.NaN
'NaN'::double precision	Double.NaN
'NaN'::numeric	Double.NaN

対処方法

ありません。

2.4.25 リソースバンドル取得時のロケールを変更

非互換

FUJITSU Enterprise Postgres 11では、リソースバンドル取得時のロケールをデフォルトロケールからLocale.Category.DISPLAYで取得されるロケールに変更します。

対処方法

必要に応じてプロパティファイル名をLocale.Category.DISPLAYの値に合わせて変更してください。

2.4.26 PGObject型のgetStringメソッドの返却値の変更

非互換

FUJITSU Enterprise Postgres 11では、PGObject型のgetStringメソッドで取得できる値がnullの場合、nullを返却するように変更します。

FUJITSU Enterprise Postgres 10以前の場合

PGObject型のgetStringメソッドで取得できる値がnullの場合、“null”文字列を返却します。

FUJITSU Enterprise Postgres 11の場合

PGObject型のgetStringメソッドで取得できる値がnullの場合、nullを返却します。

対処方法

ありません。

2.4.27 sslオプションにtrueを指定した場合のsslmodeのデフォルト値の変更

非互換

FUJITSU Enterprise Postgres 11では、sslオプションにtrueを指定した場合のsslmodeのデフォルト値を変更します。

FUJITSU Enterprise Postgres 10以前の場合

sslオプションにtrueを指定した場合、sslmodeのデフォルト値はpreferです。

FUJITSU Enterprise Postgres 11の場合

sslオプションにtrueを指定した場合、sslmodeのデフォルト値はverify-fullです。

対処方法

必要に応じてsslmodeオプションにpreferを設定してください。

2.4.28 接続パラメータの削除

非互換

FUJITSU Enterprise Postgres 10では、以下の接続パラメータを削除します。

- compatibleパラメータ
- charSetパラメータ

上記パラメータの削除に伴い、PGConnectionPoolDataSourceクラスとPGXADataSourceクラスから以下のメソッドを削除します。

- getCharset
- setCharset
- getCompatible
- setCompatible

対処方法

ありません。

2.4.29 ログの取得方法の変更

非互換

FUJITSU Enterprise Postgres 10では、以下のとおり、ログの取得方法を変更します。

FUJITSU Enterprise Postgres 9.6以前の場合

loglevelパラメータでログの出力レベルを指定します。PGConnectionPoolDataSourceクラス、またはPGXADataSourceクラスを使用する場合は、setLogLevelメソッドでログの出力レベルを指定することも可能です。ログの出力先はsetLogWriterメソッドで指定します。

FUJITSU Enterprise Postgres 10以降の場合

loggerLevelパラメータ、およびloggerFileパラメータでログの出力レベルと出力先を指定します。PGConnectionPoolDataSourceクラス、またはPGXADataSourceクラスを使用する場合は、setLoggerLevelメソッド、およびsetLoggerFileメソッドでログの出力レベルと出力先を指定することも可能です。

対処方法

ありません。

2.4.30 カレンダを使用した場合のgetTimeメソッドの返却値の変更

非互換

FUJITSU Enterprise Postgres 10では、TimeクラスまたはTimestampクラスのgetTimeメソッドにおいて、ResultSetクラスのgetTimeメソッドまたはgetTimestampメソッドにカレンダを指定している場合は、カレンダのタイムゾーンを使用せずに算出したミリ秒数を返すよう変更します。

FUJITSU Enterprise Postgres 9.6以前の場合

カレンダのタイムゾーンを使用して算出したミリ秒数を返す場合があります。

FUJITSU Enterprise Postgres 10以降の場合

カレンダのタイムゾーンを使用せずに算出したミリ秒数を返します。

対処方法

ありません。

2.4.31 24:00:00以上の時刻を入力した場合のgetTimeメソッドの返却値の変更

非互換

FUJITSU Enterprise Postgres 10では、TimeクラスまたはTimestampクラスのgetTimeメソッドにおいて、ResultSetクラスのgetTimeメソッドまたはgetTimestampメソッドで24:00:00以上の時刻を入力した場合の返却値を変更します。

FUJITSU Enterprise Postgres 9.6以前の場合

- ResultSetクラスのgetTimeメソッドまたはgetTimestampメソッドの取得対象のデータ型がtime型またはtime with timezone型の場合
- 取得したデータが24:00:00の場合、TimeクラスまたはTimestampクラスのgetTimeメソッドは0ミリ秒を返します。
- ResultSetクラスのgetTimeメソッドの取得対象のデータ型がtimestamp型の場合
- TimeクラスのgetTimeメソッドは、日付を除いた時刻部分だけから算出したミリ秒数を返します。時刻が24:00:00の場合は0ミリ秒を返します。
- ResultSetクラスのgetTimeメソッドの取得対象のデータ型が文字列型の場合
- 時刻データが24:00:00の場合、TimeクラスのgetTimeメソッドは0ミリ秒を返します。以降、24時間周期で0ミリ秒に戻った値を返します。
- タイムスタンプデータの場合、TimeクラスのgetTimeメソッドは日付を除いた時刻部分だけから算出したミリ秒数を返します。算出方法は時刻データと同じです。

FUJITSU Enterprise Postgres 10以降の場合

- ResultSetクラスのgetTimeメソッドまたはgetTimestampメソッドの取得対象のデータ型がtime型またはtime with timezone型の場合
- 取得したデータが24:00:00の場合、TimeクラスまたはTimestampクラスのgetTimeメソッドは86400000ミリ秒を返します。
- ResultSetクラスのgetTimeメソッドの取得対象のデータ型がtimestamp型の場合
- TimeクラスのgetTimeメソッドは、1970-01-01 00:00:00からのミリ秒数を返します。
- ResultSetクラスのgetTimeメソッドの取得対象のデータ型が文字列型の場合
- TimeクラスのgetTimeメソッドは、時刻データの場合は00:00:00からのミリ秒数、タイムスタンプデータの場合は1970-01-01 00:00:00からのミリ秒数を返します。
-

対処方法

ありません。

2.4.32 DriverManagerクラスのgetConnectionメソッドのURLへのデータベース名および接続引数の値の指定方法の変更

非互換

FUJITSU Enterprise Postgres 9.6では、データベースの接続を試みるDriverManagerクラスのgetConnectionメソッドにおいて、データベース名および接続引数をURLに指定する際、一部の値をそのまま指定することができなくなります。

[メソッド]

getConnection(String url)

注) 以下のメソッドにより、url以外の引数に指定する場合は影響ありません。

- getConnection(String url, Properties info)
- getConnection(String url, String user, String password)

[接続引数]

- データベース名
- "user"プロパティ
- "password"プロパティ
- その他、任意のString型の値を指定するプロパティ

[値]

- %
- +

FUJITSU Enterprise Postgres 9.5以前の場合

getConnectionメソッドの引数であるurlに[値]を含む[接続引数]をそのまま指定することができます。

[例]

```
String url = "jdbc:postgresql://sv1/test?user=fsepuser
               &password=sec+ret";
Connection conn = DriverManager.getConnection(url);
```

FUJITSU Enterprise Postgres 9.6以降の場合

getConnectionメソッドの引数であるurlに[値]を含む[接続引数]を指定する場合には、URLエンコードする必要があります。

[例]

```
String url = "jdbc:postgresql://sv1/test?user=fsepuser&password="
             + URLEncoder.encode("sec+ret","UTF-8");
Connection conn = DriverManager.getConnection(url);
```

対処方法

以下の何れかの方法で、アプリケーションを修正してください。

- 接続引数の値をURLEncoderクラスのencodeメソッドによりURLエンコードする。
- 接続引数をPropertiesクラスのsetPropertyメソッドによりPropertiesオブジェクトに設定し、それをgetConnectionメソッドの引数に指定する。
- "user"プロパティまたは"password"プロパティの場合は、getConnectionメソッドの引数に指定する。

2.4.33 connectTimeoutパラメータのデフォルト値の変更

非互換

FUJITSU Enterprise Postgres 9.6以降では、connectTimeoutパラメータのデフォルト値を0から10に変更します。

対処方法

必要に応じてconnectTimeoutパラメータの設定を変更してください。

2.5 ODBCドライバに関する非互換

項目	移行前のバージョン				
	9.5	9.6	10	11	12
SQLGetDescField関数の変更	○	○	○	○	－
SQLDescribeCol関数で取得するカラムの属性の変更	○	○	○	○	－

項目	移行前のバージョン				
	9.5	9.6	10	11	12
PGAPI_GetInfo関数の引数rgbInfoValueの戻り値の変更	○	○	○	○	－
クライアントのロケール選定方法の変更	○	○	○	－	－
SQLTables関数およびSQLTablePrivileges関数でパーティションテーブルを表示するよう変更	○	○	○	－	－
存在しないテーブルを指定した場合のSQLSpecialColumns関数の動作変更	○	○	○	－	－
ODBCドライバのデフォルト値の変更	○	○	－	－	－
クライアント符号化方式のデフォルト変更	○	○	－	－	－
UTF-8の1文字あたりの最大バイト数の変更	○	○	－	－	－
通信リンク障害発生時のSQLSTATEの変更	○	○	－	－	－

◎:FUJITSU Enterprise Postgresによる非互換あり

○:OSSによる非互換あり

－:非互換なし

2.5.1 SQLGetDescField関数の変更

非互換

FUJITSU Enterprise Postgres 12では、SQLGetDescField(arg1,arg2,SQL_DESC_COUNT,arg4,arg5,arg6)関数の正常終了した場合の戻り値と、6番目のパラメータの戻り値を変更します。

FUJITSU Enterprise Postgres 11以前の場合

- 関数の戻り値は「SQL_IS_SMALLINT(-8)」になります。
- 6番目のパラメータの戻り値は「sizeof(SQLINTEGER)」になります。

FUJITSU Enterprise Postgres 12の場合

- 関数の戻り値は「SQL_SUCCESS(0)」になります。
- 6番目のパラメータの戻り値は「sizeof(SQLSMALLINT)」になります。

対処方法

ありません。

2.5.2 SQLDescribeCol関数で取得するカラムの属性の変更

非互換

FUJITSU Enterprise Postgres 12では、SQLDescribeCol関数で取得するカラムの属性を小文字から大文字に変更します。

対処方法

ありません。

2.5.3 PGAPI_GetInfo関数の引数rgbInfoValueの戻り値の変更

非互換

FUJITSU Enterprise Postgres 12では、PGAPI_GetInfo関数の引数rgbInfoValueの戻り値を変更します。

FUJITSU Enterprise Postgres 11以前の場合

- SQL_TIMEDATE_FUNCTIONSの場合、戻り値は(SQL_FN_TD_NOW)です。
- SQL_LIKE_ESCAPE_CLAUSEの場合、戻り値は"N"です。
- SQL_NUMERIC_FUNCTIONSの場合、戻り値は0です。
- SQL_STRING_FUNCTIONSの場合、戻り値は以下のようになります。

  SQL_FN_STR_CONCAT |
  SQL_FN_STR_LCASE |
  SQL_FN_STR_LENGTH |
  SQL_FN_STR_LOCATE |
  SQL_FN_STR_LTRIM |
  SQL_FN_STR_RTRIM |
  SQL_FN_STR_SUBSTRING |
  SQL_FN_STR_UCASE
- SQL_SYSTEM_FUNCTIONSの場合、戻り値は0です。

FUJITSU Enterprise Postgres 12の場合

- SQL_TIMEDATE_FUNCTIONSの場合、戻り値は以下のようになります。

  SQL_FN_TD_NOW | SQL_FN_TD_CURRENT_DATE | SQL_FN_TD_CURRENT_TIME
  | SQL_FN_TD_CURRENT_TIMESTAMP | SQL_FN_TD_CURDATE | SQL_FN_TD_CURTIME
  | SQL_FN_TD_DAYNAME | SQL_FN_TD_DAYOFMONTH | SQL_FN_TD_DAYOFWEEK
  | SQL_FN_TD_DAYOFYEAR | SQL_FN_TD_HOUR | SQL_FN_TD_MINUTE
  | SQL_FN_TD_MONTH | SQL_FN_TD_MONTHNAME | SQL_FN_TD_NOW
  | SQL_FN_TD_QUARTER | SQL_FN_TD_SECOND | SQL_FN_TD_WEEK | SQL_FN_TD_YEAR
- SQL_LIKE_ESCAPE_CLAUSEの場合、戻り値は"Y"になります。
- SQL_NUMERIC_FUNCTIONSの場合、戻り値は以下のようになります。

  SQL_FN_NUM_ABS | SQL_FN_NUM_ACOS | SQL_FN_NUM_ASIN | SQL_FN_NUM_ATAN |
  SQL_FN_NUM_ATAN2 | SQL_FN_NUM_CEILING | SQL_FN_NUM_COS | SQL_FN_NUM_COT |
  SQL_FN_NUM_DEGREES | SQL_FN_NUM_EXP | SQL_FN_NUM_FLOOR | SQL_FN_NUM_LOG |
  SQL_FN_NUM_LOG10 | SQL_FN_NUM_MOD | SQL_FN_NUM_PI | SQL_FN_NUM_POWER |
  SQL_FN_NUM_RADIANS | SQL_FN_NUM_RAND | SQL_FN_NUM_ROUND | SQL_FN_NUM_SIGN |
  SQL_FN_NUM_SIN | SQL_FN_NUM_SQRT | SQL_FN_NUM_TAN | SQL_FN_NUM_TRUNCATE
- SQL_STRING_FUNCTIONSの場合、戻り値は以下のようになります。

  SQL_FN_STR_ASCII | SQL_FN_STR_CHAR | SQL_FN_STR_CONCAT |
  SQL_FN_STR_INSERT | SQL_FN_STR_LCASE | SQL_FN_STR_LEFT |
  SQL_FN_STR_LENGTH | SQL_FN_STR_LOCATE | SQL_FN_STR_LOCATE_2 |
  SQL_FN_STR_LTRIM | SQL_FN_STR_REPEAT | SQL_FN_STR_RIGHT |
  SQL_FN_STR_RTRIM | SQL_FN_STR_SPACE | SQL_FN_STR_SUBSTRING | SQL_FN_STR_UCASE
- SQL_SYSTEM_FUNCTIONSの場合、戻り値は以下のようになります。

  SQL_FN_SYS_IFNULL | SQL_FN_SYS_USERNAME

対処方法

ありません。

2.5.4 クライアントのロケール選定方法の変更

非互換

FUJITSU Enterprise Postgres 11では、ODBCを使用したクライアントのカレントロケール設定において、環境変数のロケール設定カテゴリにLC_ALLが設定されていない場合、LC_CTYPE、LANGの順に参照して設定するように変更します。

FUJITSU Enterprise Postgres 10以前の場合

LC_ALLが設定されていない場合、エンコーディングをSQL_ASCIIに設定します。

FUJITSU Enterprise Postgres 11の場合

LC_ALLが設定されていない場合、LC_CTYPE、LANGの順にロケール設定を参照して設定します。

LC_CTYPE、LANGが設定されていない場合、エンコーディングをSQL_ASCIIに設定します。

対処方法

SQL_ASCIIを設定する場合、環境変数PGCLIENTENCODINGにSQL_ASCIIを設定してください。

2.5.5 SQLTables関数およびSQLTablePrivileges関数でパーティションテーブルを表示するよう変更

非互換

FUJITSU Enterprise Postgres 11では、SQLTables関数およびSQLTablePrivileges関数を使用時において、パーティションテーブルを表示するよう変更します。

FUJITSU Enterprise Postgres 10以前の場合

SQLTables関数およびSQLTablePrivileges関数を使用時において、パーティションテーブルは表示しません。

FUJITSU Enterprise Postgres 11の場合

SQLTables関数およびSQLTablePrivileges関数を使用時において、パーティションテーブルを表示します。

対処方法

ありません。

2.5.6 存在しないテーブルを指定した場合のSQLSpecialColumns関数の動作変更

非互換

FUJITSU Enterprise Postgres 11では、存在しないテーブルを指定してSQLSpecialColumns関数を実行した場合、エラーとなるように変更します。

FUJITSU Enterprise Postgres 10以前の場合

存在しないテーブルを指定してSQLSpecialColumns関数を実行した場合、oid/xminを返します。

FUJITSU Enterprise Postgres 11の場合

存在しないテーブルを指定してSQLSpecialColumns関数を実行した場合、エラーとなります。

対処方法

ありません。

2.5.7 ODBCドライバのデフォルト値の変更

非互換

FUJITSU Enterprise Postgres 10では、以下のとおり、ODBCドライバのオプションのデフォルト値を変更します。

オプション	FUJITSU Enterprise Postgres 9.6以前	FUJITSU Enterprise Postgres 10
ExtraSysTablePrefixes	dd_	-(空)
ByteaAsLongVarBinary	0	1

対処方法

必要に応じて、オプションの設定を変更してください。

2.5.8 クライアント符号化方式のデフォルト変更

非互換

FUJITSU Enterprise Postgres 10では、Linux/Solarisにおいて、環境変数PGCLIENTENCODINGなどでクライアント符号化方式を指定していない場合のデフォルトを変更します。

FUJITSU Enterprise Postgres 9.6以前の場合

クライアント符号化方式はデータベースのエンコーディングとなります。

FUJITSU Enterprise Postgres 10以降の場合

クライアント符号化方式はクライアントロケールのエンコーディングとなります。

対処方法

クライアント符号化方式をデータベースのエンコーディングにする場合は、環境変数PGCLIENTENCODINGまたはclient_encodingパラメータで設定してください。

2.5.9 UTF-8の1文字あたりの最大バイト数の変更

非互換

FUJITSU Enterprise Postgres 10では、ODBCドライバにおいて、UTF-8の1文字あたりの最大バイト数を6から4に変更します。本変更により、SQLColAttribute関数にSQL_DESC_OCTET_LENGTHを指定して取得するデータのバイト数が変わります。

対処方法

ありません。

2.5.10 通信リンク障害発生時のSQLSTATEの変更

非互換

FUJITSU Enterprise Postgres 10では、ODBCアプリケーションの実行中に通信リンク障害が発生した場合、SQLSTATE "08S01"を返却するよう変更します。FUJITSU Enterprise Postgres 9.6以前は"08001"を返却していました。

対処方法

ありません。

2.6 .NET Data Providerに関する非互換

項目	移行前のバージョン				
	9.5	9.6	10	11	12
TargetServerTypeの値の変更	◎	◎	◎	◎	◎
空コマンドの実行時の例外発生に関する変更	○	○	○	○	○
tsquery型の値の表示の変更	○	○	○	○	○
Npgsql.NetTopologySuiteに必要なパッケージの変更	○	○	○	○	○
UseSslStreamプロパティの廃止	○	○	○	○	○
SnakeCaseNameTranslatorの動作の変更	○	○	○	○	○
.NET Frameworkアプリケーションの開発および動作に必要な環境の変更	○	○	○	○	○
組み込みのPostGISデータ型をNpgsql.LegacyPostGISプラグインに変更	○	○	○	○	－

項目	移行前のバージョン				
	9.5	9.6	10	11	12
NpgsqlParameterクラスのEnumTypeプロパティおよびSpecificTypeプロパティの削除	○	○	○	○	－
NpgsqlDataReaderクラスのGetDataTypeNameメソッドの返却値の変更	○	○	○	○	－
パラメータの型のチェックの強化	○	○	○	○	－
パラメータ名のチェックの強化	○	○	○	○	－
パラメータ名を省略した場合の動作の変更	○	○	○	○	－
関数を呼び出す場合の変更	○	○	○	○	－
NpgsqlInet構造体の削除	○	○	○	○	－
DateTimeマッピングの変更	○	○	○	○	－
DateTimeOffSetのDateの変更	○	○	○	○	－
小数点以下が28桁を超えるdecimal/numeric(浮動小数点でない)の変更	○	○	○	○	－
NpgsqlBinaryImporterの動作の変更	○	○	○	○	－
NpgsqlBinaryImporter.Cancelの削除	○	○	○	○	－
Enlistのデフォルト値の変更	○	○	○	○	－
マップされていないenum型を文字列として書き込む場合の設定方法の変更	○	○	○	○	－
TLS/SSLの内部実装の変更	○	○	○	○	－
独立性レベルの変更	○	○	○	○	－
IsAutoIncrementの返却値の変更	○	○	○	－	－
.NET Data ProviderのVisual Studio連携でのサーバーエクスプローラーの表示変更	◎	◎	－	－	－
エンティティフレームワークの名前の変更	○	○	－	－	－
SSLキーワードの削除	○	○	－	－	－
SslmodeキーワードのAllowモードの削除	○	○	－	－	－
PreloadReaderキーワードの削除	○	○	－	－	－
UseExtendedTypesキーワードの削除	○	○	－	－	－
AlwaysPrepareキーワードの削除	○	○	－	－	－
Protocolキーワードの削除	○	○	－	－	－
Compatibleキーワードの削除	○	○	－	－	－
SyncNotificationキーワードの削除	○	○	－	－	－
ConnectionLifeTimeキーワードの名前とデフォルト値の変更	○	○	－	－	－
MinPoolSizeキーワードとMaxPoolSizeキーワードのデフォルト値の変更	○	○	－	－	－
CommandTimeoutのデフォルト値の変更	○	○	－	－	－
NpgsqlTypes.NpgsqlMacAddress構造体の削除	○	○	－	－	－
NpgsqlTypes.BitString構造体の削除	○	○	－	－	－
NpgsqlTypes.NpgsqlTime構造体の削除	○	○	－	－	－
NpgsqlTypes.NpgsqlTimeZone構造体の削除	○	○	－	－	－

項目	移行前のバージョン				
	9.5	9.6	10	11	12
NpgsqlTypes.NpgsqlTimeTZ構造体の削除	○	○	－	－	－
NpgsqlDataReaderクラスのHasOrdinalメソッドの削除	○	○	－	－	－
NpgsqlParameterCollectionクラスのAdd(string, object)メソッドの削除	○	○	－	－	－
NpgsqlConnectionクラスのSupportsHexByteFormatプロパティの削除	○	○	－	－	－
NpgsqlExceptionクラスのBaseMessageプロパティの削除	○	○	－	－	－
NpgsqlConnectionクラスのSupports_E_StringPrefixプロパティの名前変更	○	○	－	－	－
NpgsqlInet構造体のフィールドの変更	○	○	－	－	－
通信データの暗号化を使用する場合にサーバ証明書を検証するように変更	○	○	－	－	－
NpgsqlParameterクラスのValueプロパティへのNULL値の設定方法の変更	○	○	－	－	－
パラメータへのIEnumerable<T>の値の設定不可	○	○	－	－	－
整数型のパラメータへの列挙体の設定不可	○	○	－	－	－
NpgsqlTypes.NpgsqlTimeStamp構造体の無限大の扱いの変更	○	○	－	－	－
NpgsqlDataReaderクラスのGetFieldTypeメソッドおよびGetProviderSpecificFieldTypeメソッドの返却値の変更	○	○	－	－	－
NpgsqlDataReaderクラスのGetDataTypeNameメソッドの返却値の変更	○	○	－	－	－
カーソルを返却するユーザ定義関数の使用不可	○	○	－	－	－
データベース接続時の送信情報の変更	○	○	－	－	－
NpgsqlConnectionクラスのCloneメソッドで複製されるNpgsqlConnectionオブジェクトの接続状態の変更	○	○	－	－	－
CommandType.TableDirectを指定して検索する場合の表の指定方法の変更	○	○	－	－	－
ユーザ定義関数を呼び出す場合の関数名の指定方法の変更	○	○	－	－	－
UTF-8として無効な文字を入力した場合にエラーにするように変更	○	○	－	－	－
STDINまたはSTDOUTを指定した場合のCOPY文の実行方法の変更	○	○	－	－	－
NpgsqlParameterクラスのSizeプロパティに指定されたサイズでパラメータの値を設定するように変更	○	○	－	－	－
トランザクションの隔離レベルにChaosを指定した場合の動作変更	○	○	－	－	－
NpgsqlConnectionクラスのStateプロパティとFullStateプロパティの設定値の追加	○	○	－	－	－
NpgsqlConnectionクラスのDataSourceプロパティの形式変更	○	○	－	－	－
コネクション接続後に接続文字列からパスワードの情報を除去するように変更	○	○	－	－	－

項目	移行前のバージョン				
	9.5	9.6	10	11	12
サーバでエラーが発生した場合にPostgresExceptionを返すように変更	○	○	－	－	－
SQLSTATEの取得方法の変更	○	○	－	－	－
NpgsqlDataReaderオブジェクトのオープン中に準備文を破棄できないように変更	○	○	－	－	－
環境変数PGCLIENTENCODINGを指定した場合にクライアント側符号化方式を設定するように変更	○	○	－	－	－
レコード型を返す関数の返却値のデータ型の変更	○	○	－		－
ユーザ定義関数を呼び出す場合の名前付きパラメータのサポート	○	○	－		－
NpgsqlConnectionクラスのコンストラクタの引数の変更	○	○	－		－

◎:FUJITSU Enterprise Postgresによる非互換あり

○:OSSによる非互換あり

－:非互換なし

2.6.1 TargetServerTypeの値の変更

非互換

FUJITSU Enterprise Postgres 13では、接続文字列に指定するTargetServerTypeの値を変更しました。そのため、以前利用していた値は利用できなくなります。

対処方法

TargetServerTypeの値を指定している場合は、以下のように値を変更してください。

表2.1 ターゲットサーバへの指定値

サーバの選択順	FUJITSU Enterprise Postgres 9.6以前	FUJITSU Enterprise Postgres 10/11/12	FUJITSU Enterprise Postgres 13
プライマリサーバ	primary	master	primary
スタンバイサーバ	-	slave	standby
スタンバイサーバの優先	prefer_standby	preferSlave	preferStandby
任意	-	any	any

2.6.2 空コマンドの実行時の例外発生に関する変更

非互換

FUJITSU Enterprise Postgres 13では、空コマンドの実行時に、今までは例外を発生させていませんでしたが、InvalidOperationExceptionを発生させるように修正しました。

そのため、空コマンドの実行時に例外処理を行っていない場合は、アプリの実行時にエラーが発生します。

対処方法

空コマンドを実行している場合は、適切な例外処理を行うように修正してください。

2.6.3 tsquery型の値の表示の変更

非互換

FUJITSU Enterprise Postgres 13では、tsquery型の列に記録した値を取得し表示する際、データベースに登録された値と逆順に表示される問題を修正しました。そのため、以前まで逆の順序で表示されていた値が、正しい順序で表示されるようになります。

対処方法

特にありません。

2.6.4 Npgsql.NetTopologySuiteに必要なパッケージの変更

非互換

FUJITSU Enterprise Postgres 13では、Npgsql.NetTopologySuiteのバージョンを上げました。このためにNetTopologySuite 2.0.0とEntity Framework Core 3.xが必要です。Net TopologySuite2.0.0を利用するためには、Entity Framework Core 2.xを引き続き利用することは出来ません。

対処方法

NetTopologySuite 2.0.0とEntity Framework Core 3.xをインストールしてください。

2.6.5 UseSslStreamプロパティの廃止

非互換

FUJITSU Enterprise Postgres 13では、UseSslStreamプロパティを廃止しました。

対処方法

FUJITSU Enterprise Postgres 12以前では、Npgsql独自のTLSではなく、.NET StreamSslを利用したい場合は、UseSslStreamの値をtrueに指定していました。FUJITSU Enterprise Postgres 13では、デフォルトで.NET StreamSslを利用するため、UseSslStreamの値を指定していた場合は、削除してください。

2.6.6 SnakeCaseNameTranslatorの動作の変更

非互換

FUJITSU Enterprise Postgres 13では、SnakeCaseNameTranslatorの動作を変更します。また、FUJITSU Enterprise Postgres 12でも、SnakeCaseNameTranslatorの動作を変更しています。

FUJITSU Enterprise Postgres 11以前の場合

例えば、IsJSONというプロパティ名を指定すると、出力はis_j_s_o_nになります。

FUJITSU Enterprise Postgres 12の場合

例えば、IsJSONというプロパティ名を指定すると、出力はis_jsonになります。

例えば、Xml2Jsonというプロパティ名を指定すると、出力はxml2_jsonになります。

FUJITSU Enterprise Postgres 13の場合

例えば、Xml2Jsonというプロパティ名を指定すると、出力はxml2jsonになります。

プロパティ名が変換された文字列が以前のものと異なる場合で、例えばその値を列名として利用していた場合には、移行前と移行後で列名が異なることになるため、移行後にSQL文が失敗する可能性があります。

対処方法

FUJITSU Enterprise Postgres 11以前と同じ動作を選択する場合には、legacyMode=trueを設定してください。

FUJITSU Enterprise Postgres 12と同じ動作は選択できません。

FUJITSU Enterprise Postgres 13への移行後には、実行されるSQL文が正しく動作するかを確認し、SQL文が失敗する場合には、移行後に正しく変換された文字列と、データベース内に存在する文字列とが一致するように修正してください。単純な修正方法としては、データ

ベース内に存在する誤って変換された文字列を正しく変換された場合の文字列に修正するものや、プロパティ名について、変換後の値が、データベース内に存在する文字列と一致するように、プロパティ名を修正するものなどがあります。

2.6.7 .NET Frameworkアプリケーションの開発および動作に必要な環境の変更

非互換

FUJITSU Enterprise Postgres 13では、.NET Frameworkアプリケーションの開発および動作に必要な環境を.NET Framework 4.6.1以降、または、.NET Standard 2.0以降に変更します。

対処方法

ありません。

2.6.8 組み込みのPostGISデータ型をNpgsql.LegacyPostGISプラグインに変更

非互換

FUJITSU Enterprise Postgres 12では、組み込みのPostGISデータ型を、Npgsql.LegacyPostGISプラグインに変更します。

対処方法

FUJITSU Enterprise Postgres 11までで、組み込みのPostGISデータ型を使用しているアプリケーションは、Npgsql.LegacyPostGISプラグインをセットアップして使用するように修正してください。

2.6.9 NpgsqlParameterクラスのEnumTypeプロパティおよびSpecificTypeプロパティの削除

非互換

FUJITSU Enterprise Postgres 12では、NpgsqlParameterクラスのEnumTypeプロパティおよびSpecificTypeプロパティを削除します。

対処方法

データベースとアプリケーションで列挙型のマッピングを作成する必要があります。

1. データベース・サーバで列挙型を作成します。

   ```
   > "CREATE TYPE pg_temp.mood1 AS ENUM ('sad', 'ok', 'happy')"
   ```

2. アプリケーションで、コネクション上に列挙型のマッピングを作成します。

   ```
   NpgsqlConnection.GlobalTypeMapper.MapEnum<Mood>("mood1");
   ```

マッピングが作成されると、通常どおりCLRの型の読み取りと書き込みが可能になります。

2.6.10 NpgsqlDataReaderクラスのGetDataTypeNameメソッドの返却値の変更

非互換

FUJITSU Enterprise Postgres 12では、NpgsqlDataReaderクラスのGetDataTypeNameメソッドが返却するデータ型名を変更します。

対処方法

既存のアプリケーションを確認し、データ型名をロジックに使用する場合は、変更後のデータ型名に修正する必要があります。

以下に変更の例を示しますが、これは変更されたすべてのデータ型名の完全なリストではありません。

- int2 → smallint
- int4 → integer
- int4 → integer[]

- int8 → bigint
- bool → Boolean
- numeric → numeric(11,7)
- varchar → varchar(10)

2.6.11 パラメータの型のチェックの強化

非互換

FUJITSU Enterprise Postgres 12では、設定されるすべてのパラメータ値は、パラメータに宣言されているものと同じ型である必要があります。

対処方法

既存のアプリケーションを確認し、設定されているすべてのパラメータ値が明示的にパラメータの型と同じ型になるように修正してください。

2.6.12 パラメータ名のチェックの強化

非互換

FUJITSU Enterprise Postgres 12では、設定されるすべてのパラメータ名は、SQLで指定されたパラメータ名と空白も含めて同じである必要があります。

対処方法

既存のアプリケーションを確認し、設定されるすべてのパラメータ名が、SQLで指定されたパラメータ名と空白も含めて同じであるように修正してください。

2.6.13 パラメータ名を省略した場合の動作の変更

非互換

FUJITSU Enterprise Postgres 12では、パラメータ名を省略した場合の動作を変更します。

FUJITSU Enterprise Postgres 11以前の場合

パラメータ名を省略した場合、デフォルトの名前はParameter1、Parameter2などになります。

FUJITSU Enterprise Postgres 12の場合

パラメータ名を指定しない場合、パラメータ名は設定されません。

対処方法

既存のアプリケーションを確認し、すべてのパラメータにパラメータ名が設定されるように修正してください。

2.6.14 関数を呼び出す場合の変更

非互換

FUJITSU Enterprise Postgres 12では、関数を呼び出すときは、NpgsqlCommand.CommandTypeをCommandType.StoredProcedureに明示的に設定する必要があります。

FUJITSU Enterprise Postgres 11以前の場合

NpgsqlCommand.CommandTypeをCommandType.StoredProcedureに設定しなくても関数を呼び出すことができます。

FUJITSU Enterprise Postgres 12の場合

NpgsqlCommand.CommandTypeをCommandType.StoredProcedureに明示的に設定する必要があります。

対処方法

既存のアプリケーションを確認し、NpgsqlCommandで関数を呼び出す場合に、NpgsqlCommand.CommandTypeをCommandType.StoredProcedureに明示的に設定する必要があります。

2.6.15 NpgsqlInet構造体の削除

非互換

FUJITSU Enterprise Postgres 12では、NpgsqlInet構造体を削除します。

対処方法

NpgsqlInet構造体を、ValueTuple<IpAdress, int>に置き換えてください。

2.6.16 DateTimeマッピングの変更

非互換

FUJITSU Enterprise Postgres 12では、DateTimeのマッピングを変更します。

FUJITSU Enterprise Postgres 11以前の場合

- localまたはunspecifiedとして指定されたNpgsqlDatetimeは、timestampにマップされます。
- UTCとして指定されたNpgsqlDatetimeは、timestamptzにマップされます。
- localまたはunspecifiedとして指定されたDatetimeは、timestampにマップされます。
- UTCで指定されたDatetimeは、timestamptzにマップされます。

FUJITSU Enterprise Postgres 12の場合

- DateTimeは、その種類にかかわらず、常にデフォルトでtimestampとして送信されます。NpgsqlDbType.TimestampTzを指定することもできます。この場合、ローカルDateTimeは送信前にUTCに変換されます。
- timestamptzをDateTimeOffsetとして読み取る場合、マシンのローカルオフセットが使用されます。
- DateTimeOffsetをtimestampとして読み書きすることはできなくなりました。timestamptzとしてのみ読み書き可能です。

対処方法

既存のアプリケーションを確認し、日時に関する値が期待通りに読み書きできるように修正する必要があります。

2.6.17 DateTimeOffSetのDateの変更

非互換

FUJITSU Enterprise Postgres 12では、PostgreSQLのtimetzに対して返されるDateTimeOffSetインスタンスのDateの値を変更します。

FUJITSU Enterprise Postgres 11以前の場合

PostgreSQL timetzに対して返されるDateTimeOffSetインスタンスのDateの値は0001-01-01に設定されます。

FUJITSU Enterprise Postgres 12の場合

PostgreSQL timetzに対して返されるDateTimeOffSetインスタンスのDateの値が0001-01-02に設定されます。

対処方法

既存のアプリケーションを確認し、DateTimeOffsetのDateの値を使用しないように修正してください。

2.6.18 小数点以下が28桁を超えるdecimal/numeric(浮動小数点でない)の変更

非互換

FUJITSU Enterprise Postgres 12では、28を超えるdecimal/numeric(浮動小数点でない)が正しく保管されるようになりました。

FUJITSU Enterprise Postgres 11以前の場合

小数点以下が28桁を超えるdecimal/numeric(浮動小数点でない)は正しく保管されません。

FUJITSU Enterprise Postgres 12の場合

小数点以下が28桁を超えるdecimal/numeric(浮動小数点でない)が正しく保管されます。

対処方法

FUJITSU Enterprise Postgres 11以前のNpgsqlによって挿入されたdecimal/numericのデータに問題がある場合、データを修正するためのスクリプトを実行してください。

https://github.com/npgsql/Npgsql.EntityFrameworkCore.PostgreSQL/issues/438#issuecomment-394014075

2.6.19 NpgsqlBinaryImporterの動作の変更

非互換

FUJITSU Enterprise Postgres 12では、NpgsqlBinaryImporterの動作を変更します。

FUJITSU Enterprise Postgres 11以前の場合

NpgsqlBinaryImporterは、closeまたはdispose時にインポートしたデータを自動的に保存します。

FUJITSU Enterprise Postgres 12の場合

NpgsqlBinaryImporterは、closeまたはdispose時にインポートしたデータを自動的に保存しないため、データを保存するためにはNpgsqlBinaryImporter.Commit()を呼び出す必要があります。

対処方法

NpgsqlBinaryImporterを使用する既存のアプリケーションは、インポートの最後にNpgsqlBinaryImporter.Commitを呼び出す必要があります。呼び出さない場合、データは保存されません。

2.6.20 NpgsqlBinaryImporter.Cancelの削除

非互換

FUJITSU Enterprise Postgres 12では、NpgsqlBinaryImporter.Cancelを削除します。

対処方法

ありません。

2.6.21 Enlistのデフォルト値の変更

非互換

FUJITSU Enterprise Postgres 12では、Enlistのデフォルト値はtrueとなります。

FUJITSU Enterprise Postgres 11以前の場合

Enlistのデフォルト値はfalseです。

FUJITSU Enterprise Postgres 12の場合

Enlistのデフォルト値はtrueです。

対処方法

既存のアプリケーションを確認し、アプリケーションがトランザクションスコープに参加しない場合は、Enlist=falseとなるように接続文字列を更新する必要があります。

2.6.22 マップされていないenum型を文字列として書き込む場合の設定方法の変更

非互換

FUJITSU Enterprise Postgres 12では、マップされていないenum型を文字列として書き込む場合の設定方法を変更します。

FUJITSU Enterprise Postgres 11以前の場合

マップされていないenumを文字列として書き込むには、NpgsqlDbType.Unknownを設定する必要があります。

FUJITSU Enterprise Postgres 12の場合

マップされていないenumを文字列として書き込むには、NpgsqlParameter.DataTypeNameをenum型の名前に設定する必要があります。

対処方法

enumを文字列として書き込む既存のアプリケーションは、NpgsqlSbType.Unknownフラグを設定する必要がなくなりました。

NpgsqlParameter.DataTypeNameを明示的にenum型の名前に設定するようにアプリケーションを修正する必要があります。

2.6.23 TLS/SSLの内部実装の変更

非互換

FUJITSU Enterprise Postgres 12では、TLS/SSLの内部実装を変更します。

FUJITSU Enterprise Postgres 11以前の場合

独自のTLS/SSLが使用されていました。

FUJITSU Enterprise Postgres 12の場合

デフォルトでは、.NET SslStreamが使用されるようになりました。

対処方法

アプリケーションが、以前のバージョンの内部SSLの実装を必要とする場合には、接続文字列にSSLStream=falseを指定してください。この実装は将来のリリースで削除される予定なので、この機能を必要とするユーザは、NET SSL Streamへの移行を計画する必要があることに注意してください。

2.6.24 独立性レベルの変更

非互換

FUJITSU Enterprise Postgres 12では、IslationLevel.SnapshotはRepeatableReadにマップされます。

FUJITSU Enterprise Postgres 11以前の場合

IsolationLevel.SnapshotはSerilizableにマップされます。

FUJITSU Enterprise Postgres 12の場合

IslationLevel.SnapshotはRepeatableReadにマップされます。

対処方法

FUJITSU Enterprise Postgres 11以前と同じ動作を選択する場合には、IsolationLevel.Snapshotの代わりに IsolationLevel.Serializableを明示的に使用する必要があります。

2.6.25 IsAutoIncrementの返却値の変更

非互換

FUJITSU Enterprise Postgres 11では、IDENTITYを指定して作成した列に対して、NpgsqlDataReaderクラスのIsAutoIncrementメソッドでメタデータの取得を行った場合、trueを返却するするように変更します。

FUJITSU Enterprise Postgres 10以前の場合

falseを返却します。

FUJITSU Enterprise Postgres 11の場合

trueを返却します。

対処方法

ありません。

2.6.26 .NET Data ProviderのVisual Studio連携でのサーバーエクスプローラーの表示変更

非互換

FUJITSU Enterprise Postgres 10以降では、サーバーエクスプローラーでデータベース資源を表示した場合に以下の変更があります。

- Tablesフォルダ配下に"スキーマ名.テーブル名"で表示されます。
- Indexesフォルダは表示されません。
- Triggersフォルダは表示されません。
- Sequencesフォルダ表示されません。
- Proceduresフォルダは表示されません。

対処方法

ありません。

2.6.27 エンティティフレームワークの名前の変更

非互換

FUJITSU Enterprise Postgres 10では、以下のとおり、エンティティフレームワークの名前を変更します。

FUJITSU Enterprise Postgres 9.6以前	FUJITSU Enterprise Postgres 10
Npgsql.EntityFramework	Npgsql.EntityFramework6
Npgsql.EntityFrameworkLegacy	Npgsql.EntityFramework5

対処方法

ありません。

2.6.28 SSLキーワードの削除

非互換

FUJITSU Enterprise Postgres 10では、SSLキーワードを削除します。

対処方法

クライアントとサーバ間の通信路をSSLプロトコルで暗号化する場合、Sslmodeキーワードを設定してください。

2.6.29 SslmodeキーワードのAllowモードの削除

非互換

FUJITSU Enterprise Postgres 10では、SslmodeキーワードのAllowモードを削除します。

対処方法

ありません。

2.6.30 PreloadReaderキーワードの削除

非互換

FUJITSU Enterprise Postgres 10では、PreloadReaderキーワードを削除します。

対処方法

ありません。

2.6.31 UseExtendedTypesキーワードの削除

非互換

FUJITSU Enterprise Postgres 10では、UseExtendedTypesキーワードを削除します。

対処方法

標準のCLRのデータ型で表現できないデータにアクセスする場合は、NpgsqlDataReaderクラスのGetProviderSpecificValueメソッド、またはGetFieldValueメソッドを使用して、目的とするオブジェクトの型でデータにアクセスしてください。

2.6.32 AlwaysPrepareキーワードの削除

非互換

FUJITSU Enterprise Postgres 10では、AlwaysPrepareキーワードを削除します。

対処方法

ありません。

2.6.33 Protocolキーワードの削除

非互換

FUJITSU Enterprise Postgres 10では、Protocolキーワードを削除し、常にプロトコルバージョン3として動作するように変更します。

対処方法

ありません。

2.6.34 Compatibleキーワードの削除

非互換

FUJITSU Enterprise Postgres 10では、Compatibleキーワードを削除します。

対処方法

ありません。

2.6.35 SyncNotificationキーワードの削除

非互換

FUJITSU Enterprise Postgres 10では、SyncNotificationキーワードを削除します。

対処方法

通知イベントを同期して待つ場合は、NpgsqlConnectionクラスのWaitメソッドで待ち合わせるように変更してください。通知イベントを非同期で待つ場合は、NpgsqlConnectionクラスのWaitAsyncメソッドで待ち合わせるように変更してください。

2.6.36 ConnectionLifeTimeキーワードの名前とデフォルト値の変更

非互換

FUJITSU Enterprise Postgres 10では、ConnectionLifeTimeキーワードの名前をConnectionIdleLifetimeに変更します。また、デフォルト値を15秒から300秒に変更します。

対処方法

必要に応じてConnectionIdleLifetimeキーワードの設定を変更してください。

2.6.37 MinPoolSizeキーワードとMaxPoolSizeキーワードのデフォルト値の変更

非互換

FUJITSU Enterprise Postgres 9.6では、MinPoolSizeキーワードとMaxPoolSizeキーワードのデフォルト値を以下のとおりに変更します。

パラメータ	FUJITSU Enterprise Postgres 9.6以前	FUJITSU Enterprise Postgres 10
MinPoolSize	1	0
MaxPoolSize	20	100

対処方法

必要に応じてMinPoolSizeキーワードとMaxPoolSizeキーワードの設定を変更してください。

2.6.38 CommandTimeoutのデフォルト値の変更

非互換

FUJITSU Enterprise Postgres 10では、CommandTimeoutのデフォルト値を20秒から30秒に変更します。

対処方法

必要に応じてCommandTimeoutの設定を変更してください。



参考

CommandTimeoutは以下の方法で設定することが可能です。

- 接続文字列のCommandTimeoutキーワード
- NpgsqlConnectionStringBuilderのCommandTimeoutプロパティ
- NpgsqlConnectionのCommandTimeoutプロパティ
- NpgsqlCommandのCommandTimeoutプロパティ

2.6.39 NpgsqlTypes.NpgsqlMacAddress構造体の削除

非互換

FUJITSU Enterprise Postgres 10では、NpgsqlMacAddress構造体を削除します。

対処方法

NpgsqlMacAddress構造体をSystem.Net.NetworkInformation.PhysicalAddressクラスに置き換えてください。

2.6.40 NpgsqlTypes.BitString構造体の削除

非互換

FUJITSU Enterprise Postgres 10では、BitString構造体を削除します。

対処方法

BitString構造体をSystem.Collections.BitArrayクラス、またはSystem.Boolean構造体の配列に置き換えてください。

2.6.41 NpgsqlTypes.NpgsqlTime構造体の削除

非互換

FUJITSU Enterprise Postgres 10では、NpgsqlTime構造体を削除します。

対処方法

NpgsqlTime構造体をSystem.DateTime構造体に置き換えてください。

2.6.42 NpgsqlTypes.NpgsqlTimeZone構造体の削除

非互換

FUJITSU Enterprise Postgres 10では、NpgsqlTimeZone構造体を削除します。

対処方法

NpgsqlTimeZone構造体をSystem.TimeSpan構造体に置き換えてください。

2.6.43 NpgsqlTypes.NpgsqlTimeTZ構造体の削除

非互換

FUJITSU Enterprise Postgres 10では、NpgsqlTimeTZ構造体を削除します。

対処方法

NpgsqlTimeTZ構造体をSystem.DateTimeOffset構造体に置き換えてください。

2.6.44 NpgsqlDataReaderクラスのHasOrdinalメソッドの削除

非互換

FUJITSU Enterprise Postgres 10では、NpgsqlDataReaderクラスのHasOrdinalメソッドを削除します。

対処方法

NpgsqlDataReaderクラスのFieldCountプロパティとGetNameメソッドを使用して列名の存在確認を行うことが可能です。

2.6.45 NpgsqlParameterCollectionクラスのAdd(string, object)メソッドの削除

非互換

FUJITSU Enterprise Postgres 10では、NpgsqlParameterCollectionクラスのAdd(string, object)メソッドを削除します。

対処方法

NpgsqlParameterCollectionクラスのAddWithValueメソッドに置き換えてください。

2.6.46 NpgsqlConnectionクラスのSupportsHexByteFormatプロパティの削除

非互換

FUJITSU Enterprise Postgres 10では、NpgsqlConnectionクラスのSupportsHexByteFormatプロパティを削除します。

対処方法

ありません。

2.6.47 NpgsqlExceptionクラスのBaseMessageプロパティの削除

非互換

FUJITSU Enterprise Postgres 10では、NpgsqlExceptionクラスのBaseMessageプロパティを削除します。

対処方法

ありません。

2.6.48 NpgsqlConnectionクラスのSupports_E_StringPrefixプロパティの名前変更

非互換

FUJITSU Enterprise Postgres 10では、NpgsqlConnectionクラスのSupports_E_StringPrefixプロパティの名前をSupportsEStringPrefixに変更します。

対処方法

ありません。

2.6.49 NpgsqlInet構造体のフィールドの変更

非互換

FUJITSU Enterprise Postgres 10では、NpgsqlInet構造体のaddrフィールド、およびmaskフィールドをそれぞれAddressプロパティ、およびNetmaskプロパティに変更します。

対処方法

NpgsqlInet構造体のaddrフィールド、およびmaskフィールドをそれぞれAddressプロパティ、およびNetmaskプロパティへ変更してください。

2.6.50 通信データの暗号化を使用する場合にサーバ証明書を検証するように変更

非互換

FUJITSU Enterprise Postgres 10では、通信データの暗号化を使用する場合、デフォルトでサーバ証明書を検証するように変更します。

FUJITSU Enterprise Postgres 9.6以前の場合

デフォルトではサーバ証明書の検証を実施しません。このため、サーバ証明書として自己署名証明書を使用できます。

FUJITSU Enterprise Postgres 10以降の場合

デフォルトでサーバ証明書の検証を実施します。このため、サーバ証明書として自己署名証明書は使用できません。

対処方法

サーバ証明書を検証する必要がない場合、TrustServerCertificateキーワードを使用して、サーバ証明書検証を無効に設定してください。

2.6.51 NpgsqlParameterクラスのValueプロパティへのNULL値の設定方法の変更

非互換

FUJITSU Enterprise Postgres 10では、NpgsqlParameterクラスのValueプロパティへNULL値を設定する方法が変更となります。

FUJITSU Enterprise Postgres 9.6以前の場合

NpgsqlParameterクラスのValueプロパティにnullを設定することでNULL値を設定することが可能です。ただし、本方法によるNULL値の設定は、ADO.NETに準拠していません。

FUJITSU Enterprise Postgres 10以降の場合

NpgsqlParameterクラスのValueプロパティにnullを設定した場合、例外を返します。

対処方法

NULL値を扱う場合、NpgsqlParameterクラスのValueプロパティにDBNull.Valueを設定してください。

2.6.52 パラメータへのIEnumerable<T>の値の設定不可

非互換

FUJITSU Enterprise Postgres 10では、パラメータへIEnumerable<T>の値を設定できなくするように変更します。

対処方法

ToListメソッド、ToArrayメソッドなどを使用して、配列値としてパラメータを設定してください。

2.6.53 整数型のパラメータへの列挙体の設定不可

非互換

FUJITSU Enterprise Postgres 10では、整数型のパラメータへ列挙体を設定できなくするように変更します。

FUJITSU Enterprise Postgres 9.6以前の場合

整数型のパラメータへ列挙体を設定することができます。

FUJITSU Enterprise Postgres 10以降の場合

整数型のパラメータへ列挙体を設定することはできません。

対処方法

整数型への明示的なキャストを付加してください。

2.6.54 NpgsqlTypes.NpgsqlTimeStamp構造体の無限大の扱いの変更

非互換

FUJITSU Enterprise Postgres 10では、NpgsqlTimeStamp構造体の無限大として扱う値を変更します。

FUJITSU Enterprise Postgres 9.6以前の場合

正の無限大をDateTime.MAX、負の無限大をDateTime.MINとして扱います。

FUJITSU Enterprise Postgres 10以降の場合

正の無限大をNpgsqlTimeStamp.Infinity、負の無限大をNpgsqlTimeStamp.NegativeInfinityとして扱います。また、下記のようなinfinity、-infinityをDate型へ変換できなくなります。

[例]

SELECT 'infinity'::Date, '-infinity'::Date

対処方法

ConvertInfinityDateTimeキーワードにtrueを設定することで、infinity、-infinityをDateTime.MAX、DateTime.MINとして扱うことが可能です。

2.6.55 NpgsqlDataReaderクラスのGetFieldTypeメソッドおよびGetProviderSpecificFieldTypeメソッドの返却値の変更

非互換

FUJITSU Enterprise Postgres 10では、NpgsqlDataReaderクラスのGetFieldTypeメソッド、およびGetProviderSpecificFieldTypeメソッドにおいて、フィールドのデータ型が配列型の場合、Array型として返却するように変更します。

FUJITSU Enterprise Postgres 9.6以前の場合

任意の配列型(例えば、int[])として返却します。

FUJITSU Enterprise Postgres 10以降の場合

Array型として返却します。

対処方法

ありません。

2.6.56 NpgsqlDataReaderクラスのGetDataTypeNameメソッドの返却値の変更

非互換

FUJITSU Enterprise Postgres 10では、NpgsqlDataReaderクラスのGetDataTypeNameメソッドがPostgreSQLのデータ型名を返却するように変更します。

FUJITSU Enterprise Postgres 9.6以前の場合

OIDを返却します。

FUJITSU Enterprise Postgres 10以降の場合

PostgreSQLのデータ型名を返却します。

対処方法

ありません。

2.6.57 カーソルを返却するユーザ定義関数の使用不可

非互換

FUJITSU Enterprise Postgres 10では、カーソルを返却するユーザ定義関数を使用できないように変更します。

対処方法

ありません。

2.6.58 データベース接続時の送信情報の変更

非互換

FUJITSU Enterprise Postgres 10では、データベースに接続する際、ロケールと書式設定に関する下記の情報を送信しないように変更します。

- DateStyle = ISO

- lc_monetary = C
- extra_float_digits = 3
-

対処方法

必要に応じてSET文などを使用してロケールまたは書式設定に関する情報を設定してください。

2.6.59 NpgsqlConnectionクラスのCloneメソッドで複製されるNpgsqlConnectionオブジェクトの接続状態の変更

非互換

FUJITSU Enterprise Postgres 10では、NpgsqlConnectionクラスのCloneメソッドで複製されるNpgsqlConnectionオブジェクトの接続状態を変更します。

FUJITSU Enterprise Postgres 9.6以前の場合

複製元のNpgsqlConnectionオブジェクトがデータベースと接続確立状態である場合、複製元と同じ接続文字列を使用してデータベースと接続を確立したNpgsqlConnectionオブジェクトを複製します。複製元に設定されている通知イベントハンドラ(NoticeEventHandler)は、複製したNpgsqlConnectionオブジェクトにも設定します。

FUJITSU Enterprise Postgres 10以降の場合

複製元のNpgsqlConnectionオブジェクトのデータベースとの接続状態によらず、複製元と同じ接続文字列を保有するデータベースと未接続状態のNpgsqlConnectionオブジェクトを複製します。複製元に設定されている通知イベントハンドラは、複製したNpgsqlConnectionオブジェクトには設定しません。

対処方法

NpgsqlConnectionクラスのCloneメソッドで複製されたNpgsqlConnectionオブジェクトを使用して、データベースに接続してください。必要に応じて複製されたNpgsqlConnectionオブジェクトに通知イベントハンドラを設定してください。

2.6.60 CommandType.TableDirectを指定して検索する場合の表の指定方法の変更

非互換

FUJITSU Enterprise Postgres 10では、NpgsqlCommandクラスのCommandTypeプロパティにCommandType. TableDirectを指定して表を検索する場合の表の指定方法を変更します。

FUJITSU Enterprise Postgres 9.6以前の場合

NpgsqlCommandクラスのCommandTextには複数の表を指定して、それぞれの表を検索することができます。

FUJITSU Enterprise Postgres 10以降の場合

NpgsqlCommandクラスのCommandTextには表は1つしか指定できません。複数指定した場合はエラーになります。

対処方法

ありません。

2.6.61 ユーザ定義関数を呼び出す場合の関数名の指定方法の変更

非互換

FUJITSU Enterprise Postgres 10では、NpgsqlCommandクラスのCommandTypeプロパティにCommandType.StoredProcedureを指定してユーザ定義関数を呼び出す場合の関数名の指定方法を変更します。

FUJITSU Enterprise Postgres 9.6以前の場合

NpgsqlCommandクラスのCommandTextには関数名のほか、括弧やパラメータ情報も指定できます。

FUJITSU Enterprise Postgres 10以降の場合

NpgsqlCommandクラスのCommandTextには関数名のみ指定します。括弧やパラメータ情報を指定した場合はエラーになります。

対処方法

ありません。

2.6.62 UTF-8として無効な文字を入力した場合にエラーにするように変更

非互換

FUJITSU Enterprise Postgres 10では、UTF-8として無効な文字を入力した場合に、そのまま処理を継続するのではなく、例外を返すように変更します。

対処方法

ありません。

2.6.63 STDINまたはSTDOUTを指定した場合のCOPY文の実行方法の変更

非互換

FUJITSU Enterprise Postgres 10では、STDIN、またはSTDOUTを指定した場合のCOPY文の実行方法を変更し、新しいAPIを追加します。これに伴い、FUJITSU Enterprise Postgres 9.6以前に使用していたAPIは削除します。

対処方法

.NETアプリケーションでのCOPY文の実行方法を“Npgsql - .NET Access to PostgreSQL”のドキュメントで確認し、アプリケーションを修正してください。

2.6.64 NpgsqlParameterクラスのSizeプロパティに指定されたサイズでパラメータの値を設定するように変更

非互換

FUJITSU Enterprise Postgres 10では、以下のとおり、NpgsqlParameterクラスのSizeプロパティに指定されたサイズでパラメータの値を設定するように変更します。

FUJITSU Enterprise Postgres 9.6以前の場合

NpgsqlParameterクラスのSizeプロパティの値にかかわらず、指定された値すべてをパラメータに設定します。

FUJITSU Enterprise Postgres 10以降の場合

NpgsqlParameterクラスのSizeプロパティに指定したサイズでパラメータを設定します。-1を指定した場合は、指定された値すべてをパラメータに設定します。-2以下を指定した場合は、例外を返します。

対処方法

ありません。

2.6.65 トランザクションの隔離レベルにChaosを指定した場合の動作変更

非互換

FUJITSU Enterprise Postgres 10では、トランザクションの隔離レベルにChaosを指定した場合に例外を返すように変更します。

FUJITSU Enterprise Postgres 9.6以前の場合

トランザクションの隔離レベルにIsolationLevel.Chaosを指定した場合、隔離レベルをREAD COMMITTEDにします。

FUJITSU Enterprise Postgres 10以降の場合

トランザクションの隔離レベルにIsolationLevel.Chaosを指定した場合、例外を返します。

対処方法

ありません。

2.6.66 NpgsqlConnectionクラスのStateプロパティとFullStateプロパティの設定値の追加

非互換

FUJITSU Enterprise Postgres 10では、以下のとおり、NpgsqlConnectionクラスのStateプロパティとFullStateプロパティに新しい設定値を追加します。

プロパティ	FUJITSU Enterprise Postgres 9.6以前	FUJITSU Enterprise Postgres 10
State	ConnectionState.Closed ConnectionState.Open	ConnectionState.Closed ConnectionState.Connecting ConnectionState.Open
FullState	ConnectionState.Closed ConnectionState.Open	ConnectionState.Broken ConnectionState.Closed ConnectionState.Connecting ConnectionState.Executing ConnectionState.Fetching ConnectionState.Open

対処方法

ありません。

2.6.67 NpgsqlConnectionクラスのDataSourceプロパティの形式変更

非互換

FUJITSU Enterprise Postgres 10では、NpgsqlConnectionクラスのDataSourceプロパティの形式を以下のとおりに変更します。

FUJITSU Enterprise Postgres 9.6以前の場合

"ホスト名またはIPアドレス"

FUJITSU Enterprise Postgres 10以降の場合

"tcp://ホスト名またはIPアドレス:ポート番号"

対処方法

ありません。

2.6.68 コネクション接続後に接続文字列からパスワードの情報を除去するように変更

非互換

FUJITSU Enterprise Postgres 10では、コネクション接続後に接続文字列からパスワードの情報を除去するように変更します。

対処方法

PersistSecurityInfoキーワードにtrueを指定してください。

2.6.69 サーバでエラーが発生した場合にPostgresExceptionを返すように変更

非互換

FUJITSU Enterprise Postgres 10では、サーバでエラーが発生した場合、NpgsqlExceptionではなく、NpgsqlExceptionのサブクラスであるPostgresExceptionを返すように変更します。

対処方法

サーバのエラー情報を参照する場合は、PostgresExceptionをキャッチするか、NpgsqlExceptionをPostgresExceptionにキャストしてからプロパティを参照してください。

2.6.70 SQLSTATEの取得方法の変更

非互換

FUJITSU Enterprise Postgres 10では、SQLSTATEの取得方法を以下のとおりに変更します。

FUJITSU Enterprise Postgres 9.6以前の場合

NpgsqlExceptionクラスのCodeプロパティから取得します。

FUJITSU Enterprise Postgres 10以降の場合

PostgresExceptionのSqlStateプロパティから取得します。NpgsqlExceptionクラスのCodeプロパティからは取得できません。

対処方法

ありません。

2.6.71 NpgsqlDataReaderオブジェクトのオープン中に準備文を破棄できないように変更

非互換

FUJITSU Enterprise Postgres 10では、NpgsqlDataReaderオブジェクトのオープン中に準備文を破棄できないように変更します。

対処方法

NpgsqlDataReaderオブジェクトのクローズ後に破棄してください。

2.6.72 環境変数PGCLIENTENCODINGを指定した場合にクライアント側符号化方式を設定するように変更

非互換

FUJITSU Enterprise Postgres 10では、環境変数PGCLIENTENCODINGを指定した場合、クライアント側符号化方式を設定するように変更します。

FUJITSU Enterprise Postgres 9.6以前の場合

環境変数PGCLIENTENCODINGの値にかかわらず、クライアント側符号化方式はUTF-8となります。

FUJITSU Enterprise Postgres 10以降の場合

クライアント側符号化方式は環境変数PGCLIENTENCODINGに指定した値となります。環境変数PGCLIENTENCODINGを指定していない場合はUTF-8となります。

対処方法

ありません。

2.6.73 レコード型を返す関数の返却値のデータ型の変更

非互換

FUJITSU Enterprise Postgres 10では、レコード型を返す関数の返却値のデータ型をstringクラスからObjectクラスの配列に変更します。

対処方法

ありません。

2.6.74 ユーザ定義関数を呼び出す場合の名前付きパラメータのサポート

非互換

FUJITSU Enterprise Postgres 10では、NpgsqlCommandクラスのCommandTypeプロパティにCommandType.StoredProcedureを指定してユーザ定義関数を呼び出す場合の名前付きパラメータをサポートします。

FUJITSU Enterprise Postgres 9.6以前の場合

パラメータの名前は照合されません。AddWithValueメソッドを呼び出した順に値が設定されます。

FUJITSU Enterprise Postgres 10以降の場合

パラメータの名前を照合して値を設定します。このため、パラメータ名が一致しない場合はエラーになります。

対処方法

AddWithValueメソッドに指定しているパラメータ名がユーザ定義関数の引数の名前と異なる場合は、一致するように合わせてください。

2.6.75 NpgsqlConnectionクラスのコンストラクタの引数の変更

非互換

FUJITSU Enterprise Postgres 10では、NpgsqlConnectionクラスのコンストラクタの引数を変更し、stringクラスのみ指定可能にするように変更します。

FUJITSU Enterprise Postgres 9.6以前の場合

NpgsqlConnectionクラスのコンストラクタにはstringクラスとNpgsqlConnectionStringBuilderクラスを指定することができます。

FUJITSU Enterprise Postgres 10以降の場合

NpgsqlConnectionクラスのコンストラクタにはstringクラスのみが指定可能です。

対処方法

NpgsqlConnectionクラスのコンストラクタの引数にNpgsqlConnectionStringBuilderオブジェクトを指定している場合は、ConnectionStringプロパティを使用するなど、文字列を渡すように変更してください。

2.7 C言語用ライブラリ(libpq)に関する非互換

項目	移行前のバージョン				
	9.5	9.6	10	11	12
接続状態関数PQhostの動作の変更	○	○	○	－	－
ホスト名を省略した場合における接続状態関数PQhostの返却値の変更	○	－	－	－	－

◎:FUJITSU Enterprise Postgresによる非互換あり

○:OSSによる非互換あり

ー:非互換なし

2.7.1 接続状態関数PQhostの動作の変更

非互換

FUJITSU Enterprise Postgres 11では、接続状態関数PQhostにおいて、以下のパラメータを指定しない場合の動作を変更します。

- hostパラメータとhostaddrパラメータの両方を指定する
- hostパラメータを指定しない、かつhostaddrパラメータを指定する
- hostパラメータとhostaddrパラメータの両方を指定しない

FUJITSU Enterprise Postgres 9.6以前の場合

ー hostパラメータとhostaddrパラメータの両方を指定する場合
ホスト名を返します。
接続にパスワードが必要な場合、hostパラメータをキーにパスワードファイルのhostnameフィールドを検索します。

ー hostパラメータを指定しない、かつhostaddrパラメータを指定する場合
デフォルトホスト名(/tmpまたはlocalhost)を返します。
接続にパスワードが必要な場合、“localhost”をキーにパスワードファイルのhostnameフィールドを検索します。

ー hostパラメータとhostaddrパラメータの両方を指定しない場合
デフォルトホスト名(/tmpまたはlocalhost)を返します。
接続にパスワードが必要な場合、“localhost”をキーにパスワードファイルのhostnameフィールドを検索します。

FUJITSU Enterprise Postgres 10の場合

ー hostパラメータとhostaddrパラメータの両方を指定する場合
ホスト名を返します。
接続にパスワードが必要な場合、hostaddrパラメータをキーにパスワードファイルのhostnameフィールドを検索します。

ー hostパラメータを指定しない、かつhostaddrパラメータを指定する場合
デフォルトホスト名(/tmpまたはlocalhost)を返します。
接続にパスワードが必要な場合、“localhost”をキーにパスワードファイルのhostnameフィールドを検索します。

ー hostパラメータとhostaddrパラメータの両方を指定しない場合
デフォルトホスト名(/tmpまたはlocalhost)を返します。
接続にパスワードが必要な場合、“localhost”をキーにパスワードファイルのhostnameフィールドを検索します。

FUJITSU Enterprise Postgres 11の場合

ー hostパラメータとhostaddrパラメータの両方を指定する場合
ホスト名を返します。
接続にパスワードが必要な場合、hostパラメータをキーにパスワードファイルのhostnameフィールドを検索します。

ー hostパラメータを指定しない、かつhostaddrパラメータを指定する場合
IPアドレスを返します。
接続にパスワードが必要な場合、hostaddrパラメータをキーにパスワードファイルのhostnameフィールドを検索します。

ー hostパラメータとhostaddrパラメータの両方を指定しない場合
デフォルトホスト名(/tmpまたはlocalhost)を返します。
接続にパスワードが必要な場合、“localhost”をキーにパスワードファイルのhostnameフィールドを検索します。

対処方法

ありません。

2.7.2 ホスト名を省略した場合における接続状態関数PQhostの返却値の変更

非互換

FUJITSU Enterprise Postgres 9.6以降では、接続状態関数PQhostにおいて、ホスト名を省略した場合はNULLではなく、デフォルトのソケットディレクトリのパスを返すように変更します。

対処方法

ありません。

2.8 oracle_fdwに関する非互換

項目	移行前のバージョン				
	9.5	9.6	10	11	12
外部テーブルのパーティションテーブルに対してINSERT ... ON CONFLICTを実行した時のエラーメッセージを変更	○	○	○	○	○
外部テーブルのパーティションテーブルに対してINSERT ... RETURNINGを実行した時に挿入された通りの行を返却するように変更	○	○	○	○	○
外部テーブル使用時にOracleからPostgreSQLへのデータ型の変換チェック処理を追加	―	―	○	―	―
外部テーブルの外部結合をOracleにプッシュダウンするように変更	―	―	○	―	―

◎:FUJITSU Enterprise Postgresによる非互換あり

○:OSSによる非互換あり

―:非互換なし

2.8.1 外部テーブルのパーティションテーブルに対してINSERT ... ON CONFLICTを実行した時のエラーメッセージを変更

非互換

FUJITSU Enterprise Postres 13では、外部テーブルのパーティションテーブルに対してINSERT ... ON CONFLICTを実行した時のエラーメッセージを変更します。

FUJITSU Enterprise Postres 12以前の場合

外部テーブルのパーティションテーブルに対してINSERT ... ON CONFLICTを実行した時、下記のメッセージが出力されます。

```
ERROR:  error executing query: OCIStmtExecute failed to execute remote query
DETAIL:  ORA-00001: unique constraint (SCOTT.SYS_C007621) violated
```

[例]

```
=# ¥d
                リレーション一覧
 スキーマ |   名前    |         タイプ          | 所有者
----------+-----------+-------------------------+--------
 public   | ft1       | 外部テーブル            | user
 public   | ft1_part  | パーティションテーブル | user

=# INSERT INTO ft1_part values(1,'row1') ON CONFLICT DO NOTHING;
ERROR:  error executing query: OCIStmtExecute failed to execute remote query
DETAIL:  ORA-00001: unique constraint (SCOTT.SYS_C007621) violated
```

FUJITSU Enterprise Postres 13の場合

外部テーブルのパーティションテーブルに対してINSERT ... ON CONFLICTを実行した時、下記のメッセージが出力されます。

```
ERROR:  INSERT with ON CONFLICT clause is not supported
```

[例]

```
=# ¥d
                リレーション一覧
 スキーマ |   名前    |        タイプ         | 所有者
----------+-----------+-----------------------+--------
 public   | ft1       | 外部テーブル          | user
 public   | ft1_part  | パーティションテーブル | user

=# INSERT INTO ft1_part values(2,'row2') ON CONFLICT DO NOTHING;
ERROR:  INSERT with ON CONFLICT clause is not supported
```

対処方法

ありません。

2.8.2 外部テーブルのパーティションテーブルに対してINSERT ... RETURNINGを実行した時に挿入された通りの行を返却するように変更

非互換

FUJITSU Enterprise Postres 13では、外部テーブルのパーティションテーブルに対してINSERT ... RETURNINGを実行した時に挿入された通りの行を返却するように変更します。

FUJITSU Enterprise Postres 12以前の場合

外部テーブルのパーティションテーブルに対してINSERT ... RETURNINGを実行した時、NULLが返却されます。

[例]

```
=# ¥d
                リレーション一覧
 スキーマ |   名前    |        タイプ         | 所有者
----------+-----------+-----------------------+--------
 public   | ft1       | 外部テーブル          | user
 public   | ft1_part  | パーティションテーブル | user

=# INSERT INTO ft1_part VALUES(1,'row1') RETURNING *;
 id | val
----+-----
    |
(1 行)
```

FUJITSU Enterprise Postres 13の場合

外部テーブルのパーティションテーブルに対してINSERT ... RETURNINGを実行した時、挿入された通りの行が返却されます。

[例]

```
=# ¥d
                リレーション一覧
 スキーマ |   名前    |        タイプ         | 所有者
----------+-----------+-----------------------+--------
 public   | ft1       | 外部テーブル          | user
 public   | ft1_part  | パーティションテーブル | user

=# INSERT INTO ft1_part VALUES(2,'row2') RETURNING *;
 id | val
----+-----
  2 | row2
(1 行)
```

対処方法

ありません。

2.8.3 外部テーブル使用時にOracleからPostgreSQLへのデータ型の変換チェック処理を追加

非互換

FUJITSU Enterprise Postgres 11では、外部テーブル使用時のOracleからPostgreSQL側へのデータ型変換において、データ型変換チェック処理を追加するよう変更します。

FUJITSU Enterprise Postgres 10以前の場合

外部テーブルを使用しPostgreSQLに存在しないOracleのデータ型を参照した場合、正常終了します。

[例]

```
=# ¥det
        外部テーブル一覧
 スキーマ | テーブル  | サーバー
----------+-----------+----------
 ・・・
 public   | ft1       | oracle
(2 行)
=# SELECT * FROM ft1;
     c1     |  c2  |     bfile
------------+------+----------------
 001        | row1 | this is bfile.
            |      |
(1 行)
```

FUJITSU Enterprise Postgres 11の場合

外部テーブルを使用しPostgreSQLに存在しないOracleのデータ型を参照した場合、エラーとなります。

[例]

```
=# ¥det
        外部テーブル一覧
 スキーマ | テーブル  | サーバー
----------+-----------+----------
 ・・・
 public   | ft1       | oracle
(2 行)
=# SELECT * FROM ft1;
ERROR:  column "bfile" of foreign table "ft1" cannot be converted to or from Oracle data type
```

対処方法

ありません。

2.8.4 外部テーブルの外部結合をOracleにプッシュダウンするように変更

非互換

FUJITSU Enterprise Postgres 11では、外部テーブルの外部結合時において、PostgreSQL側からOracle側へプッシュダウンするよう変更します。

FUJITSU Enterprise Postgres 10以前の場合

Oracle query側では各外部テーブルの抽出のみ実施し、外部結合はPostgreSQL側で実施します。

[例]

```
=# EXPLAIN SELECT a.c1, a.c2, b.c3 FROM ft1 a LEFT OUTER JOIN ft2 b ON a.c1 = b.c1 AND substr(a.c2, -1,1) = 'd';

                QUERY PLAN

-----------------------------------------------------------------------------------------------------

 Hash Left Join  (cost=30012.50..40215.00 rows=1000 width=66)

   Hash Cond: (a.c1 = b.c1)

   Join Filter: (substr((a.c2)::text, '-1'::integer, 1) = 'd'::text)

   ->  Foreign Scan on ft1 a  (cost=10000.00..20000.00 rows=1000 width=62)

         Oracle query: SELECT /*7543493300debf649a78c5dbe11c9be4*/ r1."C1", r1."C2" FROM "T1" r1

   ->  Hash  (cost=20000.00..20000.00 rows=1000 width=8)

         ->  Foreign Scan on ft2 b  (cost=10000.00..20000.00 rows=1000 width=8)

               Oracle query: SELECT /*afb5375733f8fc53a93d0df585c01b9f*/ r2."C1", r2."C3" FROM "T2" r2

(8 行)

=# SELECT a.c1, a.c2, b.c3 FROM ft1 a LEFT OUTER join ft2 b ON a.c1 = b.c1 AND substr(a.c2, -1,1) = 'd';

 c1 |  c2  | c3
----+------+----
  1 | abcd |
  2 | efgh |
  3 | ijkl |
(3 行)
```

FUJITSU Enterprise Postgres 11の場合

Oracle query側で外部結合と抽出を実施します。

[例]

```
=# EXPLAIN SELECT a.c1, a.c2, b.c3 FROM ft1 a LEFT OUTER JOIN ft2 b ON a.c1 = b.c1 AND substr(a.c2, -1,1) = 'd';

                QUERY PLAN

-----------------------------------------------------------------------------------------------------

 Foreign Scan  (cost=10000.00..20000.00 rows=1000 width=66)

   Oracle query: SELECT /*5e03719c069bac004e84a38149e37775*/ r1."C1", r1."C2", r2."C3" FROM ("T1" r1 LEFT JOIN "T2" r2
ON (substr(r1."C2", -1, 1) = 'd') AND (r1."C1" = r2."C1"))

(2 行)

=# SELECT a.c1, a.c2, b.c3 FROM ft1 a LEFT OUTER JOIN ft2 b ON a.c1 = b.c1 AND substr(a.c2, -1,1) = 'd';

 c1 |  c2  |  c3
----+------+------
  1 | abcd | 1000
  2 | efgh |
  3 | ijkl |
(3 行)
```

対処方法

Oracleにプッシュダウンさせない場合、プッシュダウンさせない箇所の処理に共通テーブル式を使用して記述してください。

2.9 pgAdmin4に関する非互換

項目	移行前のバージョン				
	9.5	9.6	10	11	12
デスクトップアプリケーションからシステムトレイアプリケーションに変更	―	―	○	―	―
データベースとの接続が切断された時の動作の変更	―	―	○	―	―
設定の項目SQL エディタの名称をクエリツールに変更	―	―	○	―	―
クエリツールのクエリの履歴パネルでの表示内容の変更	―	―	○	―	―
SQLタブのスクリプト表示仕様の変更	―	―	○	―	―
関数の実行でエラーが発生した場合の出力メッセージの変更	―	―	○	―	―

◎:FUJITSU Enterprise Postgresによる非互換あり

○:OSSによる非互換あり

ー:非互換なし

2.9.1 デスクトップアプリケーションからシステムトレイアプリケーションに変更

非互換

FUJITSU Enterprise Postgres 11では、pgAdmin4をデスクトップアプリケーションからシステムトレイアプリケーションに変更します。これに伴い、ユーザーインターフェースはWebブラウザを起動して表示します。

なお、Microsoft Edgeが既定のWebブラウザーに設定されている環境では、ループバックによるネットワークアクセスが無効になっている場合、pgAdmin4を起動することができません。

対処方法

Microsoft EdgeでpgAdmin4を利用する場合は、Microsoft Edgeにおいて、ループバックによるネットワークアクセスを有効にしてください。また、Microsoft Edgeをループバック除外リストに登録してください。

2.9.2 データベースとの接続が切断された時の動作の変更

非互換

FUJITSU Enterprise Postgres 11では、データベースとの接続が切断された後で、データベースと通信を行う場合、警告ダイアログを表示するように変更します。

FUJITSU Enterprise Postgres 10以前の場合

データベースとの接続が切断された後で、データベースと通信を行う場合、自動で再接続します。

FUJITSU Enterprise Postgres 11の場合

データベースとの接続が切断された後で、データベースと通信を行う場合、警告ダイアログを表示しますが、再接続しません。

対処方法

再接続を行う場合は、“続行”ボタンを押下してください。

2.9.3 設定の項目SQL エディタの名称をクエリツールに変更

非互換

FUJITSU Enterprise Postgres 11では、設定の項目である“SQL エディタ”の名称を“クエリツール”に変更します。

対処方法

ありません。

2.9.4 クエリツールのクエリの履歴パネルでの表示内容の変更

非互換

FUJITSU Enterprise Postgres 11では、クエリツールのクエリの履歴パネルで表示されるSQLの実行時間において、実行に1分以上かかるSQLは秒単位で表示するよう変更します。

FUJITSU Enterprise Postgres 10以前の場合

実行に1分以上かかるSQLの実行時間は、分単位で表示します。

[例]

```
1 min
```

FUJITSU Enterprise Postgres 11の場合

実行に1分以上かかるSQLの実行時間は、秒単位で表示します。

[例]

```
1 min 45 secs
```

対処方法

ありません。

2.9.5 SQLタブのスクリプト表示仕様の変更

非互換

FUJITSU Enterprise Postgres 11では、関数ノードから大文字を含む関数を選択し、SQLタブを参照することで表示されるスクリプトにおいて、CREATE文の関数名をダブルクォートで括るように変更します。

FUJITSU Enterprise Postgres 10以前の場合

関数ノードから大文字を含む関数を選択し、SQLタブを参照することで表示されるスクリプトにおいて、CREATE文の関数名をダブルクォートで括りません。

[例]

```
CREATE OR REPLACE FUNCTION public.Function1(
    integer,
    integer)
    RETURNS integer
    LANGUAGE 'sql'

    COST 100
    IMMUTABLE STRICT
AS $BODY$select $1 + $2;$BODY$;
```

FUJITSU Enterprise Postgres 11の場合

関数ノードから大文字を含む関数を選択し、SQLタブを参照することで表示されるスクリプトにおいて、CREATE文の関数名をダブルクォートで括ります。

[例]

```
CREATE OR REPLACE FUNCTION public."Function1"(
    integer,
    integer)
    RETURNS integer
    LANGUAGE 'sql'
```

```
    COST 100
    IMMUTABLE STRICT
AS $BODY$select $1 + $2;$BODY$;
```

対処方法

ありません。

2.9.6 関数の実行でエラーが発生した場合の出力メッセージの変更

非互換

FUJITSU Enterprise Postgres 11では、RAISE文を含む関数の実行がエラーになった場合、RAISE文のINFO、NOTICE、WARNINGレベルのメッセージを出力するように変更します。

対処方法

必要に応じてpostgresql.confファイルのclient_min_messagesパラメータを変更してください。

2.10 pgauditに関する非互換

項目	移行前のバージョン				
	9.5	9.6	10	11	12
トリガ関数実行時に、監査ログにNEW、OLDの値を追加で出力するように変更	―	―	◎	◎	◎
パーティションテーブルまたはそのインデックスのオブジェクトタイプの変更	―	―	○	○	○
監査ログのコマンドタグの出力値を「UNKNOWN」から「???」に変更	―	―	○	○	○

◎:FUJITSU Enterprise Postgresによる非互換あり

○:OSSによる非互換あり

―:非互換なし

2.10.1 トリガ関数実行時に、監査ログにNEW、OLDの値を追加で出力するように変更

非互換

FUJITSU Enterprise Postgres 13では、pgaudit.log_parameterをonに設定している場合に、トリガ関数が監査ログにNEW、OLDの値を追加で出力します。

FUJITSU Enterprise Postgres 12以前の場合

以下の(18)で、New、OLDの値が出力されません。

[例]

```
AUDIT: SESSION,WRITE,2020-09-03 07:07:39 UTC,
        (1)     (2)    (3)
[local],9775,psql,k5user,postgres,3/536, 1, 2, INSERT, ,
(4)      (5)  (6)  (7)     (8)       (9)    (10)(11)(12)(13)
TABLE,public.trig_audit, ,
(14)  (15)     (16)
 "INSERT INTO trig_audit SELECT 'U', now(), user, OLD.*, NEW.*",
(17)
trig_audit AFTER ROW UPDATE 92027 trig_test trig_test public 0  f aaaa
(18)
```

FUJITSU Enterprise Postgres 13の場合

New、OLDの値が出力されます。

［例］

```
AUDIT: SESSION,WRITE,2020-09-03 07:07:39 UTC,
[local],9775,psql,k5user,postgres,3/536,1,2,INSERT,,TABLE,public.
trig_audit,,
"INSERT INTO trig_audit SELECT 'U', now(), user, OLD.*, NEW.*",
(bbb) (aaa) trig_audit AFTER ROW UPDATE 92027 trig_test trig_test
public 0  f aaaa
```

対処方法

トリガ関数が監査ログに出力する文字列を監視して動作するアプリケーションを使用している場合、NEW、OLDの値が出力されても動作するようにアプリケーションを修正してください。

2.10.2 パーティションテーブルまたはそのインデックスのオブジェクトタイプの変更

非互換

FUJITSU Enterprise Postgres 13では、パーティションテーブルまたはそのインデックスのオブジェクトタイプをUNKNOWNからTABLEまたはINDEXに変更します。これにより、下記の3点の影響があります。

1. pgauditのobject_typeパラメータにUNKNOWNを指定せず、TABLEまたはINDEXを指定していた場合に、パーティションテーブルまたはそのインデックスの監査ログが出力されるようになります。
2. pgauditのobject_typeパラメータにUNKNOWNのみを指定していた場合に、パーティションテーブルまたはそのインデックスに対するDML操作の監査ログが出力されていたものが、出力されなくなります。
3. パーティションテーブルまたはそのインデックスに対するDML操作の監査ログの出力内容について、そのリレーションのオブジェクトタイプを表す文字列として、以前はUNKNOWNと出力されていたものが、TABLEまたはINDEXと出力されるようになります。

対処方法

1. 出力される監査ログの量が増加する可能性があるため、必要に応じてディスクの容量を増加してください。
2. 必要に応じて、object_typeパラメータに、TABLEまたはINDEXを指定してください。
3. 監査ログに出力するオブジェクトタイプを監視して動作するアプリケーションを使用している場合、Nオブジェクトタイプの値が変更されても動作するようにアプリケーションを修正してください。

2.10.3 監査ログのコマンドタグの出力値を「UNKNOWN」から「???」に変更

非互換

FUJITSU Enterprise Postgres 13では、監査ログのコマンドタグの出力値として「UNKNOWN」と出力されていたものを、「???」と出力するように変更します。

対処方法

FUJITSU Enterprise Postgres 12までで、監査ログのコマンドタグの値として「UNKNOWN」を判別して処理を行っていたアプリケーションは、「???」を判別して処理を行うように修正してください。

オプション「-m,--maxlength」を用いて、クエリの最大文字数を設定することができます。

第3章 プログラム修正一覧

プログラム修正情報については、“プログラム修正一覧”を参照してください。

索　引

[数字]

13で追加された機能.....1

[は]

プログラム修正一覧.....168