



Migration Portal

リリース *5.0.0*

EnterpriseDB Corporation

2026 年 06 月 09 日

目次

第 1 章	Release notes	3
1.1	Migration Portal 5.0.0 release notes	3
1.2	Migration Portal 4.14.0 release notes	4
1.3	Migration Portal 4.13.0 release notes	5
1.4	Migration Portal 4.12.0 release notes	5
1.5	Migration Portal 4.11.0 release notes	6
1.6	Migration Portal 4.10.0 release notes	6
1.7	Migration Portal 4.9.0 release notes	7
1.8	Migration Portal 4.8.0 release notes	8
1.9	Migration Portal 4.7.1 release notes	8
1.10	Migration Portal 4.7.0 release notes	8
1.11	Migration Portal 4.6.0 release notes	9
1.11.1	サポート終了のお知らせ	10
1.12	Migration Portal 4.5.1 release notes	10
1.12.1	サポート終了のお知らせ	10
1.13	Migration Portal 4.5.0 release notes	11
1.13.1	サポート終了のお知らせ	12
1.14	Migration Portal 4.4.0 release notes	12
1.14.1	サポート終了のお知らせ	13
1.15	Migration Portal 4.3.0 release notes	14
1.16	Migration Portal 4.2.0 release notes	14
1.17	Migration Portal 4.1.0 release notes	16
1.18	Migration Portal 4.0.0 release notes	17
第 2 章	Supported platforms	19
2.1	サポートされているブラウザ	19
第 3 章	Using Migration Portal	21
3.1	Overview of the Migration Portal home page	21
3.2	Overview of the Migration Portal Projects page	22
3.3	Migration Portal quick start	24
3.4	AI Copilot	24
3.4.1	ユースケース	25
3.4.2	AI コパイロットを使用する	25
3.4.3	フィードバック	25
第 4 章	Migrating a database	31
4.1	Performing a schema extraction	31
4.1.1	スキーマ抽出方法	31
4.1.2	スキーマとオブジェクトのサポート	31

4.2	Performing a schema migration	40
4.2.1	オフライン移行	40
4.2.2	オンライン移行	41
4.3	Performing a data migration	45
4.3.1	高度なデータの移行	46
4.4	Performing a schema assessment	46
4.4.1	スキーマ評価を実行する	46
4.4.2	評価レポートを生成する	49
4.4.3	評価レポートを評価する	50
第 5 章	Known issues, limitations, and notes	55
5.1	Migration Portal	55
5.1.1	ラップされたオブジェクト	55
5.1.2	削除されたオブジェクト	55
5.1.3	ファイルエンコード	56
5.1.4	ホワイトラベルエラー	56
5.1.5	ALTER ステートメント	56
5.1.6	ユーザー、ロール、プロファイル、および付与	56
5.1.7	付与	56
5.1.8	特権	57
5.1.9	パスワード	57
5.1.10	プロファイル	57
5.1.11	Oracle のデフォルトの場合を使用する	58
5.1.12	不完全なオブジェクト定義	61
5.2	EDB DDL エクストラクタ	61
5.2.1	一般的な制限事項	61
5.2.2	「スナップショットが古すぎます」エラー	62
5.3	Oracle Data Pump ユーティリティ	63
5.4	AI コパイロット	64

移行ポータルは、Oracle データベーススキーマを EDB Postgres Advanced Server プラットフォームに移行するための Web ベースのサービスです。移行ポータルは、Oracle データベーススキーマを評価および分析し、以下を変換します。

- テーブル
- シークエンス
- コンストレイン
- インデックス LOB インデックスとマテリアライズドビューのインデックスを除く
- シノニム
- ビュー
- マテリアライズドビュー
- DB リンク
- 型と型本体
- トリガー
- ファンクション
- プロシージャ
- パッケージ
- ユーザー
- ロール
- プロフィール
- ロールとオブジェクトの付与

移行ポータルは、EDB Postgres Advanced Server と互換性のあるデータ定義言語 DDL ステートメントを生成します。

使いやすいポータルインターフェイスにより、評価と移行プロセスが簡素化されます。ポータルにログインし、[Migration Portal quick start](#) を起動します。

EnterpriseDB は、企業が既存のデータベースシステムを長年 Postgres に移行するのを支援してきました。詳細については、[EnterpriseDB website](#) を参照してください。

第1章 Release notes

Migration Portal のドキュメントでは、Migration Portal 5.0 の最新バージョンについて説明しています。

Version	Release date
<i>How to create good prompts</i>	27 May 2026
<i>How to create good prompts</i>	18 Dec 2025
<i>How to create good prompts</i>	01 Jul 2025
<i>How to create good prompts</i>	25 Nov 2024
<i>How to create good prompts</i>	20 Aug 2024
<i>How to create good prompts</i>	27 Jun 2024
<i>How to create good prompts</i>	21 May 2024
<i>How to create good prompts</i>	16 Apr 2024
<i>How to create good prompts</i>	16 Jan 2024
<i>How to create good prompts</i>	20 Nov 2023
<i>How to create good prompts</i>	19 Oct 2023
<i>How to create good prompts</i>	12 Jul 2023
<i>How to create good prompts</i>	18 May 2023
<i>How to create good prompts</i>	14 Feb 2023
<i>How to create good prompts</i>	08 Dec 2022
<i>How to create good prompts</i>	04 Oct 2022
<i>How to create good prompts</i>	04 Aug 2022
<i>How to create good prompts</i>	22 Apr 2022

1.1 Migration Portal 5.0.0 release notes

リリース日 2026 年 5 月 27 日

移行ポータル 5.0.0 の新機能、機能強化、バグ修正、およびその他の変更には次のものが含まれます。

Type	Description
Breaking change	EDB DDL エクストラクタをバージョン 3.0.0 に更新しました。スクリプトは、すべてのテーブル制約とインデックスを `CREATE TABLE` ステートメントのインラインではなく別の定義として抽出するようになったため、データ移行シナリオでの制約とインデックスを無効または除外することが簡単になります。古いエクストラクターバージョン 3.0.0 より前を使用して抽出された SQL ファイルは、パーサーロジックの変更のため、受け入れられなくなりました。移行ポータルは、プロジェクト ページから最新の `edb_ddl_extractor.sql` をダウンロードし、アップロードする前にスキーマを再抽出します。
Enhancement	Oracle 21c データベースから抽出した DDL ファイルをアップロードし、ファイルをアップロードするときにソースデータベースとして Oracle 21c を選択できるようになりました。将来のリリースでは、追加の Oracle 21c 変換が利用できるようになる予定です。
Enhancement	既存のターゲットデータベースへのオンラインスキーマの移行を実行できるようになりました。選択したターゲットデータベースが既に存在する場合、移行ポータルは、オンライン移行を適用する前に、選択されたスキーマが存在する場合を削除します。
Bug fix	お問い合わせ フォームに利用規約が表示されない問題を修正しました。
Bug fix	評価レポート の互換性のない機能へのリンクが壊れていた問題を修正しました。
Bug fix	オブジェクト依存関係グラフに多数のノードが含まれている場合、移行ポータルレポートの生成が応答しなくなる問題を修正しました。依存関係グラフが 5,000 ノードを超える場合、レポートの応答性を維持するために 5,000 ノードに切り捨てられるようになりました。
Bug fix	移行ポータルが `ALTER TYPE ADD/MODIFY` 属性ステートメントを無視するため、変更された定義の代わりに元のタイプ定義が使用される問題を修正しました。

1.2 Migration Portal 4.14.0 release notes

リリース日 2025 年 12 月 18 日

移行ポータル 4.14.0 の新機能、機能強化、バグ修正、およびその他の変更には次のものが含まれます。

Type	Description
Enhancement	EDB Postgres Advanced Server バージョン 18 のサポートを追加しました。これは、いくつかの新しい Oracle 互換性機能を導入しました。移行ポータルユーザーは、EPAS 18 に対して Oracle スキーマを評価し、その後、スキーマを EPAS 18 データベースに移行できるようになりました。
Enhancement	移行ポータルには、強化されたオブジェクト依存関係マッピングが含まれるようになりました。依存関係は階層ツリー構造で表示されるため、ユーザーは子オブジェクトのカスケード障害をトリガーする親オブジェクトをすばやく特定できます。この可視性により、より戦略的な再評価の優先順位付けが可能になり、移行効率が大幅に向上します。

次のページに続く

表 3 – 前のページからの続き

Type	Description
Enhancement	JSON 形式で移行レポートを生成できるようになりました。この構造化データのサポートにより、外部ツールおよび下流の自動ワークフローとのシームレスな統合が可能になります。
Bug fix	大容量または大きなサイズのスキーマファイルを処理するときにオフラインエクスポートファイルのダウンロードを防止する問題を解決しました。

1.3 Migration Portal 4.13.0 release notes

リリース日 2025 年 7 月 1 日

移行ポータル 4.13.0 の新機能、機能強化、バグ修正、およびその他の変更には次のものが含まれます。

Type	Description
Enhancement	EDB Postgres Advanced Server <i>SPL_CHECK</i> 拡張機能のサポートを追加しました。すべての <i>Procedures</i> 、 <i>Functions</i> 、 <i>Package Body</i> 、 <i>Type Body</i> 、および <i>Trigger</i> 定義も、 <i>SPL_CHECK</i> 拡張機能を使用して自動的に評価され、潜在的な実行時エラーを検出できるようになりました。詳細については、 :ref: `SPL チェック` <SPL チェック> を参照してください。
Bug fix	Oracle 11g ソースデータベースの <code>ORACLE_MAINTAINED</code> フラグに関連するエラーメッセージを発生させた <code>edb_ddl_extractor.sql</code> スクリプトのバグを修正しました。
Bug fix	<code>http.p</code> パッケージのプログラム呼び出しがターゲットデータベースバージョン 15、16、および 17 と互換性がないと誤って識別したバグを修正しました。
Bug fix	DDL 解析プロセスによって大量のエラーが検出されたときに、プロジェクト評価が失敗する問題を修正しました。

1.4 Migration Portal 4.12.0 release notes

リリース日 2024 年 11 月 25 日

移行ポータル 4.12.0 の新機能、機能強化、バグ修正、およびその他の変更には次のものが含まれます。

Type	Description
Enhancement	EDB Postgres Advanced Server バージョン 17 のサポートを追加しました。これは、いくつかの新しい Oracle 互換性機能を導入しました。移行ポータルユーザーは、EPAS 17 に対して Oracle スキーマを評価し、その後、スキーマを EPAS 17 データベースに移行できるようになりました。
Enhancement	評価通知メッセージは、パーティション化テーブルに多数のパーティションが含まれている場合に発生する可能性のある遅延についてユーザーに通知するようになりました。

次のページに続く

表 5 – 前のページからの続き

Type	Description
Enhancement	ERH-2107 repair handler が強化され、`CREATE INDEX` ステートメントから `COMPUTE STATISTICS` キーワードを削除しました。
Bug fix	まれに解析失敗が発生したため、移行ポータルがオブジェクトの DDL の一部をスキップした問題を修正しました。
Bug fix	ターゲット DDL に新しい行のない長いコード行がある場合、 Reassess ボタンが移動されるというマイナーな UI の問題を修正しました。
Bug fix	`PUBLIC SYNONYMS` が抽出されない EDB DDL エクストラクタスクリプトの問題を修正しました。修正は、移行ポータルからダウンロードできる EDB DDL エクストラクタスクリプトの新しいバージョン 2.4.5 に含まれています。
Security fix	ソフトウェアライブラリを更新し、その他のセキュリティ強化を実装しました。

1.5 Migration Portal 4.11.0 release notes

リリース日 2024 年 8 月 20 日

移行ポータル 4.11.0 の新機能、機能強化、バグ修正、およびその他の変更には次のものが含まれます。

Type	Description
Enhancement	ロゴ、フォント、レイアウトのスタイルを更新して、ルックアンドフィールを向上させました。
Security fix	ソフトウェアライブラリを更新し、その他のセキュリティ強化を実装しました。
Bug fix	パブリックシノニムの're レーションが存在しない' エラーを修正しました。
Bug fix	互換性のある `MERGE` および `UNISTR` ステートメントのエラーが報告されていた問題を修正しました。
Bug fix	プロジェクトに大量のオブジェクトが含まれている場合、レポートの生成に時間がかかる問題を修正しました。
Bug fix	タブを変更するときに AI Copilot で観察された自動スクロールの問題を修正しました。
Bug fix	データベースリンク名で「@」サインを使用できない問題を修正しました。
Bug fix	複数のドットを含むデータベースリンク名が誤って引用される問題を修正しました。
Bug fix	スキーマ評価の後、移行ポータルがプロジェクトのロードに時間がかかる問題を修正しました。
Bug fix	大規模なソース DDL ファイルの評価を妨げた問題を修正しました。

1.6 Migration Portal 4.10.0 release notes

リリース日 2024 年 6 月 27 日

移行ポータル 4.10.0 の新機能、機能強化、バグ修正、およびその他の変更には次のものが含まれます。

Type	Description
Enhancement	互換性のない DDL オブジェクトのユーザー修復での Postgres `crosstab` などの Postgres `tablefunc` 拡張機能のサポートを追加しました。`crosstab` ファンクションは、Postgres では使用できない Oracle `PIVOT` 演算子の回避策として一般的に使用されます。
Enhancement	Oracle `PIVOT` オペレーターの回避策としての Postgres `tablefunc` 拡張 `crosstab` ファンクションの使用を説明する移行ポータルナレッジベースエントリを追加しました。
Bug fix	Portal Wiki タブを選択した後に Projects タブを選択すると、移行ポータルが更新される問題を修正しました。
Bug fix	AI Copilot のエラー処理機能を向上させました。バックエンドサーバーから返されたエラーに関するより適切な情報を提供するようになりました。
Bug fix	AI Copilot の垂直スクロールを修正して、新しい情報が表示されるとスクロールするようにしました。
Bug fix	別のタブに移動したときに AI Copilot が進行中のストリーミング応答を失う原因になったエラーを修正しました。
Bug fix	[AI Copilot] タブから移動するときにワークスペースページの [Quick Help] セクションが空白になる原因となったエラーを修正しました。
Bug fix	AI コパイロットを改善して、Enter と修飾キー Control、Command、Option、Shift キーを使用して AI Copilot テキストボックスへの改行の追加をサポートしました。

1.7 Migration Portal 4.9.0 release notes

リリース日 2024 年 5 月 21 日

移行ポータル 4.9.0 の新機能、機能強化、バグ修正、およびその他の変更には次のものが含まれます。

Type	Description
Feature	データベースの移行に関連する質問をし、特定の Oracle から EDB Postgres Advanced Server スキーマの非互換性問題を解決するための提案を要求できる AI コパイロット機能を追加しました。詳細については、 Migration Portal AI Copilot documentation を参照してください。
Enhancement	EDB Postgres Advanced Server でサポートされていないため、`PARTITIONED TABLE` および `MVVIEW` 定義から `PARALLEL/NOPARALLEL` 句を削除する修復ハンドラー ERH-2108 を追加しました。
Bug fix	移行ポータルが `TABLE` 定義から `XMLTYPE` 列プロパティ句を削除できない問題を修正しました。この句は、EDB Postgres Advanced Server ではサポートされていません。
Bug fix	解析失敗のため DDL ファイルの最後の部分が削除される問題を修正しました。

1.8 Migration Portal 4.8.0 release notes

リリース日 2024 年 4 月 16 日

移行ポータル 4.8.0 の新機能、機能強化、バグ修正、およびその他の変更には次のものが含まれます。

Type	Description
Enhancement	EDB Postgres Advanced Server は `VARRAY` 列プロパティをサポートしていないため、`CREATE TABLE` ステートメントから `VARRAY` プロパティ列句を削除する新しい修復ハンドラー ERH-2106 を追加しました。
Enhancement	EDB Postgres Advanced Server がサポートしていないため、`CREATE TABLE` ステートメントから `USING INDEX COMPUTE STATISTICS` 原因を削除する新しい修復ハンドラー ERH-2107 を追加しました。
Security fix	移行ポータルが使用する PostgreSQL JDBC ライブラリを更新して、 CVE-2024-1597 で特定されたセキュリティ脆弱性に対処しました。
Bug fix	大規模な DDL の解析失敗により、ソース DDL からワードがスキップされる問題を修正しました。
Bug fix	特殊な Unicode 文字を含む DDL ファイルが評価に失敗し、ユーザーがプロジェクトを表示できない問題を修正しました。

1.9 Migration Portal 4.7.1 release notes

リリース日 2024 年 1 月 16 日

移行ポータル 4.7.1 の新機能、機能強化、バグ修正、およびその他の変更には次のものが含まれます。

Type	Description
Bug fix	ソースに空白の Unicode 文字を含む DDL を完全に解析する問題を修正しました。
Security fix	複数のソフトウェアライブラリを更新し、その他のセキュリティ強化を実装しました。

1.10 Migration Portal 4.7.0 release notes

リリース日 2023 年 11 月 20 日

移行ポータル 4.7.0 の新機能、機能強化、バグ修正、およびその他の変更には次のものが含まれます。

Type	Description
Enhancement	EDB Postgres Advanced Server バージョン 16 のサポートを追加しました。

[次のページに続く](#)

表 11 – 前のページからの続き

Type	Description
Security fix	移行ポータルが使用する snakeyaml ライブラリを更新して、 CVE-2018-1000613 および CVE-2022-25857 で特定されたセキュリティ脆弱性に対処しました。

1.11 Migration Portal 4.6.0 release notes

リリース日 2023 年 10 月 19 日

移行ポータル 4.6.0 の新機能、機能強化、バグ修正、およびその他の変更には次のものが含まれます。

Type	Description
Enhancement	多数の Oracle オブジェクトを含む大規模なスキーマファイルのスキーマ解析フェーズのパフォーマンスの向上。
Enhancement	移行ポータルは、`DEFAULT` プロファイルの評価と移行をサポートするようになりました。
Enhancement	`XMLTable` 呼び出しに単一の引数がある場合、`XMLTable` を `XMLParse` に置き換える新しい修復ハンドラー ERH-2104 が追加されました。
Enhancement	`SELECT` ステートメント内の `IN` 句の値を括弧で囲む新しい修復ハンドラー ERH-2105 が追加されました。
Bug Fix	repair handler によって `REFERENCING` 句で更新された NEW および OLD 変数名が、トリガー定義の `WHEN` 句でも更新されない ERH-2030 repair handler の問題を修正しました。
Bug Fix	ERH-2065 repair handler を修正して、`SUBPARTITION TEMPLATE` 句から `INMEMORY` および `NO INMEMORY` 句を削除します。
Bug Fix	一部のシナリオで適用されなかった ERH-2002 修復ハンドラーの問題を修正しました。
Bug Fix	一部のシナリオで適用されなかった ERH-2009 修復ハンドラーの問題を修正しました。
Bug Fix	PL/SQL オブジェクト内での `UTL_HTTP.END_OF_BODY` 例外の発生が、解析フェーズで処理されない問題を修正しました。
Bug Fix	`GRANT EXECUTE ON DIRECTORY` ステートメントが、アップロードされたソース DDL スクリプトから削除されない問題を修正しました。
Bug Fix	`XMLRoot` ファンクションが EDB Postgres Advanced Server でサポートされているにもかかわらず、非互換性として誤ってフラグが付けられた問題を修正しました。
Bug Fix	一部のシナリオで `OutOfMemoryError` 例外のため、評価が失敗する問題を修正しました。
Bug Fix	スキーマファイルでの `REMARK`、`HOST`、および `PROMPT` キーワードの使用に関連する解析の問題を修正しました。
Bug Fix	プロジェクトで Use Oracle のデフォルトのケースをスキーマ移行に使用するオプションが選択されている場合、チェック制約内の列名が二重引用符で囲まれなかった問題を修正しました。

次のページに続く

表 12 – 前のページからの続き

Type	Description
Bug Fix	Oracle バージョンチェックと実際のスキーマ抽出を実行する前に権限チェックを実行するように EDB DDL エクストラタスクリプトを修正しました。
Bug Fix	DDL パーサーを更新して、スキーマのないロールに対して生成された `GRANT RESOURCE` ステートメントをコメントアウトしました。

1.11.1 サポート終了のお知らせ

EDB Postgres Advanced Server バージョン 11 の標準サポートは、2023 年 11 月 20 日までに終了します。EDB Postgres Advanced Server バージョン 11 のサポートは、2023 年 5 月 20 日に移行ポータルから削除され、これ以降に新しい EDB Postgres Advanced Server バージョン 11 プロジェクトが作成されないように日付。

利用可能なサポートの期間を最大化し、新機能、特に Oracle との互換性を追加する機能を利用するには、最新の EDB Postgres Advanced Server バージョンを移行ポータルプロジェクトのターゲットとして選択することをお勧めします。各 EDB Postgres Advanced Server メジャーバージョンのサポート期間の詳細については、[EDB Platform Compatibility page](#) を参照してください。

1.12 Migration Portal 4.5.1 release notes

リリース日 2023 年 7 月 12 日

移行ポータル 4.5.1 の新機能、機能強化、バグ修正、およびその他の変更には次のものが含まれます。

Type	Description
Bug fix	何千ものテーブルを含む大規模なスクリプトを処理するときに、メモリの過剰使用が原因で評価フェーズがスタックする問題を修正しました。
Bug fix	[Add suffix to make object names unique] オプションが選択され、データベースオブジェクト名に `\$` 文字が含まれる場合、評価が失敗する問題を修正しました。

1.12.1 サポート終了のお知らせ

EDB Postgres Advanced Server バージョン 11 の標準サポートの終了は 2023 年 11 月 20 日に予定されているため、新しい移行ポータルプロジェクトの 2023 年以降にサポートされる EDB Postgres Advanced Server のより新しいバージョンを選択します。EDB Postgres Advanced Server 11 のサポートは、2023 年 5 月 20 日に移行ポータルから削除され始める予定で、この日以降に新しい EDB Postgres Advanced Server 11 プロジェクトが作成されないようにします。

利用可能なサポートの期間を最大化し、新機能、特に Oracle との互換性を追加する機能を利用するには、最新の EDB Postgres Advanced Server バージョンを移行ポータルプロジェクトのターゲットとして選択することをお勧めします。各 EDB Postgres Advanced Server メジャーバージョンのサポート期間の詳細については、[EDB Platform Compatibility](#) ページを参照してください。

1.13 Migration Portal 4.5.0 release notes

リリース日 2023 年 5 月 18 日

移行ポータル 4.5.0 の新機能、機能強化、バグ修正、およびその他の変更には次のものが含まれます。

Type	Description
Enhancement	移行ポータルの移行の複雑さと機能分析機能は、以前にキャプチャされなかった Oracle 機能の存在をキャプチャおよび報告するようになりました。 <i>SELECT FOR UPDATE OF</i> 、 <i>ANY with = <> != operators</i> 、 <i>ANY with > < >= <= operators</i> 、 <i>IS JSON CHECK CONSTRAINT</i> 、 <i>LISTAGG ON OVERFLOW</i> 、 <i>UNDER IN TYPE DEFINITION</i> 、 <i>REF IN TYPE SPECIFICATIONS</i> 、 <i>OVERRIDING SUBPROGRAM DECLARATION IN TYPE</i> 、 <i>OF IN OBJECT VIEW CLAUSE</i> 。
Enhancement	EDB DDL エクストラクタユーティリティは、スキーマを抽出するときに必要な場合にのみ `GRANT` ステートメントを抽出するオプションを提供するようになりました。以前は、`GRANT` ステートメントが常に抽出されました。ここで、それらを抽出することを選択できます。デフォルトでは、`GRANT` ステートメントは抽出されません。
Enhancement	移行ポータルは、プロジェクトが通常のサポートの終了が近づいている EDB Postgres Advanced Server データベースバージョンをターゲットとしている場合を示し、ターゲットデータベースのサポートが移行ポータルから削除されると、プロジェクトのそれ以上の更新を制限します。
Enhancement	移行ポータルには、プロジェクトの作成日と、プロジェクトが作成されたときに使用された移行ポータルのバージョンが、移行ポータルのユーザーインターフェイスの 概要 セクションに表示されるようになりました。
Enhancement	移行ポータルは、ターゲットデータベースバージョン 11~14 の互換性のないファンクションとして `SQLERRM()` ファンクションを検出するようになりました。
Enhancement	移行ポータルに、新しい修復ハンドラー ERH-2103 があり、 <i>= ANY</i> 演算子を <i>IN</i> に、 <i>!= ANY</i> 演算子を <i>NOT IN</i> に置き換えます。
Enhancement	Repair handler ERH-2100 は、EDB Postgres Advanced Server でサポートされていない <i>WRITE ON DIRECTORY</i> Grant 特権を検出および削除するようになりました。
Bug fix	`IMPORT FULL` や `EXPORT FULL` のような一部のサポートされていないシステム付与が移行ポータルによって削除されない問題を修正しました。
Bug fix	`DBMS_UTILITY.FORMAT_CALL_STACK` および `DBMS_UTILITY.LNAME_ARRAY` ファンクションが EDB Postgres Advanced Server でサポートされている場合、互換性がないとして誤ってフラグが付けられていた問題を修正しました。
Bug fix	別のロールに割り当てられているロールから `CREATE ROLE` ステートメントが抽出されない問題を修正しました。
Bug fix	不要な `SET SEARCH_PATH=GLOBAL_OBJECTS` ステートメントをプロジェクトのエクスポート SQL ファイルから削除しました。
Bug fix	特定のユースケースのファイル解析フェーズ中に移行ポータル評価プロセスがエラーアウトする原因となっていた問題を修正しました。

次のページに続く

表 14 – 前のページからの続き

Type	Description
Bug fix	Oracle Data Pump ユーティリティによって抽出され、アップロードされたソース DDL に含まれているネストされたテーブルが解析失敗を発生させていた問題を修正しました。EDB Postgres Advanced Server は、ネストされたテーブルをサポートしていません。その結果、これらは移行ポータルによる処理と評価の前にソース DDL から削除されます。
Security fix	移行ポータルが使用する ejs ライブラリ Node.js 用が更新されて、 CVE-2022-29078 および CVE-2021-23566 で特定されたセキュリティ脆弱性に対処しました。
Security fix	移行ポータルが使用する terserNPM ライブラリが更新されて、 CVE-2022-25858 で特定されたセキュリティ脆弱性に対処しました。

1.13.1 サポート終了のお知らせ

EDB Postgres Advanced Server バージョン 11 の標準サポートの終了は 2023 年 11 月 20 日に予定されているため、新しい移行ポータルプロジェクトの場合、2023 年以降にサポートされる EPAS のより新しいバージョンを選択します。EPAS 11 のサポートは、2023 年 5 月 20 日に移行ポータルから削除され始める予定で、この日以降に新しい EPAS 11 プロジェクトが作成されないようにします。

利用可能なサポートの期間を最大化し、新機能、特に Oracle との互換性を追加する機能を利用するには、移行ポータルプロジェクトのターゲットとして最新の EPAS バージョンを選択することをお勧めします。各 EDB Postgres Advanced Server メジャーバージョンのサポート期間の詳細については、[EDB Platform Compatibility](#) ページを参照してください。

1.14 Migration Portal 4.4.0 release notes

リリース日 2023 年 2 月 14 日

移行ポータル 4.4.0 の新機能、機能強化、バグ修正、およびその他の変更には次のものが含まれます。

Type	Description
Enhancement	移行ポータルは、EDB EDB Postgres Advanced Server EDB Postgres Advanced Server バージョン 15 をサポートするようになりました。
Enhancement	移行ポータルが強化され、移行プロジェクトのオブジェクト名に Oracle のデフォルトの大文字と小文字を保持するオプションを提供します。Postgres では、オブジェクト名のデフォルトの大文字は小文字です。Oracle では、大文字です。ほとんどの場合、データベースを Postgres に移行するときは、Postgres のデフォルトのケースを使用する必要があります。ただし、この新しいオプションは、データベースクエリで引用符で囲まれた大文字の識別子を特に使用するアプリケーションのサポートを向上させるために提供されています。新しい機能により、すべてのオブジェクトを引用符で囲まれた大文字の名前で作成できますが、初期実装はプロシージャルコードで指定されたオブジェクト名参照を更新しないことに注意してください。

次のページに続く

表 15 – 前のページからの続き

Type	Description
Enhancement	移行ポータルに移行の複雑さと機能分析機能が強化され、以前はキャプチャされていなかった特定の Oracle 機能の存在をキャプチャおよび報告します。これには、外部結合の条件としての `BETWEEN` 句の使用に対応する `OUTER JOIN WITH BETWEEN` 機能が含まれますクエリーで。
Enhancement	移行ポータルは、この機能が EDB Postgres Advanced Server でサポートされていないため、`CREATE VIEW` ステートメントから `CONSTRAINT` 句を削除する新しい ERH-3008 修復ハンドラーで強化されました。この機能強化の一環として、スキーマ評価レポートの Feature Analysis セクション関連機能が更新され、'既存の使用可能な回避策を含む機能' の分類で `VIEW COLUMN CONSTRAINT` オカレンスを識別および報告します。
Enhancement	ERH-2057 repair handler が強化され、`LONG` データ型から精度部分を削除しました。
Bug fix	EDB DDL エクストラクタが更新され、以前は取得されていなかった `ROLE` から `ROLE` 許可が抽出されました。
Bug fix	日付リテラルを指定するために使用される場合、`VIEW` の DDL が `DATE` キーワードの後で切り捨てられる問題を修正しました。
Bug fix	列名として `EXTENDED` で指定されたテーブル列の `DEFAULT` 句が削除される問題を修正しました。
Bug fix	ユーザーがオブジェクトを手動で修復した後、評価レポートに誤ったオブジェクト数が表示されていた問題を修正しました。
Bug fix	Oracle `PIPE ROW` 構文から Postgres `RETURN NEXT` 構文に変換した後、スキーマ名プレフィックスで指定された行タイプオブジェクトからスキーマ名が誤って削除される ERH-2059 修復ハンドラーの問題を修正しました。
Other	単一の引数を使用した `TO_TIMESTAMP_TZ()` ファンクション呼び出しを移行するための回避策を含む移行ポータルナレッジベースエントリを追加しました。これは EDB Postgres Advanced Server でサポートされていません。
Other	EDB Postgres Advanced Server バージョン 10 のサポートが削除されました。ユーザーは、新しい移行プロジェクトのターゲットデータベースとして EDB Postgres Advanced Server 10 を選択できなくなります。また、ユーザーは新しいスキーマをアップロードし、既存の EDB Postgres Advanced Server 10 プロジェクトのオブジェクトを編集および再評価することもできません。

1.14.1 サポート終了のお知らせ

EDB Postgres Advanced Server バージョン 11 の標準サポートの終了は 2023 年 11 月 20 日に予定されているため、新しい移行ポータルプロジェクトには、2023 年以降にサポートされる EPAS のより新しいバージョンを選択する必要があります。EPAS 11 のサポートは、2023 年 5 月 20 日に移行ポータルから削除され始める予定で、この日以降に新しい EPAS 11 プロジェクトが作成されないようにします。

利用可能なサポートの期間を最大限に長くし、新機能、特に Oracle との互換性を提供する機能を利用するために、EDB は移行ポータルプロジェクトのターゲットとして最新の EPAS バージョンを選択することをお勧めします。各 EDB Postgres Advanced Server メジャーバージョンのサポート期間の詳細については、[EDB Platform Compatibility](#) ページを参照してください。

1.15 Migration Portal 4.3.0 release notes

リリース日 2022 年 12 月 8 日

移行ポータル 4.3.0 の新機能、機能強化、バグ修正、およびその他の変更には次のものが含まれます。

Type	Description
Enhancement	移行ポータルが強化され、ソース Oracle データベースからのユーザー、ロール、プロファイル、および付与の評価と移行をサポートします。
Enhancement	移行ポータルが更新され、ユーザー、ロール、プロファイル、付与の抽出のサポートを提供する EDB DDL エクストラクタの新しいバージョンへのリンクを提供します。移行ポータルは、古いバージョンの EDB DDL エクストラクタによって生成された DDL を処理および評価することができます。ただし、新しくサポートされたユーザー関連情報は、これらの古いバージョンでは抽出できません。
Enhancement	移行ポータルスキーマの移行複雑性スコアリングと機能分析機能が強化され、以前のリリースの場合と同様にバージョン 14 だけでなく、選択したターゲット EDB Postgres Advanced Server バージョンに基づくスコアと機能分析を提供します。
Enhancement	スキーマをターゲットデータベースに移行するための移行ポータルのユーザーインターフェイスが簡素化され、ステップ数と冗長オプションが削減されたオフラインおよびオンライン移行モードが有効になりました。
Enhancement	移行ポータルの移行の複雑さと機能分析機能が強化され、以前はキャプチャされていなかった Oracle 機能の存在をキャプチャおよび報告します。これには、 <i>CONNECT BY CONDITION</i> 、 <i>CONNECT BY NOCYCLE</i> 、およびビルトイン/SYS 所有ファンクション `GROUP_ID` が含まれます。
Security fix	移行ポータルが使用する jackson-databind ライブラリが更新され、 CVE-2022-42003 および CVE-2022-42004 で特定されたセキュリティ脆弱性に対処しました。
Bug fix	警告を回避し、ターゲット EDB Postgres Advanced Server データベースでサポートされている `MAXVALUE` の許可最大値に関する問題を解決するために、シーケンスの `MAXVALUE` の処理を修正しました。
Bug fix	Oracle Data Pump ユーティリティを使用して生成された抽出 DDL で見つかった仮想テーブル列に関連する `ALTER TABLE MODIFY` ステートメントの処理の問題を修正しました。
Bug fix	パッケージファンクション定義内で句が使用されている場合、`PIPELINED` 句機能が報告されない問題を修正しました。
Bug fix	特定の場合に互換性のある機能で検出および報告されないカタログビューに関連する問題を修正しました。

1.16 Migration Portal 4.2.0 release notes

リリース日 2022 年 10 月 4 日

移行ポータル 4.2.0 の新機能、機能強化、バグ修正、およびその他の変更には次のものが含まれます。

Type	Description
Enhancement	4.1.0 リリースより前に作成された既存のプロジェクトのスキーマ移行複雑度スコアの計算と機能分析の生成のサポートを追加しました。
Enhancement	スキーマ移行の複雑さのスコアリング機能を強化して、オブジェクトを手動で修復した後のスコアの更新に対応します。
Enhancement	スキーマ評価レポートの Feature Analysis セクションに、未解決のエラーを持つオブジェクトと、これらのオブジェクトで使用される互換性のない Oracle 機能のリストをリストする新しい「Unresolved Objects」テーブルを追加しました。
Enhancement	多数の Oracle 機能を含むプロジェクトで、スキーマ評価レポート生成のパフォーマンスが最大3倍向上しました。
Bug Fix	互換性のないビルトインパッケージおよび SYS スキーマ TYPE オブジェクトの使用をキャプチャおよびより正確に報告するように移行ポータルを修正しました。
Bug Fix	次の Oracle 機能の存在をより正確にキャプチャし、スキーマの移行の複雑さへの影響を評価するように移行ポータルを修正しました。USER DEFINED EXCEPTION RISING、USER DEFINED EXCEPTION HANDLING、XMLAGG、XMLQUERY、XMLTABLE、XMLPI、XMLPARSE、XMLROOT。
Bug Fix	DDL エクストラクタの新しいバージョンを作成して、CACHE キーワードで定義された Oracle テーブルの重複エントリが DDL ファイルに出力される問題を修正しました。移行ポータルが更新され、新しいバージョンの DDL エクストラクタへのリンクを提供します。移行ポータルは、古いバージョンの DDL エクストラクタによって生成された DDL を処理および評価することができます。
Bug Fix	PRAGMA EXCEPTION_INIT 宣言がスキーマ評価レポートの「Feature Analysis」セクションで互換性のない機能として誤って報告される問題を修正しました。
Bug Fix	LONG RAW が「Feature Analysis」セクションで互換性のない機能として誤って報告される問題を修正しました。
Bug Fix	ユーザー定義の例外宣言がスキーマ評価レポートの「Feature Analysis」セクションで互換性のない機能として誤って報告される問題を修正しました。
Bug Fix	BULK COLLECT INTO 句がスキーマ評価レポートの「Feature Analysis」セクションで互換性のない機能として誤って報告される問題を修正しました。
Bug Fix	スキーマ評価レポートの「Feature Analysis」セクションで NULLIF ファンクションが互換性のない機能として誤って報告される問題を修正しました。
Bug Fix	スキーマ評価レポートの「機能分析」セクションで、REGEXP_LIKE 句の回避策ステータスを「None / Needs Investigation」から「Exists」に更新しました。repair handler が現在利用でき、REGEXP_LIKE 句に自動的に適用されます。
Bug Fix	EDB Postgres Advanced Server でサポートされているものとサポートされていない Oracle の事前定義された PL/SQL 例外のリストを更新しました。
Bug Fix	ユーザーがオブジェクトのサブセットスキーマ、テーブル、ビューなどを選択し、ウィザードページを移動すると、ターゲット DDL オブジェクト選択ページのユーザー選択状態が失われる問題を修正しました。

1.17 Migration Portal 4.1.0 release notes

リリース日 2022 年 8 月 4 日

移行ポータル 4.1.0 の新機能、機能強化、バグ修正、およびその他の変更には次のものが含まれます。

Type	Description
Enhancement	移行ポータルスキーマ評価レポートには、アップロードされた DDL で見つかった Oracle コンストラクトつまり機能に関連する労力タイプと複雑さレベルに基づいて、移行の複雑さスコアを提供する新しい機能分析セクションが含まれるようになりました。この新しいセクションでは、DDL で見つかった互換性のない機能の概要も提供します。現在、Feature Analysis セクションは新しい移行プロジェクトに対してのみ生成され、このリリースより前に作成された既存のプロジェクトに対しては生成されないことに注意してください。
Enhancement	ユーザーは、プロジェクト内の使用可能なスキーマと基になるオブジェクトにフィルタリングを適用して、選択したオブジェクトタイプのサブセットのターゲットデータベース DDL を生成できるようになりました。さらに、ダウンロード用に単一の DDL SQL ファイルを作成するのではなく、zip アーカイブが作成されます。これには、選択したオブジェクトタイプごとに個別のファイルと、プロジェクトと選択したスキーマの統合スクリプトが含まれています。
Enhancement	Q 引用符で囲まれた文字列をドル引用符または E 引用符で囲まれた文字列に変換するための新しい修復ハンドラー ERH-3005 を追加しました。
Enhancement	ERH-2078 修復ハンドラーを強化して、パッケージヘッダーレベルでパイプライン特定の変換を適用します。
Enhancement	ERH-2034 修復ハンドラーを更新して、PACKAGE 内で提供される TYPE 定義から NULL を削除します。
Security fix	async、eventsources、h2、jsdom、および moment ライブラリを最新バージョンにアップグレードして、セキュリティ問題 CVE-2021-43138 、 CVE-2022-1650 、 CVE-2022-23221 、 CVE-2021-42392 、 CVE-2021-20066 、および CVE-2022-31129 に対処しました。
Bug fix	Oracle Data Pump ユーティリティを使用して生成されたソース Oracle DDL ファイルを処理するときに、一部のインデックスがスキップされるコーナーケースを修正しました。
Bug fix	ALTER TABLE ステートメントの ENABLE NOVALIDATE 句から ENABLE キーワードのみが削除される動作を修正し、full 句が削除されるようになりました。また、関連する repair handlers ERH-2003 および ERH-2005 を更新しました。
Bug fix	長期実行スキーマ評価と並列して実行されている他の評価のパフォーマンスに影響を与えていた同期スレッドの問題を修正しました。
Bug fix	Oracle Data Pump ユーティリティによって生成される CREATE TABLE ステートメントの DEDUPLICATE LOB 句により、DDL 解析関連のエラーが発生する問題を修正しました。

1.18 Migration Portal 4.0.0 release notes

リリース日 2022 年 4 月 22 日

移行ポータル 4.0.0 の新機能、機能強化、バグ修正、およびその他の変更には次のものが含まれます。

Type	Description
Enhancement	移行ポータルは、EDB DDL エクストラクタの使用を再度サポートするようになりました。DDL エクストラクタに関連する問題が解決され、DDL エクストラクタファイル形式の使用に関して導入された一時的な制限が削除されました。移行ポータルは、 documentation の指示に従って Oracle Data Pump ユーティリティの使用を引き続きサポートします。ただし、EDB DDL エクストラクタの使用をお勧めします。
Enhancement	スキーマ評価アルゴリズムは、いくつかのパフォーマンス関連の問題に対処し、全体的なスキーマ評価時間を削減するように最適化されています。
Enhancement	EDB Postgres Advanced Server がサポートしていないため、TABLE 定義から CACHE / NOCACHE キーワードを削除するための修復ハンドラーを追加しました。
Security fix	jackson-databind および h2 ライブラリは、セキュリティ問題 CVE-2020-36518、CVE-2022-23221、CVE-2021-42392 および CVE-2021-23463 に対処するために最新バージョンにアップグレードされました。
Bug fix	文法ルールを修正して、EDITIONABLE キーワードを削除する修復ハンドラーの実行をスキップする解析エラーを解決しました。
Bug fix	Oracle PIPELINED および PIPE ROW ファンクション呼び出しの適切な変換を実行するように修復ハンドラーを修正しました。
Bug fix	オブジェクト名が Unicode 文字に基づいている場合、大文字と小文字の区別が失われる問題を修正しました。
Bug fix	新しいユーザーが移行ポータルのランディングページを初めて開いたときに、はじめにウィザードが表示される問題を修正しました。
Bug fix	ターゲット EPAS DDL スクリプトでの変換中に VIEW 定義が切り詰められる問題を修正しました。
Bug fix	以前は、UTF-8 と互換性のない文字を含むアップロードされた DDL ファイルは、評価フェーズの開始時に失敗していました。移行ポータルは、SQL ファイルを UTF-8 として読み取るようになりました。非 UTF-8 ファイルがアップロードされた場合、UTF-8 と互換性のないすべての文字は置換文字 ` ` に変換され、評価は失敗せずに続行されます。
Bug fix	ソース Oracle DDL の名前のない制約の制約名前の割り当てを修正しました。
Bug fix	修復ハンドラーを更新して、Oracle Data Pump で生成されたソース DDL ファイルの OIDINDEX ストレージ句を削除しました。
Bug fix	COLLATE および COMPUTATION キーワードを識別子として使用できるようになりました。
Bug fix	単行コメントの一部として表示される複数行コメントの処理を修正しました。
Bug fix	ラップされたオブジェクトを適切に識別およびスキップするように文法を修正しました。
Bug fix	複数のプロジェクトを同時に実行しているときに発生した ConcurrentModificationException を修正しました。

次のページに続く

表 19 – 前のページからの続き

Type	Description
Bug fix	複数のスキーマにわたる一意のオブジェクト名の処理を修正しました。オブジェクト名は、異なるスキーマ間で一意である必要はありません。
Bug fix	特定の順序は必要ないため、一部の句の順序を無視するように create table 文法を修正しました。

第2章 Supported platforms

移行ポータルは、Oracle 11g、12c、18c、および 19c から EDB Postgres Advanced Server 14、15、16、17、および 18 への評価と移行をサポートします。移行ポータルは、次のブラウザーでサポートされています。

2.1 サポートされているブラウザー

最高のユーザーエクスペリエンスを得るには、Google Chrome ブラウザを使用することをお勧めします。Migration Portal は、次のブラウザーでもサポートされています。

Browser	Supported version
Apple Safari on Macintosh OS	14 and above
Google Chrome	68 and above
Microsoft Edge	42 and above
Mozilla Firefox	60 and above

第3章 Using Migration Portal

移行ポータルを使用すると、データベーススキーマを Oracle から EDB Postgres Advanced Server に簡単に移行できます。評価用のスキーマをアップロードし、すぐにフィードバックと提案を得ることができます。このポータルを使用すると、すべてのオブジェクトの評価済み DDL をダウンロードし、オンプレミスまたはクラウドで EDB Postgres データベースを作成できます。

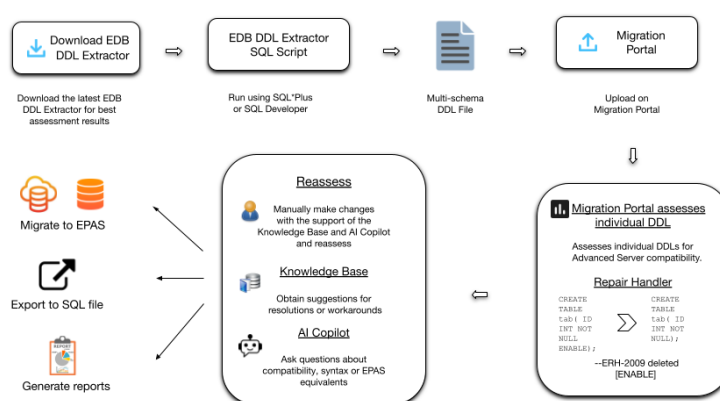


図 1: The assessment and migration process

注釈

現在、移行ポータルにはチームコラボレーション機能がありません。共同会議、画面共有、およびダウンロード可能な評価レポートを使用して結果を共有することをお勧めします。

Migration Portal にアクセスするには、Web ブラウザーを開き、[EnterpriseDB Website](#) に移動します。資格情報を使用してログインします。移行ポータルを使用して Oracle スキーマを移行する方法の概要については、[Migration Portal quick start](#) を参照してください。

3.1 Overview of the Migration Portal home page

移行ポータルのホームページから、移行ツールにすばやくアクセスできます。

Migration Portal ホームページから、次の Migration Portal 機能にアクセスできます。

- **プロジェクト** [プロジェクト] パネルには、評価されたプロジェクトのリストが表示されます。
- **プロジェクトの作成** + **New** を選択してプロジェクトを作成します。
- **プロジェクトの検索** **Search** ボックスを使用して、プロジェクトを検索します。

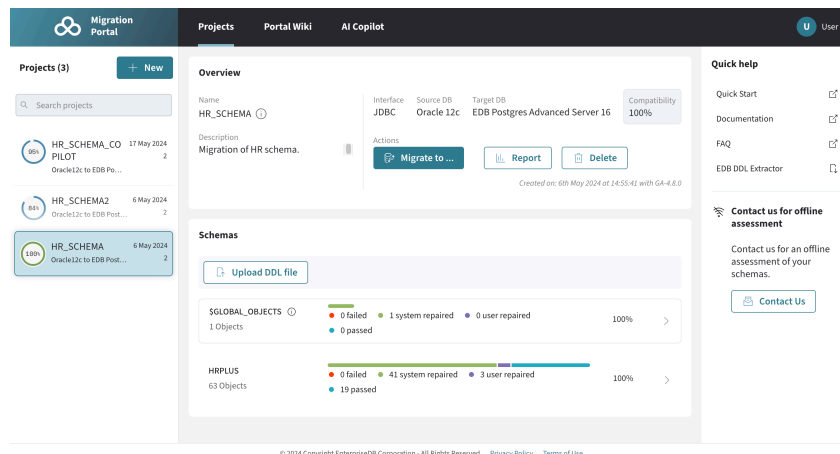


図 2: The Migration Portal home page.

- **概要** 概要パネルは、選択したプロジェクトに関する詳細を提供し、スキーマ評価後の互換性の割合が表示されます。
- **移行先** 移行先を選択して、EDB Postgres Advanced Server 互換の .sql ファイルをダウンロードするか、オンプレミスまたはクラウドでスキーマを EDB Postgres Advanced Server に移行します。詳細については、[Performing a schema migration](#) を参照してください。
- **レポート** スキーマ評価レポートを表示およびダウンロードするには、**レポート** を選択します。
- **Delete** 選択したプロジェクトを削除するには、**Delete** を選択します。
- **警告サイン** プロジェクトまたはスキーマの互換性が 70%未満であるか、複数回試行しても DDL が成功しない場合、警告メッセージが表示されます。
- **Upload DDL file** 新しい DDL ファイルをアップロードするには、**Upload new schema** を選択します。
- **Schemas** Schemas パネルには、アップロードされた DDL ファイルの評価結果が表示されます。
- **クイックヘルプ** クイックヘルプパネルには、いくつかの必要なリソースへのクイックリンクが含まれています。
- **Portal Wiki** Portal Wiki を選択して、製品情報やその他のヘルプガイドへのリンクにアクセスします。
- **AI Copilot** AI チャットインターフェイスと対話するには、**AI Copilot** を選択します。このインターフェイスは、DDL の互換性、Postgres クエリ構文、EDB Postgres Advanced Server 同等のものなどに関する情報を取得するのに役立ちます。AI Copilot を使用する前に、利用規約に同意して機能の使用をオプトインするように求められます。

3.2 Overview of the Migration Portal Projects page

Migration Portal Projects ページでは、移行に関する詳細情報を提供します。

次のリソースを使用して、移行プロジェクトに関する情報を収集します。

- **プロジェクトの互換性** プロジェクトの互換性シンボルは、評価されたスキーマの互換性の程度を示し、範囲を示す数値パーセンテージと色のグラデーションが表示されます。
- **Schema Count** プロジェクト内のスキーマの数を表示します。

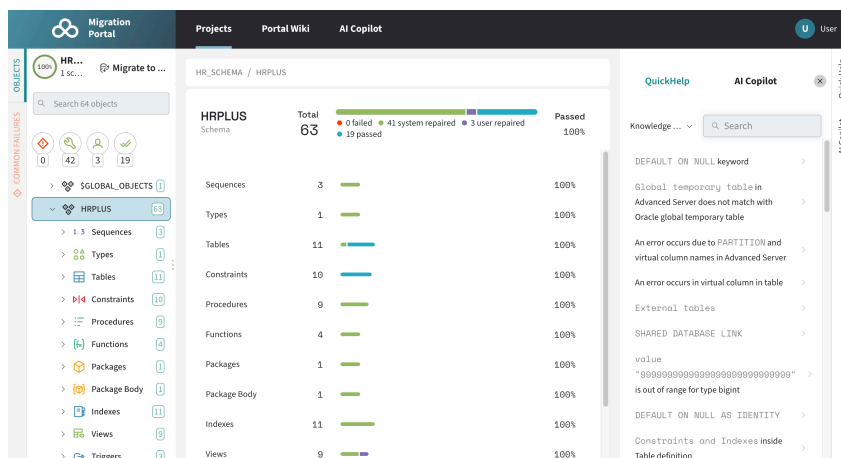


図 3: The Migration Portal Projects Page overview.

- **移行先** スキーマをオンプレミスまたはクラウド内の EDB Postgres Advanced Server に移行するには、**移行先** を選択します。
- **オブジェクトの検索** オブジェクトを検索するには、**Search** ボックスを使用します。
- **フィルタ** プロジェクトページの左側のパネルから、障害が発生した、システム修復、ユーザー修復、および自動的に渡されたオブジェクトをフィルタリングできます。1 つ以上のフィルタの組み合わせを選択して、情報を絞り込むことができます。
- **Objects** 選択したスキーマのオブジェクトを表示するタブ。
- **一般的な障害** 選択したスキーマの失敗したオブジェクトの理由を表示するタブ。

注釈

プロジェクトの一般的な障害の `.csv` ファイルをダウンロードできます。

- **Schema** 評価の結果を表示します。
- **ツールチップ** 結果セットにカーソルを合わせると、合格、失敗、および修復されたオブジェクトの数を含むツールチップが表示されます。
- **クイックヘルプ** クイックヘルプパネルには、ナレッジベース記事と修復ハンドラードキュメントへのリンクが表示されます。**Search** ボックスを使用して、ナレッジベースエントリまたは修復ハンドラードキュメントから特定の情報を検索します。
- **AI Copilot** AI インターフェイスと対話するには、**AI Copilot** を選択します。このインターフェイスは、DDL の互換性、Postgres クエリ構文、EDB Postgres Advanced Server 同等のものなどに関する情報を取得するのに役立ちます。AI Copilot を使用する前に、利用規約に同意して機能の使用をオプトインするように求められます。

3.3 Migration Portal quick start

移行ポータルを使用して Oracle スキーマを移行するには

1. スキーマのメタデータ情報を含む .sql ファイルを生成します。詳細については、[Performing a schema extraction](#) を参照してください。
2. EDB 資格情報を使用して [Migration Portal](#) にログインします。
3. 新しいプロジェクトを作成します。
4. スキーマのメタデータを使用して .sql ファイルをアップロードします。

注釈

大きな DDL ファイルをアップロードすると、ブラウザーがフリーズまたはクラッシュする場合があります。使用中のブラウザーとシステムリソースによって異なります。このシナリオでは、移行ポータルを別のブラウザーで実行するか、抽出するスキーマの数を減らし、そのファイルを移行ポータルにアップロードしてみてください。!!!

1. **作成と評価**を選択します。移行ポータルは、すべてのコンストラクトを確認し、修復アクションを実行して EDB Postgres Advanced Server との互換性を向上させ、手動介入を必要とする残りのエラーにフラグを立てます。
2. DDL オブジェクトたとえば、100 の成功率を示していないテーブルを確認します。
3. EDB Postgres Advanced Server と互換性のないオブジェクトを選択します。
1. EDB Postgres Advanced Server で互換性のないオブジェクトに考えられる回避策を検索して理解するには、ナレッジベースを参照するか、[AI Copilot](#) と対話します。.. Important:: 提案されたソリューションをすべてテストして、変換されたスキーマが予想通りに動作することを確認します。!!!
1. 必要な変更を加え、**Reassess** を選択して分析を再実行します。
2. スキーマのエラーを解決した後、変更した EDB 互換 DDL ファイルをダウンロードし、pgAdmin または PSQL クライアントなどのクライアントアプリケーションを使用し、スキーマを EDB Postgres Advanced Server に移行できます。クラウド上の EDB Postgres Advanced Server に移行することもできます。詳細については、[Performing a schema migration](#) を参照してください。

3.4 AI Copilot

AI Copilot は、AI 駆動のチャットボットツールであり、DDL ファイルのスキーマ評価中に発生する可能性のある問題を支援するように設計されています。サードパーティの AI サービスを使用し、EDB の製品ドキュメントとサポートナレッジベースの両方でトレーニングされて、ユースケース固有のソリューションを提供します。

チャットボットは移行ポータルに埋め込まれ、AI Copilot と対話できる簡単にアクセスできるインターフェイスを提供します。

3.4.1 ユースケース

次のようなトピックについて質問できます。

- 移行戦略
- Oracle と EDB Postgres Advanced Server または PostgreSQL DDL の互換性
- Oracle クエリの EDB Postgres Advanced Server または PostgreSQL 同等品
- 構文エラー
- 特定のプロシージャまたはファンクションの使用例

3.4.2 AI コパイロットを使用する

1. AI Copilot *Enabling AI Copilot* の使用を開始するには。
2. 質問を入力します。例については、*How to create good prompts* を参照してください。

3.4.3 フィードバック

高評価または低評価アイコンを選択して、特定の AI Copilot の回答に関する一般的なフィードバックを提供できます。

会話 ID をコピーしてサポートチームと共有することにより、より詳細なフィードバックを提供したり、問題を報告したりすることもできます。

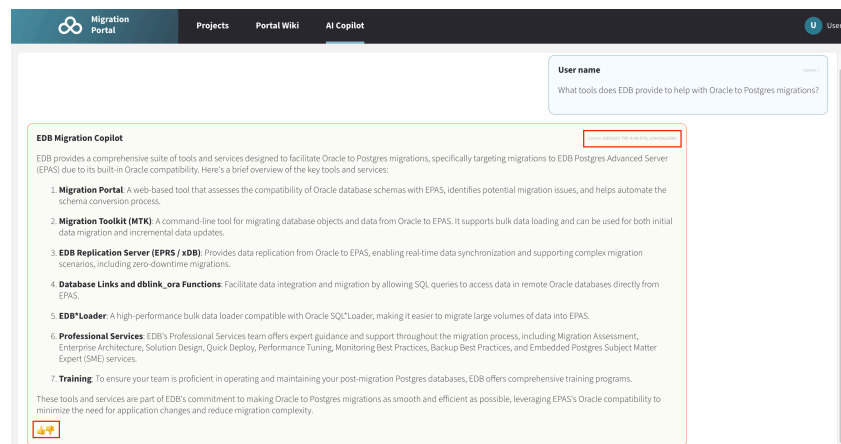


図 4: Feedback via conversation ID and icon

Enabling AI Copilot

AI Copilot を使用するには、最初にその利用規約に同意する必要があります。

AI コパイロットの有効化

1. **Migration Portal** の上部メニューバーで、**AI Copilot** を選択します。右側のペインから AI Copilot にアクセスすることもできます。このビューは、プロジェクトとスキーマを選択した後にのみ使用できます。

2. **Opt in for AI Copilot** を選択します。ポップアップウィンドウに、AI Copilot の使用に関する利用規約が表示されます。
3. 利用規約を読み、[**I have read and attempts to the tasks of the Agreement**] を選択し、ウィンドウを閉じます。

AI コパイロットが有効になっています。AI Copilot チャットを使用して、移行ポータル、移行戦略、構文エラーの解決などに関する質問ができるようになりました。使用例については、[How to create good prompts](#) を参照してください。

AI コパイロットの無効化

AI コパイロットはいつでも無効にできます。

1. Migration Portal の右上隅で、ユーザー名を選択します。
2. **設定** を選択します。
3. **AI コパイロットからオプトアウト** を選択します。

AI コパイロットは無効になっています。

How to create good prompts

AI Copilot が提供する回答の品質と精度は、プロンプトの品質によって異なります。AI Copilot は、質問、状況の詳細または詳細、および例を中心に大規模なコンテキストがある場合に最適に機能します。プロンプトをより詳細かつ正確に作成できれば、より良い回答が得られます。

他の言語モデルと同様に、AI Copilot でプロンプトエンジニアリング手法を使用して、クエリ入力を改善できます。

フルネーム、ユーザー名、パスワード、アドレスなどの機密情報をチャットボットに入力しないでください。

クエリーの例

重要

推奨されるソリューションを運用環境に適用する前に、制御されたテスト環境でソリューションをテストして、提案された修正が特定の移行要件と一致することを確認することを強くお勧めします。

一般的な質問をする

ニュートラルプロンプト

What tools are there to migrate databases?

AI Copilot は EDB 製品ドキュメントとナレッジベースでトレーニングされているため、この一般的な質問は EDB 製品提供のコンテキストで回答を提供します。ただし、チャットボットは、最も一般的なユースケース、Oracle データベースから Postgres データベースへの移行に焦点を当てていますが、これはあなたのユースケースではないかもしれません。

このプロンプトを改善するには、ソースデータベースとターゲットデータベースをより具体的にします。

プロンプトの向上

What tools does EDB provide to migrate from open-source Postgres to EDB Postgres_↵
→Advanced Server?

このプロンプトは、より正確な回答を生成し、その特定のユースケースをサポートする EDB 製品をリストします。

次に、さらに質問またはコンテキストを追加して、会話を続行できます。例

What is the difference between the Migration Portal and the Migration Toolkit?

構文エラーの解決策を見つける

移行ポータルは構文の問題を見つけると、問題のある行にフラグを立てます。



図 5: Syntax error message

クエリに関するより多くの知識を得るために一般的な質問をできます。例

What are type casts in Postgres?
How do I use `SYS.ODCIVARCHAR2LIST`?

AI Copilot に、構文エラーを含むクエリを修正するように依頼できます。この場合、問題のある行を含むクエリ全体をコピーします。

Can you correct the syntax of this query? `CREATE OR REPLACE VIEW HRPLUS.VW_STRING_↵
→LIST (STRING_VAL) AS SELECT COLUMN_VALUE FROM SYS.ODCIVARCHAR2LIST(a,b, c);`

または、Oracle ソースクエリーと同等の EDB Postgres Advanced Server ターゲットを要求できます。

```
Can you provide an EDB Postgres Advanced Server-compatible equivalent for `CREATE OR
→REPLACE FORCE EDITIONABLE VIEW "HRPLUS"."VW_STRING_LIST" ("STRING_VAL") AS SELECT
→"COLUMN_VALUE" FROM SYS.ODCIVARCHAR2LIST(a,b, c);`
```

どちらのプロンプトも、EDB Postgres Advanced Server 互換ビューを作成するための同様の構文を提案します。

```
CREATE OR REPLACE VIEW HRPLUS.VW_STRING_LIST (STRING_VAL) AS
SELECT UNNEST(ARRAY[a,b, c]) AS COLUMN_VALUE;
```

不正確な構文エラーの理解

場合によって、移行ポータルは、構文の問題に対して不正確またはあいまいなエラーメッセージを提供する場合があります。

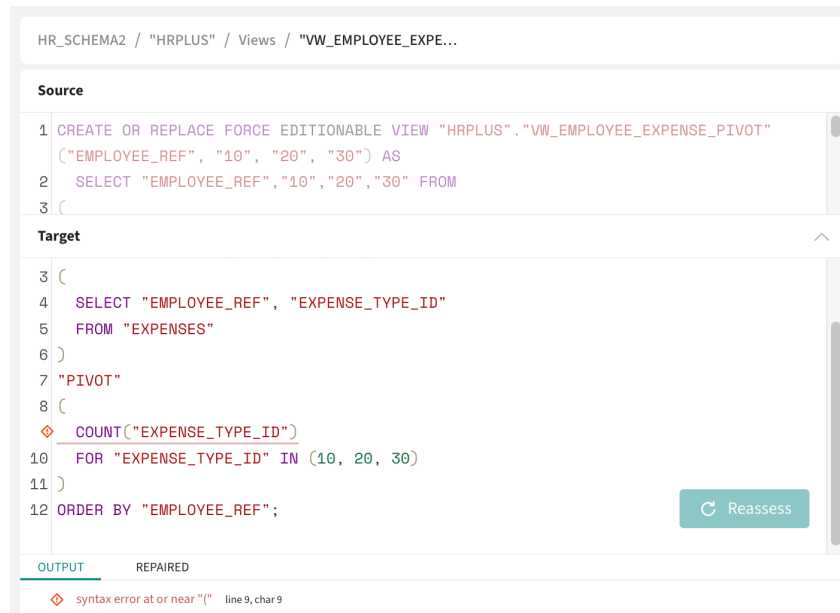


図 6: Imprecise syntax error message

AI Copilot に問題の説明を求めることができます。

```
What is the issue with the following argument when used in a Postgres query? `PIVOT (
→COUNT(expense_type_id) FOR expense_type_id IN (10, 20, 30) ) ORDER BY employee_ref;`
```

そして、同等のことを提案するには

```
Can you provide a Postgres-compatible version of this Oracle query? `SELECT EMPLOYEE_
→REF,"10","20","30" FROM ( SELECT employee_ref, expense_type_id FROM expenses )
→PIVOT ( COUNT(expense_type_id) FOR expense_type_id IN (10, 20, 30) ) ORDER BY
→employee_ref;`
```

これらのプロンプトの例に対して AI Copilot によって生成された提案は異なります。この場合、推奨されるパスは、提供された提案を使用して同等のクエリーを作成することです。次に、作成したクエリーに対

して元の動作をテストし、ターゲットクエリーが運用環境で使用する前に元の目的を満たすことを確認します。

第4章 Migrating a database

データベースを移行するには、次の手順を完了する必要があります。

1. *Performing a schema extraction* を実行します。
2. *Schema assessment checks* を実行します。
3. *Performing a schema migration* を実行します。
4. *Performing a data migration* を実行します。

4.1 Performing a schema extraction

4.1.1 スキーマ抽出方法

EDB または Oracle 抽出ツールを使用して、スキーマ抽出を実行できます。手順と例については、以下を参照してください。

- *Extracting schemas using the EDB DDL Extractor* 推奨する方法
- *Extracting schemas using Oracle Data Pump utilities*

EDB は、EDB DDL トラクターを使用してスキーマを抽出することをお勧めします。または、Oracle の Data Pump IMPDP / EXPDP ツールを使用して、Oracle データベースからスキーマをエクスポートできます。

警告

別の方法を使用してスキーマを抽出し、DDL ファイルを移行ポータルにアップロードすることを選択した場合、ポータルはファイルを正しく評価できない場合があります。

詳細については、[Known issues, limitations, and notes](#) を参照してください。

4.1.2 スキーマとオブジェクトのサポート

以下のリストと表は、移行ポータルでサポートされているおよび非サポートのスキーマとオブジェクトを示しています。

サポートされていないスキーマ

SQL ダンプファイルを生成するときに、これらの Oracle システムスキーマを除外します。

1	2	3	4	5
ANONYMOUS	APEX_PUBLIC_US	APEX_030200	APEX_040000	APEX_040200
APPQOSSYS	AUDSYS	CTXSYS	DMSYS	DBSNMP
DBSFUSER	DEMO	DIP	DMSYS	DVF
DVSYS	EXFSYS	FLows_FILES	FLows_020100	FRANCK
GGSYS	GSMAD- MIN_INTERNAL	GSMCATUSER	GSMROOTUSER	GSMUSER
LBACSYS	MDDATA	MDSYS	MGMT_VIEW	OJVM SYS
OLAPSYS	ORDPLUGINS	ORDSYS	ORDDATA	OUTLN
ORACLE_OCM	OWBSYS	OWBYSS_AUDIT	PDBADMIN	RMAN
RE- MOTE_SCHEDULE	SI_INFORMTN_SC	SPA- TIAL_CSW_ADMIN	SPA- TIAL_WFS_ADMIN	SQLTXADMIN
SQLTXPLAIN	SYS\$UMF	SYS	SYSBACKUP	SYS DG
SYSKM	SYSRAC	SYSTEM	SYSMAN	TSM SYS
WKPROXY	WKSYS	WK_TEST	WMSYS	XDB
XS\$NULL				

 注釈

EDB Postgres Advanced Server は、*pg_* で始まるスキーマ名をサポートしていません。

サポートされているオブジェクトタイプ

- テーブル
- シークエンス
- コンストレイン
- インデックス LOB インデックスとマテリアライズドビューのインデックスを除く
- シノニム
- ビュー
- マテリアライズドビュー
- DB リンク
- 型と型本体
- トリガー
- ファンクション
- プロシージャ

- パッケージ
- ユーザー
- ロール
- プロフィール
- ロールとオブジェクトの付与

i 注釈

列、テーブル、マテリアライズドビューのコメントもサポートされています。

サポートされていないオブジェクトタイプ

- エディション
- オペレーター
- スケジューラー
- LOB インデックスとマテリアライズドビューのインデックス
- XML スキーマ
- テーブルスペース
- ディレクトリ
- RLS ポリシー
- キュー*
- ライブラリー
- インデックスタイプ

* EDB Postgres Advanced Server はキューテーブルのサポートを提供していますが、移行ポータルは現在それをサポートしていません。ソース DDL のキューテーブルは、ソースおよびターゲット DDL オブジェクトとしてアップロードされません。

Extracting schemas using the EDB DDL Extractor

移行ポータルの **プロジェクト** ページから最新の EDB DDL エクストラクタスクリプトをダウンロードします。

i 注釈

システムとブラウザーのリソースの可用性によっては、非常に大きな SQL ファイルの場合、移行ポータルが応答しなくなる場合があります。この問題を解決するには、一度に抽出するスキーマの数を減らして、SQL ファイルのサイズを削減してください。

前提条件

EDB DDL エクストラクタスクリプトは、SQL Developer または SQL*Plus で実行できます。Oracle の DBMS_METADATA ビルトインパッケージを使用して、スキーマスクリプトの実行中に指定の下のさまざまなオブジェクトの DDL を抽出します。EDB DDL エクストラクタは、ポータルにアップロードされる DDL ファイルを作成し、EDB Postgres Advanced Server の互換性について分析されます。

注釈

CONNECT および `SELECT\_CATALOG\_ROLE` ロールと `CREATE TABLE` 特権が必要です。

バージョンの互換性

移行ポータル 5.0.0 以降では、EDB DDL エクストラクタバージョン 3.0.0 以降が必要です。移行ポータルは、アップロード時にエクストラクタバージョン 2.4.x 以前で生成された SQL ファイルを拒否します。

以前に古いエクストラクタバージョンを使用して DDL を抽出した場合は、移行ポータルのプロジェクトページから最新の `edb_ddl_extractor.sql` をダウンロードし、アップロードする前にスキーマを再抽出します。

SQL*Plus の場合

1. SQL*Plus に接続し、次のコマンドを実行します。

```
@edb_ddl_extractor.sql
```

スクリプトは、以降の手順で説明する詳細を提供するように求めます。

1. スクリプトを実行している Oracle ユーザーに必要な特権がある場合は、**Enter** を押してスクリプトの実行を続行します。
2. スキーマ名を指定します。複数のスキーマを抽出する場合は、コンマ ‘,’ を区切り文字として使用します。

注釈

現在のデータベースからすべてのユーザースキーマを抽出する場合は、抽出中にスキーマ名を記載しないでください。ただし、抽出するスキーマ名を記載することをお勧めします。

1. 抽出された DDL を保存するエクストラクタのパスまたはディレクトリを入力します。パスを指定しない場合、抽出された DDL は、SQL*Plus コマンドを実行したディレクトリに出力されます。
2. 他のスキーマから依存オブジェクトを抽出する場合は **yes**、抽出したくない場合は **no** と入力します。
3. **yes** と入力して他のスキーマから grant ステートメントを抽出するか、抽出したくない場合は **no** と入力します。

例

1. エクストラクタスクリプトを実行します。

```
@edb_ddl_extractor.sql
```

1. **Enter** を押します。
2. コンマで区切って3つのスキーマを指定します。

```
Enter a comma separated list of schemas to be extracted (Default all schemas): HR,
→SCOTT, FINANCE
```

1. 出力ファイルのパスを入力します。

Linux では、次を使用できます。

```
Location for output file (Default current location) : /home/oracle/extracted_ddls/
```

Windows でも同様に

```
Location for output file (Default current location) : c:\Users\Example\Desktop\
```

1. **yes** と入力して、他のスキーマの依存オブジェクトを抽出します。

WARNING:

Given schema(s) list may contain objects which are dependent on objects from other schema(s), not mentioned in the list. `Assessment may fail for such objects. It is suggested to extract all dependent objects together.

```
Extract dependent object from other schemas?(yes/no) (Default no / Ignored for all
→schemas option): yes
```

1. **yes** と入力して **grant** ステートメントを抽出します。

```
Extract GRANT statements?(yes/no) (Default no): yes
```

SQL Developer の場合

edb_ddl_extractor.sql スクリプトを SQL Developer にロードし、ソース Oracle データベースに接続した後、スクリプトを実行します。スクリプトが実行されたら、プロンプトに応答します。

1. データベース接続のユーザーが [スクリプト出力] タブにリストされている必要な特権を持っている場合、[**Yes**] を選択してスクリプトの実行を続行します。
2. スキーマのコンマ区切りリストを入力し、**OK** を選択します。
3. 出力ファイルのパスを入力し、**OK** を選択します。デフォルトは、DDL エクストラクタスクリプトの場所です。

4. **yes** または **no** と入力して依存オブジェクトを抽出し、**OK** を選択します。デフォルトでは、依存オブジェクトを抽出しません。
5. **yes** または **no** と入力して grant ステートメントを抽出し、**OK** を選択します。デフォルトでは、Grant ステートメントを抽出しません。

DDL エクストラクタ実行の出力は、[スクリプト出力] タブに表示されます。出力ファイルの名前は、スクリプト出力の **Extraction Completed** メッセージの後に表示されます。

注釈

SQL*Plus と SQL Developer の両方に単一のスキーマ名を入力することもできます。

次に、スクリプトは、ソースデータベースのオブジェクトタイプを反復します。タスクが完了すると、.SQL 出力は入力した場所たとえば `c:\Users\Example\Desktop\` に保存されます。

出力ファイルの理解

EDB DDL エクストラクタ 3.0.0 以降、抽出された SQL ファイルはテーブル定義をその制約から分離します。

- **CREATE TABLE** ステートメントには、列定義のみが含まれます。主キー、一意、外部キー、チェック、および **NOT NULL** 制約はインラインには含まれません。
- コンストレインは、抽出されたファイルの専用セクションに個別の **ALTER TABLE ... ADD CONSTRAINT** ステートメントとして出力されます。セクションは次のとおりです
 - **Primary Key Constraints**
 - **Unique Key Constraints**
 - **Foreign Keys**
 - **Check Constraints NOT NULL** コンストレインもカバーします
- 有効なコンストレインと無効なコンストレインの両方が抽出されます。

抽出したファイルのセクションマーカー **##** で始まる行は編集しないでください。移行ポータルは、それらに依存して出力を正しく解析します。

Extracting schemas using Oracle Data Pump utilities

移行ポータルでは、抽出されたスキーマを評価するために SQL ダンプファイルのメタデータのみが必要です。

前提条件

- 独自のスキーマをエクスポートする予定がある場合は、**DATAPUMP_IMP_FULL_DATABASE** ロールが割り当てられていることを確認します。それ以外の場合、独自のスキーマのみをエクスポートできます。
- テーブルスペースにオブジェクトを作成するための十分なテーブルスペースクォータがあることを確認します。

- SQL *Plus で、サーバーファイルシステム上のディレクトリを指すディレクトリオブジェクトを作成します。例 **Linux** の場合

```
create directory DMPDIR as /sample/file/path;
```

Windows の場合

```
create directory DMPDIR as c:\sample\file\path\;
```

- SQL*Plus では、Data Pump ユーティリティを実行しているユーザーに読み取り書き込み権限を付与します。

```
grant read,write on directory DMPDIR to <Username>;
```

次のいずれかの手順を実行します。

- データベース内の 1 つ以上のスキーマを抽出します
- データベース内のすべてのスキーマを抽出します

データベース内の 1 つ以上のスキーマを抽出します

1. expdp コマンドを実行する前に、サーバー上に .par 拡張子たとえば export.par を持つファイルを作成します。属性と値をファイルに追加します。

```
CONTENT=metadata_only
INCLUDE=SYNONYM
INCLUDE=SEQUENCE
INCLUDE=TYPE
INCLUDE=TABLE
INCLUDE=CONSTRAINT
INCLUDE=PROCEDURE
INCLUDE=FUNCTION
INCLUDE=PACKAGE
INCLUDE=INDEX
INCLUDE=MATERIALIZED_VIEW
INCLUDE=VIEW
INCLUDE=TRIGGER
INCLUDE=DB_LINK
INCLUDE=USER
INCLUDE=GRANT
INCLUDE=SYSTEM_GRANT
```

このコマンドの属性と値は、次のオプションを指定します。- CONTENT=metadata_only は、スキーマのメタデータのみを抽出します。- INCLUDE= は、抽出に含めるデータベースオブジェクトタイプを指定します。移行ポータルでサポートされているデータベースオブジェクトタイプのみを抽出します。[サポートされているオブジェクトタイプ](#)を参照してください。

2. コマンドラインから export コマンドを実行して、schemas_metadata.dump ファイルを生成します。たとえば、<Schema_1>、<Schema_2>、<Schema_3> などのメタデータ情報を抽出するには、次の

コマンドを実行します。

```
expdp <Username>@<ConnectIdentifier> DIRECTORY=DMPDIR SCHEMAS=<Schema_1>,<Schema_2>,  
→<Schema_3> DUMPFILE=schemas_metadata.dump parfile=export.par
```

3. ダンプファイルから SQL ファイルを生成するには、import コマンドを実行します。

たとえば、schemas_metadata.dump ファイルから YourSchemas.SQL ファイルを生成するには、次のように入力します。

```
impdp <Username>@<ConnectIdentifier> DIRECTORY=DMPDIR TRANSFORM=OID:n,SEGMENT_  
→ATTRIBUTES:n EXCLUDE=STATISTICS SQLFILE=YourSchemas.sql DUMPFILE=schemas_metadata.  
→dump
```

移行ポータルで想定されるファイルエンコード形式の詳細については、[file encoding](#) を参照してください。

データベース内のすべてのスキーマを抽出します

注釈

除外スキーマリスト [サポートされていないスキーマ](#) を参照してください。コマンドを実行しているユーザーアカウントがスキーマの除外リストに含まれている場合、*impdp* コマンドは失敗します。

1. expdp コマンドを実行する前に、サーバー上に .par 拡張子たとえば export.par を持つファイルを作成します。属性と値をファイルに追加します。

```
CONTENT=metadata_only  
FULL=Y  
INCLUDE=SYNONYM  
INCLUDE=SEQUENCE  
INCLUDE=TYPE  
INCLUDE=TABLE  
INCLUDE=CONSTRAINT  
INCLUDE=PROCEDURE  
INCLUDE=FUNCTION  
INCLUDE=PACKAGE  
INCLUDE=INDEX  
INCLUDE=MATERIALIZED_VIEW  
INCLUDE=VIEW  
INCLUDE=TRIGGER  
INCLUDE=DB_LINK  
INCLUDE=USER  
INCLUDE=PROFILE  
INCLUDE=GRANT  
INCLUDE=ROLE  
INCLUDE=ROLE_GRANT  
INCLUDE=SYSTEM_GRANT
```

このコマンドの属性と値は、次のオプションを指定します。- `CONTENT=metadata_only` は、スキーマのメタデータのみを抽出します。- `INCLUDE=` は、抽出に含めるデータベースオブジェクトタイプを指定します。移行ポータルでサポートされているデータベースオブジェクトタイプのみを抽出します。[サポートされているオブジェクトタイプ](#)を参照してください。- `FULL=Y` は、データベース内のすべてのスキーマを抽出します。

2. コマンドラインから `export` コマンドを実行して、`schemas_metadata.dump` ファイルを生成します。

```
expdp <Username>@<ConnectIdentifier> DIRECTORY=DMPDIR DUMPFILE=schemas_metadata.dump
→parfile=export.par
```

3. `impdp` コマンドを実行する前に、サーバー上に `.par` 拡張子たとえば `import.par` を持つパラメーターファイルを作成します。属性と値をファイルに追加します。

`EXCLUDE=SCHEMA:`

```
→"IN(select username as schema_name from sys.dba_users where LOWER(username) LIKE pg_% OR username IN
(ANONYMOUS,APEX_PUBLIC_USER,APEX_030200,APEX_040000,APEX_040200,APPQOSSYS,AUDSYS,
→CTXSYS,DMSYS,DBSNMP,
DBSFUSER,DEMO,DIP,DMSYS,DVF,DVSY,EXFSYS,FLWS_FILES,FLWS_020100,FRANCK,GGSYS,
→GSMADMIN_INTERNAL,
GSMCATUSER,GSMROOTUSER,GSMUSER,LBACSYS,MDDATA,MDSYS,MGMT_VIEW,OJMSYS,OLAPSYS,
→ORDPLUGINS,ORDSYS,
ORDDATA,OUTLN,ORACLE_OCM,OWBSYS,OWBYSS_AUDIT,PDBADMIN,RMAN,REMOTE_SCHEDULER_AGENT,SI_
→INFORMTN_SCHEMA,
SPATIAL_CSW_ADMIN_USR,SPATIAL_WFS_ADMIN_USR,SQLTXADMIN,SQLTXPLAIN,SYSSUMF,SY,
→SYSBACKUP,SYSDG,SYSKM,
SYSRAC,SYSTEM,SYSMAN,TSMSYS,WKPROXY,WKSYS,WK_TEST,WMSYS,XDB,XS$NULL))"
EXCLUDE=STATISTICS
```

このコマンドの属性と値は、次のオプションを指定します。- `EXCLUDE` は、除外するスキーマを指定します。[サポートされていないスキーマ](#)のリストを追加して、データポンプユーティリティがシステムスキーマ `pg_` で始まるスキーマを抽出しないことを確認します。- `EXCLUDE=STATISTICS` は、SQL 出力ファイルに統計関連の情報を含めないように指定します。これらを SQL ファイルに含めないことにより、移行ポータルの DDL を評価する機能に影響を与えることなく、ファイルのサイズが大幅に削減されます。

4. ダンプファイルから SQL ファイルを生成するには、`import` コマンドを実行します。

たとえば、`schemas_metadata.dump` ファイルから `YourSchemas.SQL` ファイルを生成するには、次のように入力します。

```
impdp <Username>@<ConnectIdentifier> DIRECTORY=DMPDIR TRANSFORM=OID:n,SEGMENT_
→ATTRIBUTES:n SQLFILE=YourSchemas.sql DUMPFILE=schemas_metadata.dump parfile=import.
→par
```

移行ポータルで想定されるファイルエンコード形式の詳細については、[file encoding](#) を参照してください。

4.2 Performing a schema migration

スキーマのエラーを解決した後、スキーマを EDB Postgres Advanced Server データベースに移行できます。

移行ポータルは、次のターゲットデータベースオプションのいずれかを使用してスキーマを移行するためのガイド付き手順を提供します。

- オフライン移行- データ定義言語 DDL SQL ファイルをダウンロードして、psql クライアントや pgAdmin などのクライアントアプリケーションを使用してターゲット EDB Postgres Advanced Server データベースにスキーマを手動で作成します。このオプションは、オンプレミスとクラウドベースの移行の両方に適しています。
- オンライン移行- ターゲット EDB Postgres Advanced Server データベースに直接接続し、その中にスキーマを自動的に作成します。通常、このオプションは、クラウド BigAnimal クラスターまたは自己管理インスタンスのいずれかで実行されているデータベースへの移行に最適です。

スキーマを移行する前に、EDB は *Known issues, limitations, and notes* ターゲットデータベースの構成要件を特定することをお勧めします。

4.2.1 オフライン移行

クライアントを使用して DDL SQL ファイルを実行することにより、Windows または Linux プラットフォームで EDB Postgres Advanced Server にスキーマを手動で移行します。

1. **Projects** ビューで、**Migrate to...** を選択します。
2. **Offline Migration** と **Next** を選択します。
3. EDB Postgres Advanced Server に移行する 1 つ以上のスキーマを選択します。スキーマを展開し、各スキーマの下で特定のオブジェクトタイプを選択することもできます。

注釈

スキーマの互換性が 100%でない場合、バナーが表示されます。必要に応じて、お問い合わせフォームに記入してください。

1. **Download SQL file** を選択して、評価されたスキーマをダウンロードします。zip パッケージは、選択したスキーマとオブジェクトの DDL を含む SQL ファイルで構成されています。DDL は、プロジェクト、スキーマ、またはオブジェクトごとに SQL ファイルにグループ化されています。

File name	Description	Example
_<ProjectName>_ALL_OBJECTS.sql	選択したすべてのスキーマとオブジェクトの DDL で構成される単一の統合ファイル。	AllTestMigration_ALL_OBJECTS.sql consists of DDLs for all objects of schemas OE, SH, IX, and PM, and only selected objects of schema HR.

次のページに続く

表 2 – 前のページからの続き

File name	Description	Example
_ <SchemaName>_.sql	1 つ以上のファイル。各ファイルは、特定のスキーマで選択したオブジェクトの DDL で構成されます。	HR.sql consists of DDLs for all the selected objects of the HR schema.
_ <SchemaName>_<ObjectType>_.sql	1 つ以上のファイル。それぞれが単一の選択したオブジェクトの DDL で構成されます。	HR_TABLE.sql consists of DDLs only for TABLE object of the HR schema.

1. zip ファイルのダウンロードが完了したら、**Done** を選択して移行ポータルメインページに戻ります。
2. スキーマをターゲットデータベースにインポートします。
3. edb psql コマンドラインユーティリティを使用して、変換されたスキーマの移行元の EDB Postgres Advanced Server データベースに接続します。
4. 新しいデータベースを作成します。

```
CREATE DATABASE <database_name>;
```

1. 次の psql コマンドを使用して、新しいデータベースに接続します。

```
\connect <database_name>
```

1. 次の psql コマンドを使用して、必要な SQL ファイルを新しいデータベースにインポートします。

```
\i <path_to_exported_schema_file>.sql
```

注釈

インポートに `pgAdmin` クライアントを使用することもできます。スキーマをターゲットデータベースにインポートします。

4.2.2 オンライン移行

ターゲット EDB Postgres Advanced Server データベースに直接接続して、スキーマを作成します。

1. **Projects** ビューで、**Migrate to...** を選択します。
2. **Online Migration** と **Next** を選択します。
3. EDB Postgres Advanced Server に移行する 1 つ以上のスキーマを選択します。スキーマを展開し、各スキーマの下で特定のオブジェクトタイプを選択することもできます。
1. **Connection** ページで、必要な接続の詳細を入力します。

HM 管理の宛先 宛先が EDB Postgres Advanced Server を実行している HM 管理の分散高可用性または高度な高可用性データベースクラスターの場合、**ターゲットデータベース** と **メンテナンスデータベース** フィールドの両方デフォルトの bdrdb データベースを指す必要があります。

Generate and download DDL SQL scripts for offline migration of schemas

1 Select Schemas

2 Download Schema

3 Finish

Project Name	Interface	Source DB	Target DB	Compatibility
HR_SCHEMA	JDBC	Oracle 12c	EDB Postgres Advanced Server 16	100%

Select one or more schemas

Note: Click the arrows next to the schema name to see and select specific object types to be migrated. Also note that the numbers in parenthesis next to the object type name indicate the number of compatible objects, total number of objects, and % of compatible objects currently available in the schema for the object type.

☒ All

> ☒ \$GLOBAL_OBJECTS

< ☒ HRPLUS

☒ CONSTRAINT (10/10 - 100%)

☒ FUNCTION (4/4 - 100%)

☒ INDEX (11/11 - 100%)

☒ PACKAGE (1/1 - 100%)

☒ PACKAGE BODY (1/1 - 100%)

☒ PROCEDURE (9/9 - 100%)

☒ SEQUENCE (3/3 - 100%)

☒ TABLE (11/11 - 100%)

☒ TRIGGER (3/3 - 100%)

☒ TYPE (1/1 - 100%)

☒ VIEW (9/9 - 100%)

Cancel

< Previous

Next >

図 1: Schema and objects are selected

Connect to database and perform online migration of schemas

1 Select Schemas

2 Connect

3 Deploy

4 Finish

Project Name	Interface	Source DB	Target DB	Compatibility
HR_SCHEMA	JDBC	Oracle 12c	EDB Postgres Advanced Server 16	100%

Select one or more schemas

Note: Click the arrows next to the schema name to see and select specific object types to be migrated. Also note that the numbers in parenthesis next to the object type name indicate the number of compatible objects, total number of objects, and % of compatible objects currently available in the schema for the object type.

☒ All

> ☒ \$GLOBAL_OBJECTS

< ☒ HRPLUS

☒ CONSTRAINT (10/10 - 100%)

☒ FUNCTION (4/4 - 100%)

☒ INDEX (11/11 - 100%)

☒ PACKAGE (1/1 - 100%)

☒ PACKAGE BODY (1/1 - 100%)

☒ PROCEDURE (9/9 - 100%)

☒ SEQUENCE (3/3 - 100%)

☒ TABLE (11/11 - 100%)

☒ TRIGGER (3/3 - 100%)

☒ TYPE (1/1 - 100%)

☒ VIEW (9/9 - 100%)

Cancel

< Previous

Next >

図 2: Schema and objects are selected

Connect to database and perform online migration of schemas

1 Select Schemas

2 Connect

3 Deploy

4 Finish

Project Name	Interface	Source DB	Target DB	Compatibility
HR_SCHEMA	JDBC	Oracle 12c	EDB Postgres Advanced Server 16	100%

Target Database

TestMigration

Host Name/Address

Host Name/Address

Port

5432

Note: Ensure that your database server is accessible from Migration Portal.

Maintenance Database

edb_admin

Username

edb_admin

Password

Password

Test Connection

Cancel

Previous

Next

図 3: Connecting to the cloud cluster

オンプレミスデータベースサーバーの宛先 宛先が EDB Postgres Advanced Server を実行している PGD クラスターの場合、**Target Database** と **Maintenance Database** の両方のフィールドが、プライマリ PGD データベースを指している必要があります。クラスター全体で複製されます。

注釈

****Edit Connection****を選択して、接続の詳細を変更し、接続の詳細を再テストできます。

1. テスト接続が成功した場合は、**Next** を選択してターゲットデータベースに接続し、ターゲットサーバーに選択したスキーマを作成します。
2. 移行が完了すると、ページに、移行が正常に完了したかエラーが発生したかを示すメッセージが表示されます。**Download Overview** ボタンを選択して、移行試行に対して生成されたログをダウンロードして確認します。
3. **Done** を選択して、移行ポータルメインページに戻ります。

4.3 Performing a data migration

スキーマの移行を実行した後、データを移行します。

1. Migration Toolkit を使用してデータを移行します。Migration Toolkit の使用の詳細については、[Migration Toolkit](#) を参照してください。
2. Migration Toolkittoolkit.properties ファイルを構成し、ソースデータベースとターゲットデータベースの接続情報がプロパティファイルで利用できることを確認します。

```
SRC_DB_URL = jdbc: oracle:thin:@localhost:1521:ORCL
SRC_DB_USER = user_name
SRC_DB_PASSWORD = password
TARGET_DB_URL= jdbc:edb://localhost:5444/migration
TARGET_DB_USER = enterprisedb
TARGET_DB-PASSWORD = password
```

詳細については、[Building the toolkit.properties File](#) を参照してください。

3. **-dataOnly** モードで Migration Toolkit を呼び出します。**-truncLoad** キーワードを含めて、テーブル間の外部キーの依存関係を解決します。

たとえば、次のコマンドは、データを **hr** という名前のスキーマに移行します。

```
runMTK.sh -dataOnly -targetSchema hr -truncLoad HR
```

このコマンドは、指定された **source_schema** を **target_schema** に移行します。データは、**enterprisedb** という名前のデータベーススーパーユーザーと **password** のパスワードを使用して、ローカルにインストールされた EDB Postgres Advanced Server インスタンスにロードされます。

注釈

テーブルは、データのロードを試行する前に切り捨てられます。

4.3.1 高度なデータの移行

並列データのロードが必要な大規模なデータベースの場合、次のいずれかの方法を使用できます。

- EDB Postgres Advanced Server データベースリンク機能 Oracle データベースとの互換性のため
- Dblink またはデータベースリンクスタイルの移行データに CLOB データが含まれる場合

詳細については、[Migration Toolkit](#) を参照してください。

4.4 Performing a schema assessment

EDB Postgres Advanced Server との互換性について Oracle データベーススキーマを評価できます。評価に関するレポートを生成することもできます。

注釈

サポートされていないスキーマオブジェクトの概要については、[Migration Portal 4.0.0 release notes](#) および [Known issues, limitations, and notes](#) を確認します。

4.4.1 スキーマ評価を実行する

1. [Migration Portal](#) に移動します。
2. EDB 資格情報を入力します。
3. **New** を選択してプロジェクトを作成します。
4. ダイアログボックスで、**プロジェクト名**フィールドにプロジェクト名を入力します。
5. プロジェクトの詳細を指定します。
 - アプリケーションインターフェイスを選択します。
 - ソースのソース DB とバージョンを選択します。
 - ターゲットのターゲット DB とバージョンを選択します。
6. スキーマメタデータで生成された .SQL ファイルをアップロードするには、**Choose file** を選択します。

注釈

.SQL ファイルは変更しないでください。

1. プロジェクトの作成時に、インデックス_IDX とコンストレイン_CON へのサフィックスの追加を指定するには、**Add suffix to make object names Unique** チェックボックスをオンにします。EDB Postgres Advanced Server はテーブルとインデックスの同じ名前をサポートしていませんが、Oracle はサポートしているため、この選択によりより良い評価結果が保証されます。
2. プロジェクトの作成時に Oracle のデフォルトの大文字と小文字を使用することを指定するには、**Use Oracle の default case for schema 移行**チェックボックスを選択します。Oracle データベースから EDB Postgres Advanced Server データベースにスキーマを移行するときに、Oracle のデフォルトの大

文字小文字機能を使用して、すべてのデータベースオブジェクトの Oracle のデフォルトの大文字の命名規則を保持します。この機能の詳細については、[Known issues, limitations, and notes](#) を参照してください。

3. DEFAULT プロファイルの評価を指定するには、プロジェクトの作成時に、**Assess DEFAULT Profile 利用可能な場合** チェックボックスを選択します。EDB DDL エクストラクタは、Oracle データベースでオーバーライドされる DEFAULT プロファイルの ALTER PROFILE ステートメントを抽出します。このオプションにより、移行ポータルは DEFAULT プロファイルを評価し、スキーマの移行時にターゲット EDB Postgres Advanced Server データベースにこれらのステートメントを適用しようとします。DDL をターゲットデータベースに移行する前に、必要に応じて、ALTER PROFILE DEFAULT ステートメントの設定を検証および更新することをお勧めします。この機能の詳細については、[Known issues, limitations, and notes](#) を参照してください。

4. **Create & assess** を選択して、新しいプロジェクトを作成し、スキーマを評価します。

分析ツールは、すべてのコンストラクトを確認し、修復アクションを実行して、EDB Postgres Advanced Server との互換性を向上させます。手動介入を必要とする残りのエラーにフラグを立てます。評価プロセスの一部として、移行ポータルは **SPL チェック** ベースの分析を自動的に実行します。

1. 100 の成功率を示さない DDL オブジェクトテーブルなどを確認します。
2. EDB Postgres Advanced Server と互換性のないオブジェクトを選択し、詳細を表示します。中央のパネルの下部に、修復アクションの詳細を表示できます。

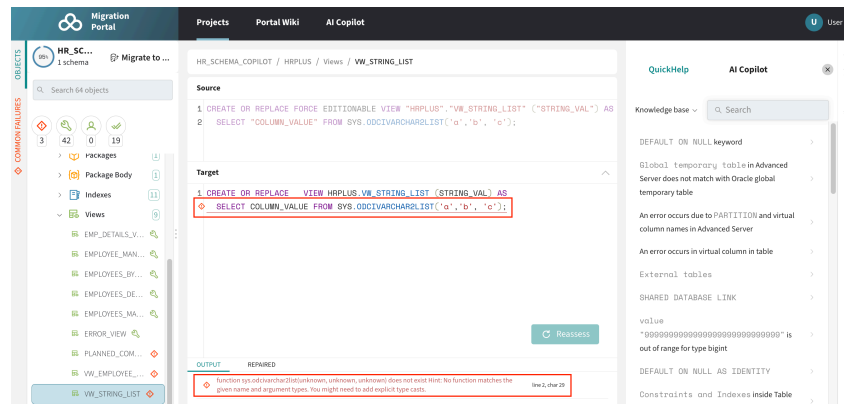


図 4: Incompatible object and hint

1. EDB Postgres Advanced Server とすぐに互換性のないオブジェクトの考えられる回避策を見つけるには、**クイックヘルプ** ナレッジベース情報を参照するか、右パネルの **AI Copilot** と対話します。

クイックヘルプ ナレッジベースを使用する場合、EDB Postgres Advanced Server と互換性のないオブジェクトのエラーメッセージを入力し、**検索**を選択します。

AI Copilot を使用する場合、チャットウィンドウを使用して AI 対応チャット機能と対話します。互換性に関する質問、エラーメッセージの入力、DDL クエリの Postgres 同等の要求構文などを行うことができます。

注釈

互換性のないオブジェクトの情報または回避策がナレッジベースまたは AI コパイロットで利用できない場合、サポートチームに連絡してヘルプを求めてください。

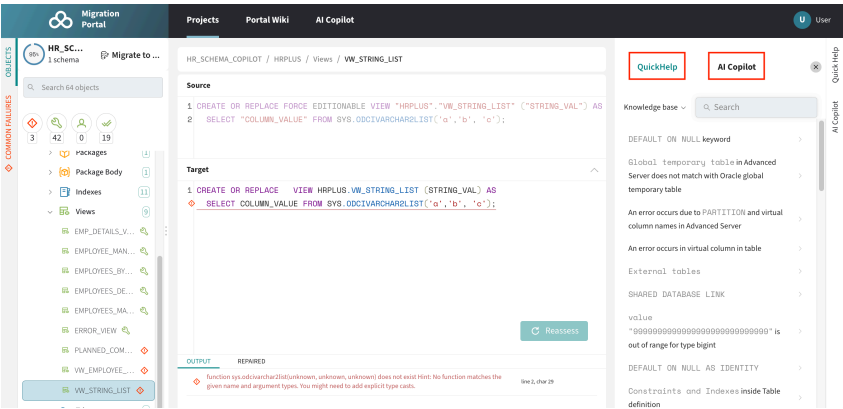


図 5: Incompatibility resolution with Quick Help and AI Copilot

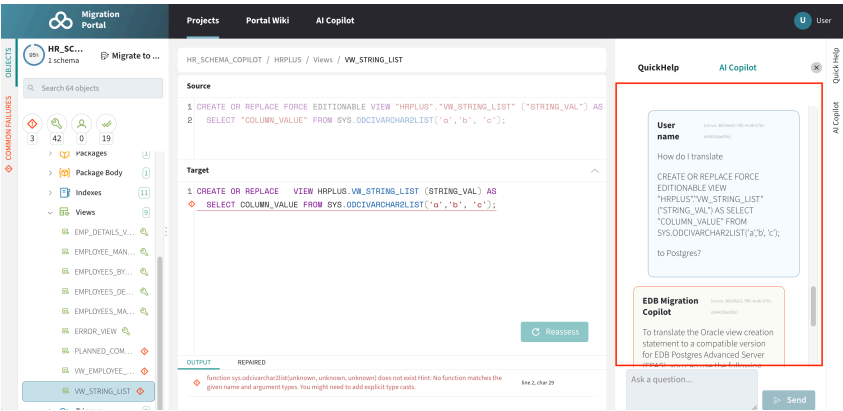


図 6: AI Copilot suggestions

- ナレッジベースと AI コパイロットから取得した情報を使用して、互換性のないすべてのオブジェクトを互換性のあるものにします。そのオブジェクトの**ターゲット** パネルで変更を手動で行い、**Reassess** を選択します。

重要

提案されたソリューションをすべてテストして、変換されたスキーマが予想通りに動作することを確認します。

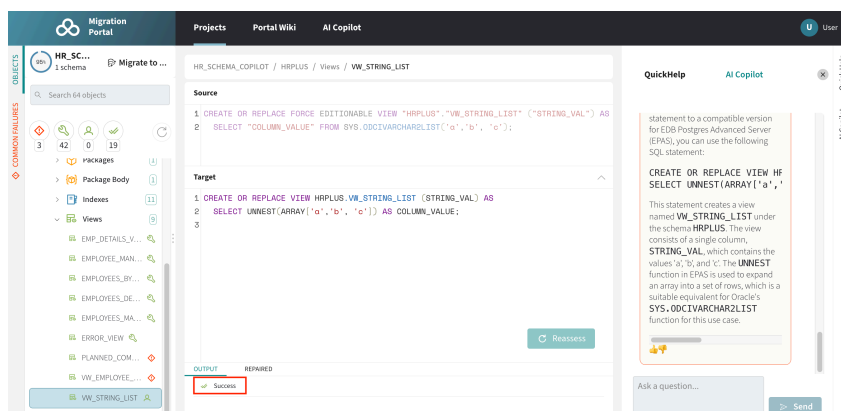


図 7: Incompatibility resolved successfully

注釈

再評価中に失敗したオブジェクトが合格した場合、そのオブジェクトの依存オブジェクトも再評価されます。

DDL での作業が終了したら、変更した EDB 互換 DDL を .sql ファイルとしてダウンロードするか、スキーマを移行できます。 [Performing a schema migration](#) を参照してください。

4.4.2 評価レポートを生成する

移行ポータルレポートは、プロジェクトについて評価されたスキーマの高レベルの評価概要を提供します。さらに、レポートは、障害が発生したオブジェクトと障害の原因に関する詳細を提供します。

- レポートを選択します。
- レポートに含めるスキーマを選択します。
- 画面上のレポートを生成するには、**Generate** を選択します。

Generate PDF または **Generate JSON** オプションを選択して、レポートを保存できます。JSON 形式を使用して、外部システムに簡単に統合したり、さらに自動処理したりできます。

選択したスキーマの DDL に適用された個別の修復ハンドラーの数を表示することもできます。

i 注釈

移行ポータルは、Oracle でラップされたオブジェクトに隠されている機密の PL/SQL ソースコードを評価しません。これらのラップされたオブジェクトは評価されたオブジェクト数に含まれないため、互換性のパーセンテージの真の値は、評価レポートで計算された値と異なる場合があります。

ラップされたオブジェクトの詳細については、[Known issues and notes](#) のノートを参照してください。

4.4.3 評価レポートを評価する

評価レポートは、スキーマおよびプロジェクト全体のオブジェクトと機能の互換性に関するデータと情報を提供します。

- **オブジェクト** 移行ポータルは、処理するすべての CREATE ステートメントのオブジェクトを生成します。さらに、アップロードされた DDL に制約を追加する各 ALTER TABLE ステートメントのオブジェクトを生成します。
- **Feature** オブジェクトの作成に使用される Oracle 句または構成体。フィーチャは 1 つ以上のオブジェクトで複数回使用できます。

評価レポートには、`_In 互換性のない機能_`および`_未解決のオブジェクト_`のテーブルがリストされています。

オブジェクトの互換性

オブジェクトは、ターゲットデータベースで正常に実行された場合、互換性があります。

オブジェクトの互換性の概要

評価レポートには、次のカテゴリに基づいてオブジェクトの互換性の概要が表示されます。

Object level	compatibility	Description
Passed		修復ハンドラーを適用せずにターゲットデータベースで実行します。
System repaired		1 つ以上の修復ハンドラーを適用した後、ターゲットデータベースで正常に実行されます。Repair handlers は、変換ロジックを使用して、DDL を EDB Postgres Advanced Server と互換性のあるものに自動的に作成します。オブジェクトの動作に予期しない影響が発生した場合に、レポートの Repair Handler apply セクションの説明を注意深く確認してください。
User repaired		ユーザーが手動で修復および再評価した後、ターゲットデータベースで正常に実行されます。
Failed objects		ターゲットデータベースで正常に実行されません。

i 注釈

表示されるオブジェクト分析は、ターゲット DDL のオブジェクトの既存の状態にもとづいています。オブジェクトが更新および再評価されると変更される可能性があります。たとえば、修正されたオブジェクトは、再評価後に `_Unresolved Objects_` テーブルから削除されます。

追加のオブジェクト互換性情報

概要の後に、オブジェクトの互換性に関連する追加セクションが表示されます。

Report section	Description	Usage
All failures	オブジェクトが変換に失敗したすべての理由を要約し、スキーマごとに影響を受ける失敗数と特定のオブジェクトタイプをリストします。プロジェクト全体にわたって「リレーションが存在しない」または「タイプが存在しない」などの一般的な問題の統合ビューが表示されます。	You can identify the most common compatibility issues, allowing you to prioritize the development of reusable fix strategies for widespread failures.
Object dependencies	スキーマ内の他のオブジェクトとの関連に焦点を当て、オブジェクト、オブジェクトタイプ、およびエラーを階層形式でリストします。多くの場合、依存関係チェーンを示し、1つのオブジェクトテーブルなどの障害が依存オブジェクトプロセスまたはビューなどにエラーをカスケードします。	You can establish the correct conversion order and prioritize fixing foundational objects first, as resolving a parent object's issue will likely clear the errors on many dependent children. When incompatible objects require refactoring (using the AI Copilot, or manual rewriting), the dependency list helps determine the exact scope of the rewrite, ensuring all calling procedures, views, or functions are adjusted correctly.
Assessment result for schemas	プロジェクトに含まれる個々のスキーマのオブジェクトの互換性の詳細な内訳を提供します。これは、スキーマごとの合計、合格、システム修復、ユーザー修復、および失敗したオブジェクトの数を要約したものです。	Scope the migration effort for each schema individually, focusing resources on schemas with the highest number of failed or system-repaired objects.

次のページに続く

表 4 – 前のページからの続き

Report section	Description	Usage
Repair handlers applied	システムで使用する自動修復ハンドラーのコード ERH-2013、短い説明、および長い説明をリストします。これは、各一意のハンドラーが適用された回数、ターゲットデータベースでオブジェクトを正常に実行した回数を示します。	Understand the extent of automated changes applied to your DDL, to help you validate the converted code and identify potential post-migration performance impacts.

機能の互換性

一部の機能は、EDB Postgres Advanced Server でネイティブに使用できない場合があります。評価レポートには、機能の発生と互換性レベルの概要が表示されます。

Feature compatibility level	Workaround status
Compatible features	•
Features with existing available workaround	Exists
Features with reasonable effort new workaround	Possible
Features with no workaround or requiring further investigation	None / Needs Investigation

注釈

回避策がない機能の場合、アプリケーションレベルの変更を介して機能の非互換性に対処できる場合があります。

制限事項

評価レポートは、ユーザーがターゲット DDL を手動で変更し、**Reassess** を選択した後に更新されます。ただし、レポートの `_In stable attempts_` テーブルは更新されず、ソース DDL 内の互換性のない機能のリストを継続します。スキーマ移行の複雑性評価スコアは、互換性のない機能を備えた互換性のないオブジェクトが解決されると更新されます。また、オブジェクトが修正され再評価された後、これらのオブジェクトで元々見つかった互換性のない機能のリストとともに、`_Unresolved Objects_` テーブルから削除されます。

Schema assessment checks

スキーマ評価中に、次のチェックが自動的に実行されます。

SPL チェック

SPL Check は、Oracle プロシージャコード内の潜在的な実行時エラーを特定するコード分析ツールです。これは、EDB Postgres Advanced Server に移行する際の評価プロセスの重要な部分です。

移行ポータルで評価されるすべてのスキーマは、SPL Check によって分析されます。このファンクションは、[EDB SPL Check](#) 拡張機能に基づいていますが、FUNCTION、PROCEDURE、PACKAGE BODY、TYPE BODY、および TRIGGER オブジェクトタイプ内の実行時エラーを検出します。

スキーマが評価された後、チェックに合格しないオブジェクトは、他の互換性問題とともに、移行ポータルプロジェクトで**失敗**としてマークされます。これにより、オブジェクトを宛先データベースにインポートする前にこれらの問題を解決でき、後で実行時エラーを防ぐことができます。

SPL Check エラーは、[spl_check] という接頭辞が付いているため、他の非互換性と区別できます。影響を受けるオブジェクトを変更して問題を修正した後、**Reassess** を選択してチェックを再実行し、編集によって問題が解決されたことを確認します。

第5章 Known issues, limitations, and notes

これらは、次の既知の問題、制限、および注意事項です。

- *Migration Portal 4.0.0 release notes*
- *Extracting schemas using the EDB DDL Extractor*
- *Extracting schemas using Oracle Data Pump utilities*

5.1 Migration Portal

5.1.1 ラップされたオブジェクト

移行ポータルは、ラップされたオブジェクトを評価できません。抽出された DDL にそれらを含める場合、それらは移行ポータルにロードされず、評価されるオブジェクトの数に含まれません。ラップされたオブジェクトを評価し、EDB Postgres Advanced Server に移行する場合は、移行ポータルにアップロードする DDL ファイルにオブジェクトのラップされていないバージョンを含めます。これを行うことをお勧めします。スキーマ抽出を実行する前に、Oracle データベース内のラップされたバージョンをクリアテキストバージョンに置き換えることです。スキーマ抽出を実行した後、オブジェクトをラップされたバージョンに置き換えることができます。

5.1.2 削除されたオブジェクト

一部のサポートされていない Oracle オブジェクトは、移行ポータルがソース DDL ファイルを評価するときに削除されます。これらの削除されたオブジェクトは変換手順から除外され、移行ポータル評価で削除されたとしてフラグが立てられません。

次のオブジェクトは、スキーマ評価中に削除されます。

- **MATERIALIZED VIEWS** に関連するオブジェクトたとえば、**MVIEW** をサポートするように作成されたバックエンド **TABLE** または **INDEX** ステートメント
- **Queues** 関連オブジェクト
- **Nested Tables** 関連オブジェクト
- **XMLType Tables** 関連オブジェクト
- **SYSTEM Schemas** に依存する型
- **PRIMARY KEY** および **UNIQUE** コンストレインに関連するインデックス
- サポートされていないシステム **GRANT** 特権

EDB は、CREATE DATABASE LINK、CREATE PUBLIC DATABASE LINK、DROP PUBLIC DATABASE LINK、および EXEMPT ACCESS POLICY システム権限の付与のみをサポートしています。他の GRANT ステートメントはサポートされておらず、DDL ファイルから削除されます。

5.1.3 ファイルエンコード

移行ポータルは、.SQL 出力ファイルを UTF-8 エンコード形式にすることをお勧めします。非 UTF-8 エンコードを使用して .SQL ファイルをアップロードする場合、UTF-8 と互換性のないすべての文字は、出力 DDL で置換文字 `☒` に変換されます。

💡 Tip

Linux では iconv ユーティリティ、または Windows では LibIconv ユーティリティを使用して、抽出したファイルを UTF-8 形式に手動で変換できます。たとえば、データベースキャラクタセットが Latin-1 ISO-8859-1 である場合、次のように、抽出したファイルを UTF-8 形式に変換できます。

```
iconv -f iso-8859-1 -t UTF-8 sample.sql > sample_utf8.sql
```

5.1.4 ホワイトラベルエラー

移行ポータルの新しいバージョンがリリースされると、移行ポータルを開くことができない問題が発生する場合があります。White label Error Page またはその他のエラーのようなエラーにより、移行ポータルを使用できません。ブラウザに保存されているキャッシュデータがこのエラーの原因となります。これが発生した場合は、ブラウザ履歴からキャッシュデータをクリアするか、シークレット/プライベートウィンドウを使用して移行ポータルにアクセスします。

5.1.5 ALTER ステートメント

ALTER PROFILE DEFAULT、ALTER TABLE、および ALTER TRIGGER を除き、移行ポータルは DDL 内の他の ALTER ステートメントを処理しません。

5.1.6 ユーザー、ロール、プロファイル、および付与

5.1.7 付与

移行ポータルは、抽出およびアップロードされたスキーマの Oracle ユーザーの評価と移行をサポートするようになりました。プロジェクトに関連付けられているスキーマのすべてのユーザー、ロール、およびプロファイルは、移行ポータルビューの GLOBAL_OBJECTS 擬似スキーマの下にグループ化されます。現在、システム特権、オブジェクト特権、およびロールの付与は、ユーザーオブジェクトの DDL に含まれています。同様に、ロールへのシステム特権およびオブジェクト特権の付与は、ロールオブジェクトの DDL に含まれます。

付与は現在ユーザーとロールオブジェクトにアタッチされているため、ユーザーに権限が付与されている 1 つ以上のオブジェクトに障害が発生すると、ユーザーオブジェクト自身自身は評価に不合格になります。この問題を解決するには、個々のステートメントの SQL 単一行コメント -- または連続する grant ステートメントの SQL 複数行コメント / ** / を使用して、障害が発生したオブジェクトの grant ステートメントをコメントアウトし、オブジェクトの再評価ができます。実行されます。失敗した grant ステートメントに対

応する障害のあるオブジェクトが修復された後、以前に失敗した `grant` ステートメントのコメントを外し、ユーザーオブジェクトを再評価できます。

RESOURCE ロールの付与は、被付与者と同じ名前を共有するスキーマの `CREATE` および `USAGE` 特権を付与することと同じです。RESOURCE ロールを付与する場合は、最初にユーザーまたはロールと同じ名前でスキーマを作成する必要があります。それ以外の場合、「スキーマは存在しません」エラーが発生します。

5.1.8 特権

スキーマをターゲット EDB Postgres Advanced Server データベースに移行するには、移行の実行に使用されるターゲットデータベースユーザーに次の特権が必要です。

- SUPERUSER プロファイルを移行する
- ユーザーとロールを移行するための SUPERUSER または CREATE ROLE

5.1.9 パスワード

ユーザーを移行する際、移行ポータルはユーザーのパスワードを移行しません。ユーザーは、パスワードなしでターゲット EDB Postgres Advanced Server データベースに作成されます。ユーザーをパスワードで構成することが望ましいまたは必要な場合、これはユーザーが移行された後、ターゲットデータベースで手動で行うことができます。

5.1.10 プロファイル

移行ポータルは、スキーマユーザーに割り当てられた非 `DEFAULT` プロファイルを移行します。Oracle プロファイルには、パスワードとリソース制限の両方が含まれます。EDB Postgres Advanced Server は、Oracle パスワード関連の制限のみをサポートしています。リソース制限はプロファイルで抽出されますが、移行ポータルはパスワード制限のみをアップロード、評価、および移行します。また、`DEFAULT` プロファイルは、Oracle データベースで新しい制限値でオーバーライドされる場合があります。EDB DDL エクストラクタは、Oracle データベースでオーバーライドされる `DEFAULT` プロファイルの `ALTER PROFILE` ステートメントを抽出し、移行ポータルはこれらのステートメントをターゲット EDB Postgres Advanced Server データベースに適用しようとします。スキーマを既存のユーザーとデータベースを含む EDB Postgres Advanced Server インスタンスに移行する場合、DDL をターゲットデータベースに移行する前に、必要に応じて `ALTER PROFILE DEFAULT` ステートメントの設定を検証および更新することをお勧めします。

注釈

バージョン 4.3.0 以降、移行ポータルは、ユーザー、ロール、プロファイル、および付与の評価をサポートしています。ただし、移行ポータルの 4.3.0 より前のバージョンを使用して作成されたプロジェクトの場合、この機能はサポートされていません。ユーザー、ロール、プロファイル、および付与を評価するには、最新バージョンの EDB DDL エクストラクタを使用し、新しいプロジェクトを作成します。

5.1.11 Oracle のデフォルトの場合を使用する

Oracle のデフォルトの大文字小文字機能を使用するオプションは、Oracle データベースから EDB Postgres Advanced Server データベースにスキーマを移行するときに、すべてのデータベースオブジェクトの Oracle のデフォルトの大文字命名規則を保持します。

EDB DDL エクストラクタおよび Oracle Data Pump ユーティリティは、PL/SQL の本文内で参照されているオブジェクト名を除き、Oracle のデフォルトの大文字小文字つまりオブジェクト名の大文字または大文字と小文字が区別されるオブジェクト名のそれぞれの大文字と小文字ですべての DDL を抽出します DDL の作成時にユーザーが明示的に二重引用符で囲まれない限り、オブジェクト。

したがって、移行ポータルは、この機能のソーススキーマファイルからすべての二重引用符を保持し、さらに、移行ポータルは以下に二重引用符を適用します。

- ビューおよびマテリアライズドビューの SELECT ステートメント内の列名。
- チェック制約内のカラム名。
- トリガーヘッダー内のテーブル名と列名。

Oracle 大文字小文字を使用するオプションが選択されていない場合、移行ポータルのデフォルト動作は、すべてのオブジェクトが小文字 EDB Postgres Advanced Server のデフォルトの大文字または引用符で囲まれた大文字と小文字の混合を使用してオブジェクト名が指定されている場合は大文字と小文字が混合で作成されることですソース DDL で。

useOraCase 機能を使用して、以下のソース DDL とターゲット DDL を見つけます。

```
-- Source DDL
CREATE TABLE "RHS_SANITY"."DEPT"
(
  "DEPTNO" NUMBER(2,0),
  "DNAME" VARCHAR2(14),
  "LOC" VARCHAR2(14),
  CONSTRAINT "DEPTDEPTNO_PK" PRIMARY KEY ("DEPTNO")
USING INDEX ENABLE,
  CONSTRAINT "DEPTDNAME_UK" UNIQUE ("DNAME")
USING INDEX ENABLE
);

-- Target DDL
CREATE TABLE "RHS_SANITY"."DEPT"
(
  "DEPTNO" NUMBER(2,0),
  "DNAME" VARCHAR2(14),
  "LOC" VARCHAR2(14),
  CONSTRAINT "DEPTDEPTNO_PK" PRIMARY KEY ("DEPTNO")
USING INDEX ENABLE,
  CONSTRAINT "DEPTDNAME_UK" UNIQUE ("DNAME")
USING INDEX ENABLE
);
```

Oracle のデフォルトのケースを使用している間、他のプロジェクトと比べて互換性率が低下する場合があります。次に、この互換性率の低下の原因と考えられる理由を示します。

- 最も一般的な理由は、オブジェクト/リレーションが存在しないことです。このシナリオは、テーブル/オブジェクトが Oracle の大文字と小文字を保持するために二重引用符で囲まれている大文字であるのに対し、障害が発生したオブジェクトでは二重引用符なしで参照されるという事実によって発生します。例

```
CREATE TABLE "HR"."EMP"
(
  "EMPNO" NUMBER(4,0),
  "ENAME" VARCHAR2(10),
  "JOB" CLOB,
  "MGR" NUMBER(4,0),
  "HIREDATE" DATE,
  "SAL" NUMBER(7,2),
  "COMM" NUMBER(7,2),
  "DEPTNO" NUMBER(2,0),
  CONSTRAINT "PK_EMP" PRIMARY KEY ("EMPNO")
USING INDEX ENABLE
);

CREATE OR REPLACE PROCEDURE "HR"."PROC"
IS
  CURSOR emp_cur IS SELECT hiredate, CAST(hiredate AS timestamp(6)) AS timestamped_
  FROM emp;
  REC emp_cur%rowtype;
BEGIN
  OPEN emp_cur;
  LOOP
    FETCH emp_cur INTO rec;
    EXIT WHEN emp_cur%NOTFOUND ;
    DBMS_OUTPUT.PUT_LINE(rec.hiredate || ||rec.timestamped);
  END LOOP;
  CLOSE EMP_CUR;
END ;
```

エラーは次のように表示されます。

```
relation "emp" does not exist
```

上記の例では、テーブル名とその列名は CREATE TABLE ステートメントで引用符で囲まれた大文字の名前で指定されています。ただし、CREATE OR REPLACE PROCEDURE ステートメントでは、テーブルとその列は、引用符で囲まれていない小文字の識別子を使用して参照されます。

- %type を使用して定義された変数は、前述の table.column-name が employees.employee_id%type のように小文字であるのに対し、実際のテーブルのテーブル名と列名は二重引用符で囲まれた大文字であるため、コードオブジェクトで失敗します。例

```
CREATE TABLE "HR"."EMPLOYEES"
(
  "EMPLOYEE_ID" NUMBER(6,0),
```

(次のページに続く)

```

"FIRST_NAME" VARCHAR2(20),
"LAST_NAME" VARCHAR2(25) CONSTRAINT "EMP_LAST_NAME_NN" NOT NULL ENABLE,
"EMAIL" VARCHAR2(25) CONSTRAINT "EMP_EMAIL_NN" NOT NULL ENABLE,
"PHONE_NUMBER" VARCHAR2(20),
"HIRE_DATE" DATE CONSTRAINT "EMP_HIRE_DATE_NN" NOT NULL ENABLE,
"JOB_ID" VARCHAR2(10) CONSTRAINT "EMP_JOB_NN" NOT NULL ENABLE,
"SALARY" NUMBER(8,2),
"COMMISSION_PCT" NUMBER(2,2),
"MANAGER_ID" NUMBER(6,0),
"DEPARTMENT_ID" NUMBER(4,0),
CONSTRAINT "EMP_SALARY_MIN" CHECK ("SALARY" > 0) ENABLE,
CONSTRAINT "EMP_EMP_ID_PK" PRIMARY KEY ("EMPLOYEE_ID")
USING INDEX ENABLE,
CONSTRAINT "EMP_EMAIL_UK" UNIQUE ("EMAIL")
USING INDEX ENABLE
);
CREATE OR REPLACE PROCEDURE "HR"."PKG" AS
-- declare variables for data fetched from cursor
empid      employees.employee_id%TYPE; -- variable for employee_id
hiredate    employees.hire_date%TYPE;   -- variable for hire_date
firstname   employees.first_name%TYPE;  -- variable for first_name
lastname    employees.last_name%TYPE;   -- variable for last_name
rowcount    NUMBER;
bonusamount NUMBER;
yearsworked NUMBER;
-- declare the cursor with a parameter
CURSOR cursor1 (thismonth NUMBER) IS
    SELECT employee_id, first_name, last_name, hire_date FROM employees
        WHERE EXTRACT(MONTH FROM hire_date) = thismonth;
BEGIN
    null;
END;
```

エラーは次のように表示されます。

```
syntax error at or near "%" (line 3, char 37)
```

- テーブルの列データ型に TYPE 参照がある場合、大文字と小文字の違いが原因で失敗します。例

```

CREATE OR REPLACE TYPE "HR"."USER_TYPE" IS OBJECT(
    "ID" NUMBER,
    "NAME" VARCHAR2(256)
);

CREATE TABLE "HR"."MY_TABLE"
```

(次のページに続く)

(前のページからの続き)

```
(
  "ID" NUMBER,
  "MY_NAME" user_type
);
```

エラーは次のように表示されます。

```
type "user_type" does not exist
LINE 4:      "MY_NAME" user_type
           ^
SQL state: 42704
Character: 58
```

5.1.12 不完全なオブジェクト定義

移行ポータルでソーススキーマを評価しているときに、まれに、ソース DDL に予約語が含まれている場合、変換されたオブジェクト定義が不完全です。

オブジェクト定義の一部が欠落しているか、ステートメントの実際の最後に到達する前に定義が終了する問題が発生した場合は、EDB Postgres Advanced Server によって予約されているキーワードを使用している可能性があります。たとえば、TABLE 定義で列名として CONSTRAINT を使用し、VIEW または MVIEW 定義で参照すると、移行ポータルはその DDL ステートメントを突然終了し、正常に評価または変換せずに次の定義にスキップする場合があります。

この問題をワークアラウンドには、次のいずれかのことができます。

- Oracle データベースの列名を変更し、DDL を再抽出して移行ポータルにアップロードします。
- 抽出した DDL ファイルの列名を変更し、移行ポータルにアップロードします。

5.2 EDB DDL エクストラクタ

5.2.1 一般的な制限事項

- EDB DDL エクストラクタスクリプトは、Flashback を使用して復元されたオブジェクトで、BIN\$b54+4XlEYwPgUAB/AQBWwA==\$0 のような名前が残っているオブジェクトを抽出しません。これらのオブジェクトを抽出する場合は、オブジェクトの名前を変更し、抽出プロセスを再実行する必要があります。
- EDB DDL エクストラクタは、nologging テーブルを通常のテーブルとして抽出します。これらのテーブルが EDB Postgres Advanced Server に移行されると、WAL ログファイルが作成されます。
- EDB DDL エクストラクタは、VALID ステータスでのみオブジェクトを抽出します。移行ポータルで評価する INVALID ステータスを持つオブジェクトについては、最初に VALID に更新します。
- EDB DDL エクストラクタは、Oracle のラップ機能を使用して難読化されたオブジェクトを抽出しません。そのため、これらのオブジェクトは、移行ポータルによって評価される DDL のセットには含まれません。これらのオブジェクトを評価し、EDB Postgres Advanced Server に移行する場合は、オ

プロジェクトのラップされたバージョンを非ラップバージョンに置き換えます。詳細については、[ラップされたオブジェクト](#) を参照してください。

- EDB DDL エクストラクタは、グローバル一時テーブルを作成して、スキーマ名とその依存関係情報を保存します。これらのテーブルは、抽出が成功した最後に削除されます。
- PostgreSQL がサポートしていないため、EDB DDL エクストラクタスクリプトは、名前が PG_ で始まるスキーマを抽出しません。これらのスキーマを抽出する場合は、抽出前にスキーマの名前を変更する必要があります。
- EDB DDL エクストラクタは、プロファイル、ロール、および付与の情報を自動的に抽出します。
- EDB DDL エクストラクタは、現在 Oracle 11g からの ROLES、SYSTEM GRANTS ON ROLES、OBJECT GRANTS ON ROLES、および ROLE GRANTS の抽出をサポートしていません。この動作により、エラーメッセージがこれらのオブジェクトタイプに対応するセクションの抽出されたファイルに書き込まれます。これらのエラーは、移行ポータルによるこれらのファイルの評価において問題を発生させることはありません。
- EDB DDL エクストラクタスクリプトは、抽出されたファイルの依存オブジェクトセクションに object "OBJECT_NAME" of type SYNONYM not found in schema "PUBLIC" エラーをログに記録する場合があります。これは、Oracle データベースがコンテナデータベースである Oracle マルチテナント環境から依存オブジェクトを抽出するオプションをユーザーが選択した場合にのみ発生します。

5.2.2 「スナップショットが古すぎます」エラー

EDB DDL エクストラクタは、データベースサーバーが UNDO テーブルスペースで適切に処理できない大量のトランザクションを生成すると、実行時にエラー「ORA-01555: snapshot too old」を表示します。

データベースサーバーが高いレートで UNDO トランザクションを生成すると、サーバーが UNDO データを保存するための領域が不足する場合があります。UNDO テーブルスペースは循環バッファとして実装されているため、古い UNDO データブロックの上書きを開始します。このエラーを解決し、EDB DDL エクストラクタを再実行して、エラーなしで DDL を抽出します。

このエラーをワークアラウンドには、UNDO_RETENTION パラメーターの割り当て領域を増やします。

最初に、元に戻す操作用に現在割り当てられている合計領域と現在の空き領域を確認します。

1. UNDO_RETENTION パラメーターの既存の値を確認します。

```
SELECT value FROM v$parameter WHERE name = undo_retention;
```

1. 次の 2 つのクエリーの結果を加算して、UNDO テーブルスペースの使用可能な合計領域を確認します。

```
SELECT SUM(bytes)/1024/1024 free_mb FROM dba_undo_extents WHERE status=EXPIRED;
```

```
SELECT dfs.tablespace_name, SUM(dfs.bytes) / 1024 / 1024 AS free_mb FROM
dba_free_space dfs
JOIN dba_tablespaces dt ON dfs.tablespace_name = dt.tablespace_name
WHERE dt.contents = UNDO GROUP BY dfs.tablespace_name; 2 3 4
```

次に、環境が必要な MB 数を決定し、割り当てられた記憶領域を増やします。

1. 使用ピーク時間にトランザクションによって生成された UNDO データの量を監視します。

```
SELECT SUM(undoblks * 8192)/1024/1024 AS undo_mbs FROM v$undostat;
```

注釈

この値は、10 分間隔で消費された UNDO ブロックの数を反映しています MB に変換!!!

1. ご使用の環境が UNDO テーブルスペースに必要な合計 MB 数を計算します。たとえば、トランザクションボリュームが 1 分ごとに X MB の UNDO データを生成し、要件が Y 分間データを保持することである場合

UNDO 領域 MB = X MB/分 × Y (分)

1. 必要な MB 単位の領域に応じて、UNDO_RETENTION パラメーターを更新します。この例では、記憶領域を 2400 MB に増加します。

```
ALTER SYSTEM SET UNDO_RETENTION = 2400;
__OUTPUT__
System altered.
```

5.3 Oracle Data Pump ユーティリティ

- Oracle の引用演算子 IDENTIFIED BY VALUES q'[:1]' などと IDENTIFIED BY 句を使用してデータベースリンクを作成する場合、移行ポータルは SQL ファイルの解析に失敗する場合があります。ファイルを正常に解析するには、実際のパスワード、たとえば、IDENTIFIED BY my_password を使用してみてください。
- Oracle Data Pump ユーティリティによって生成された DDL には、移行ポータルによって処理されない ALTER FUNCTION、ALTER PACKAGE、ALTER TYPE などの ALTER STATEMENTS が含まれる場合があります。
- Oracle Data Pump がスキーマモードで実行される場合、つまり、expdp コマンドの実行時に SCHEMAS パラメーターが使用される場合、プロファイルとロールは抽出されません。エクスポートされるスキーマユーザーにプロファイルが割り当てられている場合、プロファイルはエクスポートされた impdp ファイルの一部ではないため、移行ポータルでのスキーマユーザーオブジェクトの評価は失敗します。この問題を修正するには、ユーザーオブジェクトターゲット DDL のプロファイルの割り当てを削除し、ユーザー作成オブジェクトを再評価します。
- スキーマモードで impdp または expdp を使用して Oracle 11g から ROLES、SYSTEM GRANTS ON ROLES、OBJECT GRANTS ON ROLES、および ROLE GRANTS を抽出しようとしているときに、いくつかのエラーが表示される場合があります。回避策は、full=y オプションを使用することです。これは、エラーなしですべてのスキーマとこれらのオブジェクトタイプを抽出します。

5.4 AI コパイロット

AI Copilot は、DDL の移行中に発生する問題を支援するように設計されたツールです。このツールは問題の解決に非常に役立ちますが、生成 AI テクノロジーは不正確または無関係な応答を生成する場合があります。ことを理解することが重要です。推奨されるソリューションの精度と品質は、[How to create good prompts](#) に大きく影響されます。

提案されたソリューションを運用環境に適用する前に、制御されたテスト環境でソリューションをテストして、提案された修正が特定の移行要件と一致することを確認することを強くお勧めします。

この最初のリリースでは、AI コパイロットの応答時間が予想よりも遅くなる場合があります。