

日々是Oracle APEX

Oracle APEXを使った作業をしていて、気の付いたところを忘れないようにメモをとります。

2021年1月13日水曜日

ビルド・オプション(Build Options)を利用する

Oracle APEXのアプリケーションに組み込んだ機能の有効化/無効化をするために、いくつかの方法が提供されています。本記事では、あまり使われないと思われる**ビルド・オプション (Build Options)** について紹介します。

いままで、**サーバー側の条件**で制御していたけど、ビルド・オプションを使った方が良いかな？と思えることもあるでしょう。

ビルド・オプションの用途

[マニュアル](#)には、ビルド・オプションの3つのユース・ケースが紹介されています。

ユースケース1: インストール・タイプに基づいて機能を有効および無効にする

それぞれのビルド・オプションに紐づいたコンポーネントは完成された状態であり、ユーザーが必要性を判断してビルド・オプションのステータス(含める/除外)を設定します。アプリケーション作成ウィザードでは、機能として情報ページ、アクセス制御、アクティビティ・レポートなどが選択できますが、それがこのユースケースと言えます。

ユースケース2: インストール・タイプに基づいて機能を有効および無効にする

ビルド・オプションの**ステータス**を**含める**にして活用します。以下のシナリオが、このユースケースでの一例になります。

1. 本番稼働中のアプリケーションのコピーを、開発中のアプリケーションとします。
2. 開発中のアプリケーションに新機能を追加します。新機能にビルド・オプションを紐付けます。開発中のアプリケーションでは、**ステータス**は**含める**にします。
3. 開発中アプリケーションを本番稼働中のアプリケーションに置き換えます。新機能のビルド・オプションの**ステータス**は**除外**とし、新機能は一般の利用者に見えないようにします。
4. テストを十分行ったのち、本番稼働中のアプリケーションのビルド・オプションの**ステータス**を**含める**に変更します。この変更の後、利用者は新機能を利用できます。

ユースケース3: 機能の削除の影響を特定する

ビルド・オプションの**ステータス**を**除外**にして活用します。以下のシナリオが、このユースケースでの一例になります。

1. 本番稼働中のアプリケーションのコピーを、開発中のアプリケーションとします。

2. 開発中のアプリケーションから取り除くコンポーネントを、ビルド・オプションに紐付けます。**ステータス**を**除外**にします。
3. 開発中のアプリケーションで、コンポーネントを除外しても問題が発生しないことを確認します。
4. 開発中のアプリケーションを本番稼働中のアプリケーションに置き替えます。ビルド・オプションの**ステータス**は**除外**のままです。
5. 本番稼働中のアプリケーションでも問題がないことが確認できたら、ビルド・オプションに紐づいている**コンポーネントをアプリケーションから削除**します。

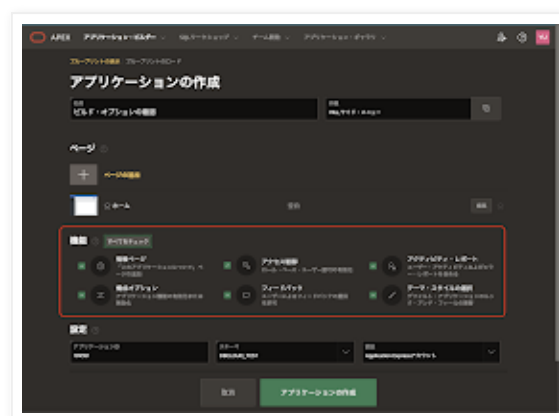
アプリケーションをバージョンアップする際には、新機能は追加対象、旧機能は除外対象として、双方のユースケース（2と3）がひとつのアプリケーションに同居することもあるでしょう。

動作確認用のアプリケーションの作成

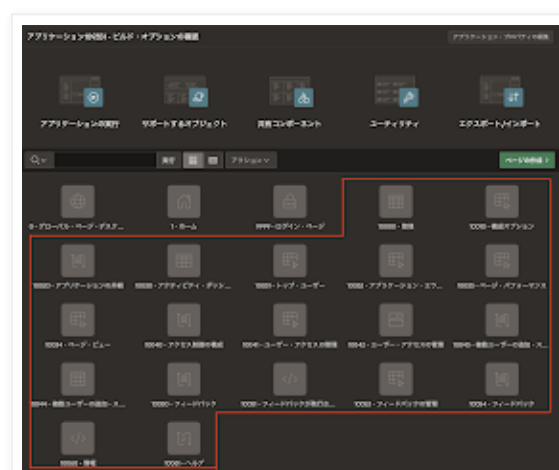
最近のOracle APEXが提供しているアプリケーション作成ウィザードでは、**機能**としてあらかじめ、**情報ページ、アクセス制御、アクティビティ・レポート、構成オプション、フィードバック、テーマスタイルの選択**、を選んで組み込むことができます。これらの機能はそれぞれビルド・オプションに紐づいています。これらの機能をすべて組み込んだ空のアプリケーションを作成し、それを使ってビルド・オプションの働きを確認します。

アプリケーション・ビルダーから**作成**を実行し、**アプリケーション作成ウィザード**を起動します。

名前をビルド・オプションの確認とし、**機能のすべてをチェック**をクリックし、すべての機能を選択したのち、**アプリケーションの作成**を実行します。



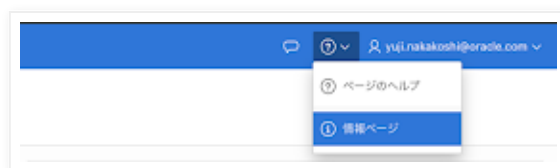
アプリケーションが作成されます。ページ番号が10000以上のページはすべて、上記でチェックを入れた機能と関連しているページです。



作成されたアプリケーションを実行します。**管理**ページを開くと、選択した機能に対応したリージョンの存在を確認できます。左上、左下、右上、右下の順番で、構成オプション、テーマ・スタイルの選択、アクティビティ・レポート、アクセス制御、フィードバックです。



情報ページがここに含まれていませんが、こちらは右上のメニューから呼び出される機能です。



ビルド・オプションを確認します。共有コンポーネントの**アプリケーション・ロジック**に含まれます。



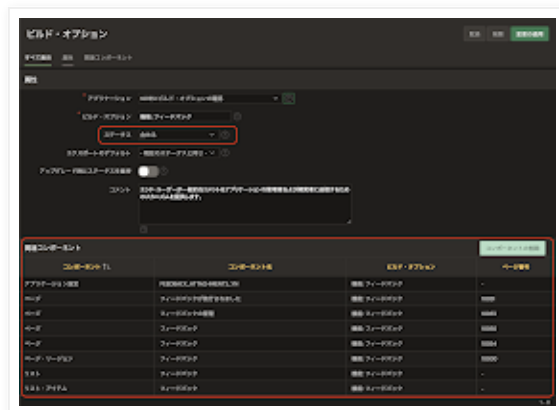
アプリケーション作成ウィザードで選択した6つの機能が、ビルド・オプションとして定義されています。

ビルド・オプションのステータス

この中から、**フィードバック**の機能を取り上げて、ビルド・オプションの働きについて確認します。



機能：フィードバックをクリックして定義を開きます。



現在のビルド・オプションのステータスは、**含める**になっています。つまり、**関連コンポーネント**としてリストされている、それぞれのコンポーネントが有効になっています。ステータスを**除外**に変更すると、関連コンポーネントは実際には削除されませんが、存在していないのと同様の扱いになります。また、ここで**コンポーネントの削除**を行うとデータとしても削除されます。

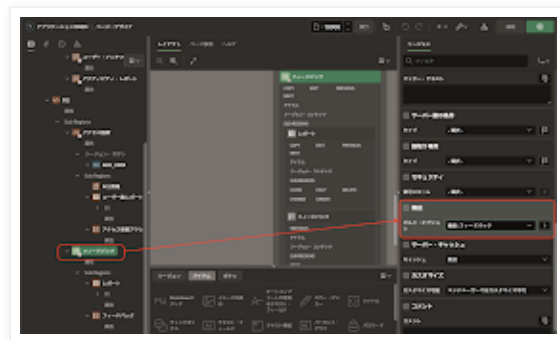
先ほどの管理ページを開きます。ここで、フィードバックに関するメニューやリージョンを赤枠で囲っています。



機能：フィードバックのステータスを**除外**に変更すると、表示が以下に変わり、メニューやリージョンが無くなります。

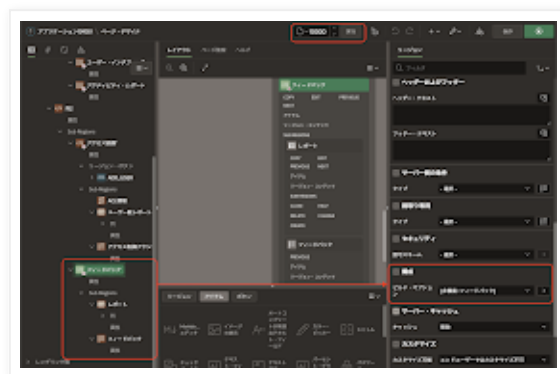


リージョンでのビルド・オプションの指定は、プロパティ・エディタの構成に含まれています。



ビルド・オプションとして機能：フィードバックが指定されているので、このビルド・オプションのステータスが含めるとなっているときに、有効になります。反対の効果となる選択肢として、{非機能：フィードバック}を選ぶことができます。この指定の場合は、ステータスが除外のときに有効になります。

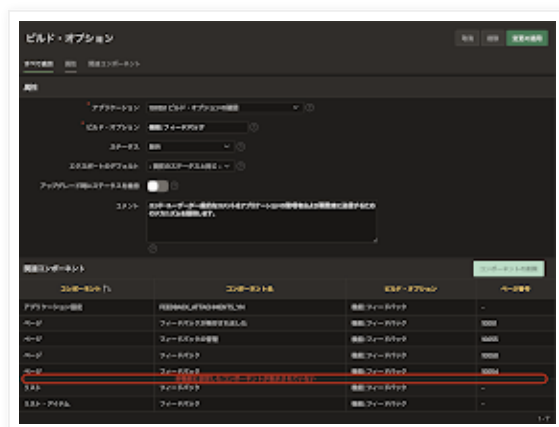
ただし、この非機能の指定は、どうしても必要な場合に限定した方がよいでしょう。非機能の指定は以下になります。



ビルド・オプション機能：フィードバックのステータスが除外のときに、ページ番号10000を開くと以下ようになります。



メニュー上のフィードバックを呼び出す項目は消去されたままで、**ビルド・オプション**として**{非機能：フィードバック}**を設定したリージョンは表示されます。これは設定した通りの効果です。
共有コンポーネントのビルド・オプションから、**機能：フィードバック**の状態を確認します。



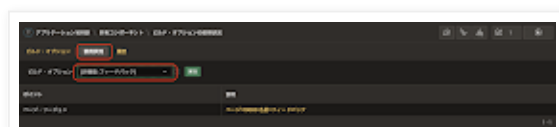
ビルド・オプションに**{非機能：フィードバック}**を設定したコンポーネントが、関連コンポーネントに現れていないことが確認できます。**機能：フィードバック**を設定していたときの関連コンポーネントのリストは以下でした。ページ番号**10000**の**ページ・リージョン**、**フィードバック**が無くなっていることがわかります。



非機能として設定されている**ビルド・オプション**があると、**コンポーネントの削除**に失敗します。



使用状況のレポートを表示することにより、**ビルド・オプション**が**非機能**で設定されているコンポーネントを調べることができます。

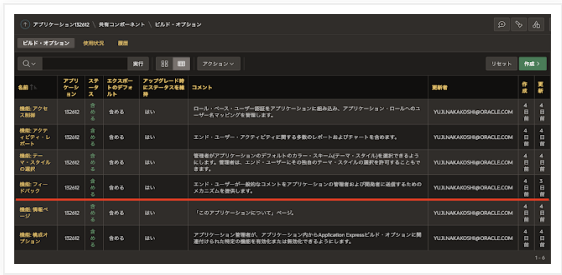


このように、ビルド・オプションのステータスを**含める**または**除外**とすることにより、そのビルド・オプションに紐づいた機能を有効にしたり、無効にしたりできます。

エクスポートのデフォルト

テスト環境ではビルド・オプションのステータスを**含める**に設定し、本番環境では一旦除外したい状況を考えます。

ここでは仮に**機能: フィードバック**を開発中とします。



名前	ステータス	タイプ	コメント	操作者	作成日時	更新日時
機能: フォー ドバック	含める	ビルド・オプション	ローカル開発、ユーザー機能もアプリケーションに組み込み、アプリケーション・ロールのユーザーがフィードバックを提出します。	YUUKA@GIGAMONK.COM	4月15日	4月15日
機能: アダ プティブな スタイル	含める	ビルド・オプション	エンド・ユーザー・アダプティブに関する多数のレポートおよびフィードバックを待ちます。	YUUKA@GIGAMONK.COM	4月15日	4月15日
機能: テ クススタイル の適用	含める	ビルド・オプション	開発者のアプリケーションのフェイルドのカラー・スキームテーマ・スタイルを適用できるようにします。開発者は、エンド・ユーザーとその独自のテーマ・スタイルの選択を許可することもできます。	YUUKA@GIGAMONK.COM	4月15日	4月15日
機能: フィ ードバック	含める	ビルド・オプション	エンド・ユーザー・提供のコメントもアプリケーションの管理画面より開発者に送信するためのメカニズムを期待します。	YUUKA@GIGAMONK.COM	4月15日	4月15日
機能: 開発 モード	含める	ビルド・オプション	「このアプリケーションについて」ページ。	YUUKA@GIGAMONK.COM	4月15日	4月15日
機能: 開発 オプション	含める	ビルド・オプション	アプリケーション開発者が、アプリケーション内からApplication Explorerから、オプションを開発者が利用できる機能を開発するための開発に使用できるようにします。	YUUKA@GIGAMONK.COM	4月15日	4月15日

この開発中アプリケーションをエクスポートし、本番環境にインポートすると、**エクスポートのデフォルト**が引き継がれ、本番環境での**ステータス**は**含める**になります。

本番環境にインポートしたときに、**機能: フィードバック**がデフォルトで**除外**となってほしい場合を考えます。ステータスを一旦**除外**にし、エクスポートしてから、再度**含める**に戻す、というのも手間がかかります。

ビルド・オプションの属性の**エクスポートのデフォルト**を**除外**に設定すると、この後にエクスポートされたアプリケーションをインポートすると、エクスポート時点のビルド・オプションのステータスにかかわらず、除外がデフォルトになります。



ビルド・オプション

すべて表示 削除 関連コンポーネント

属性

アプリケーション 12345 ビルド・オプションの属性

ビルド・オプション 機能: フィードバック

ステータス 含める

エクスポートのデフォルト 除外

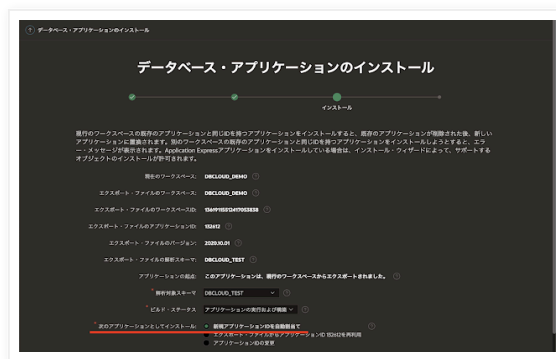
アップグレード時にステータスを維持 ☒

コメント エンド・ユーザー・提供のコメントもアプリケーションの管理画面より開発者に送信するためのメカニズムを期待します。

エクスポート前のアプリケーションでのビルド・オプションの設定は以下のように、**エクスポートのデフォルト**が**除外**です。

7 アプケーションコンパイル、両端コンポーネント、ビルド・オプション									
ビルド・オプション		修訂履歴	変更						
検索		実行	詳細	アクション				リセット	作成
名前	アプリ タイプ	ス タ タ ス	エクスポート パス	アップグレード パスを保持	コメント	変更者	作 成 日	修 正 日	
標準アプレーション	151452	正常	はい		ローカル・ユーザ・管理をアプリケーションに組み込み、アプリケーション・ローカル・ユーザ・管理をオンプラインに変更します。	YULIYNAKAKOSH@ORACLE.COM	4月11日	4月11日	
標準アクティブディレクトリ	151452	正常	はい		エンド・ユーザー・アクティビティに関する多数のレポートおよびチャットを有効にします。	YULIYNAKAKOSH@ORACLE.COM	4月11日	4月11日	
標準データベースアクセス	151452	正常	はい		管理側がアプリケーションのリアルタイム・スクリーンで、ステイプル・メニューを開くことができます。管理側は、スクリーン上の特定のメニュー・アイテムを選択許可を行います。	YULIYNAKAKOSH@ORACLE.COM	4月11日	4月11日	
標準フィードバック	151452	正常	はい	新機	エンド・ユーザー・管理にコメントをアプリケーションの管理側および関係者に送信するなどの機能を有効にします。	YULIYNAKAKOSH@ORACLE.COM	4月11日	5月11日	
標準管理ページ	151452	正常	はい		1つのアプリケーションについて、ページ	YULIYNAKAKOSH@ORACLE.COM	4月11日	4月11日	
標準開発オプション	151452	正常	はい		アプリケーション管理側で、アプリケーションとApplication Expressビルド・オプションに関連付けられた機能の有効化は非推奨されるようになります。	YULIYNAKAKOSH@ORACLE.COM	4月11日	4月11日	

エクスポートしたアプリケーションを以下の画面で、**次のアプリケーション**としてインストールとして**新規アプリケーションIDを自動割当て**を選んでインポートします。

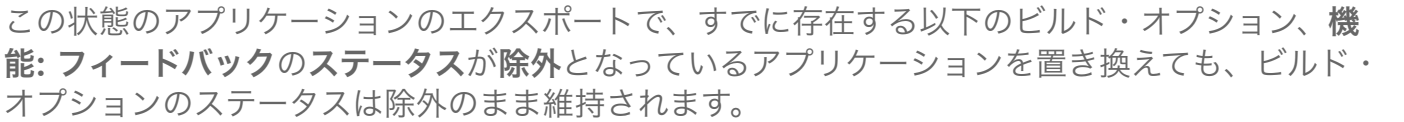


新規に作成されたアプリケーションのビルド・オプションは、エクスポートのデフォルトとして設定されていたステータスが反映されます。

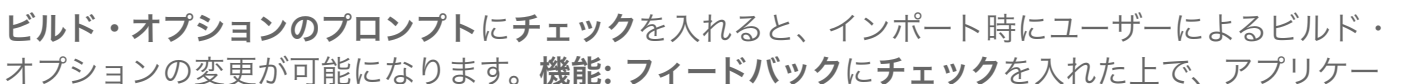
アップグレード時にステータスを維持

インポート時に既存のアプリケーションを置き換えることが可能な場合は、**アップグレード時にステータスを維持**をはいに設定することで、置き換えられるアプリケーションの設定を維持することができます。

エクスポート時のビルド・オプションのエクスポートのデフォルトが含まれる、アップグレード時にステータスを維持がはい、だったとします。



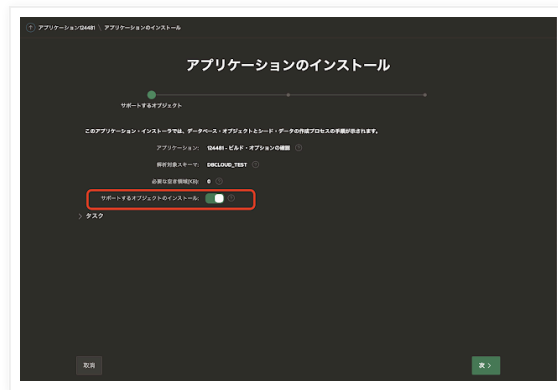
アプリケーションのインポート時にビルド・オプションを設定することも可能です。アプリケーションの**サポートするオブジェクト**を開きます。



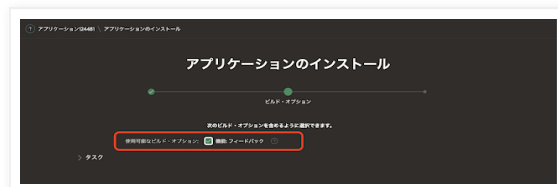
ションをエクスポートし、インポートしてみます。



アプリケーションのインポートを実行します。ビルド・オプションの設定を可能にするには、**サポートするオブジェクトのインストール**をONに指定する必要があります。



使用可能なビルド・オプションとして、ビルド・オプションのプロンプトでチェックを入れたビルド・オプションが現れます。**含める**にするか**除外**にするか、**チェック**を入れることで指定できます。



インポートされたアプリケーションのビルド・オプションのステータスは、ここで設定されたステータスになります。

APEX_UTIL.SET_BUILD_OPTION_STATUSの使用

コードからビルド・オプションを指定するAPIが提供されています。
APEX_UTIL.SET_BUILD_OPTION_STATUSです。以下のように呼び出します。

```
begin
  apex_util.set_build_option_status(
    p_application_id => 97521,
    p_id => 26026092390449248099,
    p_build_status => 'EXCLUDE'
  );
end;
```

p_idはビルド・オプションのIDになります。**APEX_APPLICATION_BUILD_OPTION_STATUS**ビューから確認することができます。以下のようなSQLにてビルド・オプション・IDを取得します。

```
select build_option_id
```

```
from apex_application_build_options
where workspace = 'DBCLOUD_DEMO'
and application_id = 97521
and feature_identifier = 'APPLICATION_FEEDBACK';
```

ビルド・オプションのステータスについては、[APEX_UTIL.GET_BUILD_OPTION_STATUS](#)を呼び出すことによって取得できます。以下のようなスクリプトになります。

```
select
APEX_UTIL.GET_BUILD_OPTION_STATUS(
    p_application_id => 97521,
    p_id => 26026092390449248099
)
from dual;
```

インストール済みのアプリケーションに含まれるビルド・オプションであれば、APIを呼び出した時点で変更されます。

アプリケーションのインストール・スクリプト内でビルド・オプションを設定する場合は、インポート時に既存のアプリケーションを置き換えること、および、**サポートするオブジェクトをインストールするをON**とすることが前提です。また、**アップグレード時にステータスを維持をはい**に設定します。

APIによるビルド・オプションの設定は、ビルド・オプション自体が定義済みであることを前提としているため、インポート時に新規にアプリケーションIDが割り振られる場合は利用できません。また、**アップグレード時にステータスを維持がはい**の場合は、APIによる変更が**エクスポートのデフォルト**として設定されている**ステータス**で上書きされてしまいます。

終わりに

以上がビルド・オプションにまつわる設定の紹介になります。コンポーネントの有効化/無効化は、サーバー側の条件や認可スキームを使うことが多いですが、ビルド・オプションを使うとひとつの設定でアプリケーション全体を変更できるため、便利に使うことができます。

完

Yuji N. 時刻: 16:26

共有

◀

ホーム

▶

[ウェブ バージョンを表示](#)

自己紹介

Yuji N.

日本オラクル株式会社に勤務していて、Oracle APEXのGroundbreaker Advocateを拝命しました。こちらの記事につきましては、免責事項の参照をお願いいたします。

[詳細プロフィールを表示](#)

