



WorkWithPlus トレーニング 実習

For WorkWithPlus 15

2023/06 初版

準備 – 課題実施のために	6
準備-1. 前提の準備	6
1 – 柔軟性	9
1-1. 「デザイン システム ウィザード」の実行	9
2.1 – Transaction	11
2.1-1. パターンの適用	11
※パターンの適用補足	11
2.1-2. 社員 (Person) のレイアウト	12
※社員 (Person) のレイアウト補足	13
2.1-3. 会社 (Company) のレイアウト 1	14
2.1-4. 会社 (Company) のレイアウト 2	15
※会社 (Company) のレイアウト 2 補足	15
2.1-5. 入力方法の変更	16
2.1-6. 必須入力の設定	17
2.1-7. 条件付き必須入力の設定	18
2.1-8. 編集不可の設定	19
2.1-9. 条件付き非表示の設定	20
2.1-10. 登録条件の追加	21
※登録条件の追加補足	21
2.1-11. 管理項目の追加	22
2.2 – List	24
2.2-1. 不要な項目の削除または非表示	24
2.2-2. 1 ページあたりの表示件数の変更	25
2.2-3. フィルタの追加	26
※フィルタの追加補足	26
2.2-4. 動的なフィルタの変更	27
※動的なフィルタの変更補足	27
2.2-5. 固定表示のフィルタ追加	28
2.2-6. タイトルフィルタのカスタマイズ	29
2.2-7. 参照画面を呼び出すアクションの追加	30
※参照画面を呼び出すアクションの追加補足	30
2.2-8. 削除を実行するアクションの追加	31
※削除を実行するアクションの追加補足	31
2.2-9. 休暇期間を延長するアクションの追加	32
※休暇期間を延長するアクションの追加補足	32
2.2-10. 1 列で表示する会社情報の追加	33
※1 列で表示する会社情報の追加補足	33
2.3 – View	35
2.3-1. 「追加情報」タブの追加	35

2.3-2. 「監査情報」パネルの追加	36
2.3-3. Autolink の無効化	37
3.1 – 既定値の設定	39
3.1-1. 既定の演算子の利用	39
3.1-2. 全画面を対象にしたタイトルフィルタのカスタマイズ	40
※全画面を対象にしたタイトルフィルタのカスタマイズ補足	40
3.2 – テンプレート	42
3.2-1. 自動ルールの追加	42
※自動ルールの追加補足	42
3.2-2. 実行モードの表示	43
※実行モードの表示補足	43
3.3 – インスタンスの更新 / 適用	45
3.3-1. 全登録画面のレイアウト変更	45
※全登録画面のレイアウト変更補足	45
4 – Web パネル	47
4-1. 固定日数の社員休暇期間 (PersonHolidays) 登録画面	47
※固定日数の社員休暇期間 (PersonHolidays) 登録画面補足	47
4-2. 会社 (Company) と社員 (Person) の同時登録	48
※会社 (Company) と社員 (Person) の同時登録補足	48
4-3. 会社一覧画面の新規作成	49
※会社一覧画面の新規作成補足	49
5 – PDF レポート	51
5-1. 社員休暇期間一覧出力	51
※社員休暇期間一覧出力補足	51
6 – セキュリティ	53
6-1. セキュリティ機能の有効化	53
※セキュリティ機能の有効化補足	53
6-2. 監査用項目への自動割り当て値更新	54
※監査用項目への自動割り当て値更新補足	54
6-3. ユーザーの追加	55
※ユーザーの追加補足	56
7 – 監査	58
7-1. 監査機能の有効化	58
※監査機能の有効化補足	58

課題

あなたはプロジェクトに参加しているメンバーの 1 人です。

このプロジェクトでは、エンドユーザーからの要望で、複数の会社とその社員を管理する Web アプリケーションの開発を行います。

また、これ以外にも関連したデータを多数管理しています。

本書では、教材で紹介した機能に基づくエンドユーザーからの要望が段階的に記載されています。

教材を参考に実装を進めてください。



WORKWITHPLUS
FOR WEB

WorkWithPlus for Web トレーニング

準備 – 課題実施のために

準備 – 課題実施のために

準備-1. 前提の準備

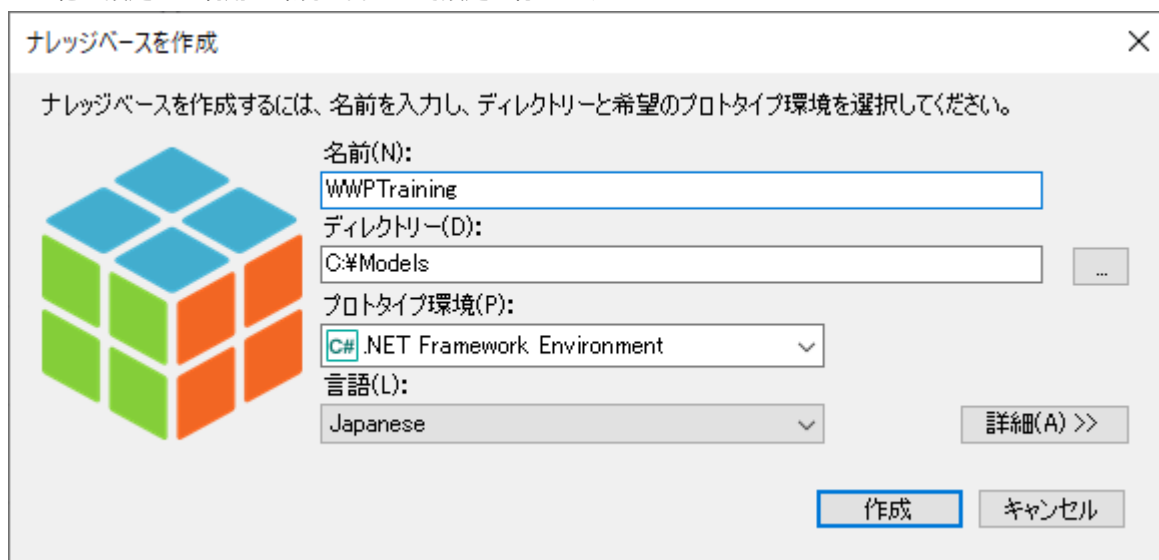
「課題」として定義した通り、本書では Web アプリケーションの開発を題材に、教材で説明した機能を利用していきます。

ただし、トレーニングコース本来の対象ではない GeneXus のみの機能を利用した範囲は、事前に用意しているものを利用します。

そのため、下記ステップにそって操作を行い、前提の準備を完了させます。

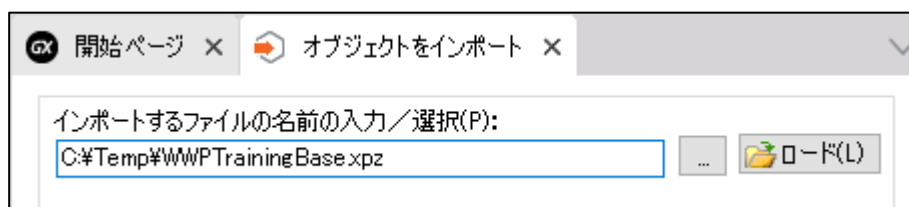
1. GeneXusを起動し、メニューバーの [ファイル(F)] → [新規(N)] → [ナレッジベース] をクリックし、表示されるダイアログにて下記の設定を行い、「作成」ボタンをクリックします。

名前： WWPTTraining
プロトタイプ環境： .NET Framework Environment
その他の設定はご利用の環境に合わせて設定を行います。



2. 前提となるオブジェクトを GeneXus のインポート機能で取り込みます。

- ① メニューバーの [ナレッジマネージャ(K)] → [インポート] をクリック
- ② 表示される「オブジェクトのインポート」画面にて、本書と同梱されていた「WWPTTrainingBase.xpz」を [インポートするファイルの名前の入力/選択(P):] で指定
※ファイルの場所は任意の場所でも可



- ③ 画面右側の [ファイル内のオブジェクト] にオブジェクトが表示されることを確認し、「インポート」ボタンをクリック

3. 「すべてビルド」を実行し、「ビルドプロセスに必要なプロパティを設定」ダイアログで下記の設定を行い、「完了」ボタンをクリックします。

Database name : WWPTTraining

Server name : localhost¥sqlexpress

4. 「影響分析」ウィンドウで「作成」ボタンをクリックします。

5. 「BaseObjects」モジュール内の「SampleData」フォルダに含まれる「CreateSampleData」プロシージャオブジェクトを右クリックし、[実行] をクリックします。

以上で課題を実施するための準備が完了しました。



WORKWITHPLUS



WORKWITHPLUS
FOR WEB

WorkWithPlus for Web トレーニング

1 - 柔軟性

1 – 柔軟性

1-1. 「デザイン システム ウィザード」の実行

エンドユーザーは、WorkWithPlus for Web の紹介を受けていたため、この Web アプリケーションには、WorkWithPlus for Web を利用したいと要望しています。

WorkWithPlus for Web の利用を始めるために、「デザイン システム ウィザード」の実行が必要です。エンドユーザーからの初期設定としては次の要望が出ているため、対応できるように設定を行います。

- ① デザインシステムは「Seine」を利用
- ② セキュリティ機能は現時点では不要
- ③ ホームページとして「バナー」と「モジュール」が表示されるものを利用



WORKWITHPLUS
FOR WEB

WorkWithPlus for Web トレーニング

2 – トランザクションオブジェクト

への適用

2.1 – Transaction

2.1 – Transaction

2.1-1. パターンの適用

エンドユーザーから会社（Company）、社員（Person）、社員休暇期間（PersonHolidays）に関して、登録されているデータをまずは一覧で表示し、必要に応じて新規登録、更新、削除の画面が呼び出せるようにしたいと要望があります。

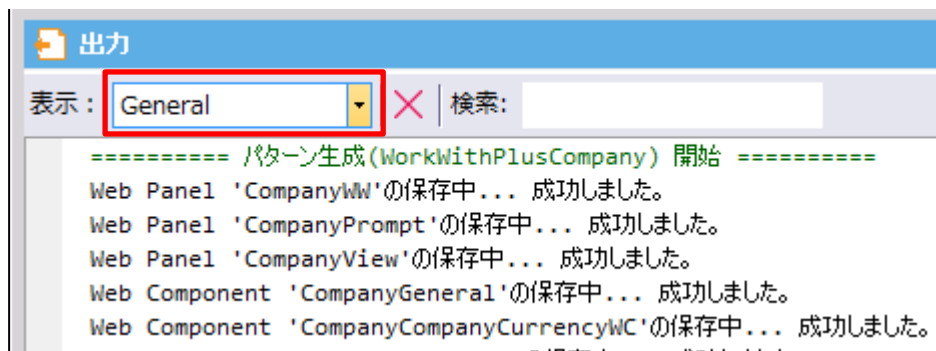
現時点で受けている要望について改めて以下に記載します。

- ① 対象画面（オブジェクト）は、会社（Company）、社員（Person）、社員休暇期間（PersonHolidays）
- ② 一覧画面の実装は標準的なものを希望
- ③ 一覧画面には、全項目の表示を希望
- ④ 一覧画面のカスタマイズについて要望なし

要望を満たせたら一度実行し、生成された Web アプリケーションをエンドユーザーに確認してもらいましょう。（GeneXus で実行し、生成された画面を確認します）

※パターンの適用補足

パターンを適用する場合、[出力] ウィンドウ（メニューバーの [表示(V)] → [その他のツールウィンドウ] → [出力]）の [表示] コンボボックスを [General] に設定することで、パターンによるオブジェクト生成で、問題が起きているかどうか確認することができます。



2.1-2. 社員（Person）のレイアウト

エンドユーザーが社員（Person）の一覧で、[追加] ボタンを押下し、新規登録画面を表示したところ、項目数が多いため、縦に長い画面となっていることに気が付きました。
そのため、この画面がスクロールせずに済むようにいくつかの項目ごとにタブにまとめて表示することを要望しています。

具体的な要望について以下に記載します。

- ① タブは「詳細情報」、「追加情報」、「社員メール」の3つを希望
- ② 「詳細情報」タブは、1行2列で次の7項目を表示
社員番号、社員姓、社員名、社員フルネーム、社員ニックネーム、社員性別、社員生年月日
表示イメージ：

社員番号	社員姓
社員名	社員フルネーム Value
社員ニックネーム	社員性別 男性
社員生年月日	31

- ③ 「追加情報」タブは、残りの第1レベルの11項目すべてを表示
ただし、監査用の値を格納する「社員登録日」、「社員登録者」、「社員更新日」、「社員更新者」については、タイトル付きのテーブル（パネル）に1行2列で表示
表示イメージ：

社員お気に入りバンド有無	<input type="checkbox"/>
社員お気に入りバンド	
社員趣味	
社員住所必須	<input type="checkbox"/>
社員住所	
社員写真	
会社	

監査情報

社員登録日	// 12:00 31	社員登録者	
社員更新日	// 12:00 31	社員更新者	

（次ページへ続く）

- ④ 「社員メール」タブは、メールアドレスを入力するための明細（グリッド）を表示
表示イメージ：

社員メールアドレス	社員メール詳細

- ⑤ 各ボタンについては、タブ外の表示を希望

要望を満たせたら一度実行し、生成された Web アプリケーションをエンドユーザーに確認してもらいましょう。
（GeneXus で実行し、生成された画面を確認します）

※社員（Person）のレイアウト補足

タブを利用したレイアウトのデザイン時には、テンプレートを変更し、既定のテンプレートから
「タブ付きデータ」というテンプレートを利用することで簡単に変更が行えます。

2.1-3. 会社（Company）のレイアウト 1

エンドユーザーが会社（Company）の一覧で、[追加] ボタンを押下し、表示される新規登録画面で、社員（Person）と同じようにタイトル付きのテーブル（パネル）を利用し、監査用項目を表示することを要望しています。

このタイトル付きのテーブルはもともとある「詳細情報」というタイトル付きのテーブルと並行して表示し、初期表示時は折りたたまれた状態となることを要望しています。

現時点で受けている要望について改めて以下に記載します。

- ① 監査用の値を格納する「会社登録日」、「会社登録者」、「会社更新日」、「会社更新者」をタイトル付きのテーブル（パネル）に 1 行 2 列で表示
- ② 表示は「詳細情報」タイトル付きのテーブル（パネル）と並行して表示
つまり、「詳細情報」タイトル付きのテーブル（パネル）の外で表示
- ③ 追加するタイトル付きのテーブル（パネル）は折りたたまれた状態で初期表示
表示イメージ：



要望を満たせたら一度実行し、生成された Web アプリケーションをエンドユーザーに確認してもらいましょう。
（GeneXus で実行し、生成された画面を確認します）

2.1-4. 会社（Company）のレイアウト 2

エンドユーザーが直前の要望に基づく実行画面を確認したところ、追加の要望が出てきました。
監査用項目を別のタイトル付きのテーブル（パネル）に移動したため、利用時に処理を実行するボタンの表示位置を 2 つのタイトル付きのテーブル（パネル）の間に移動してほしいと要望しています。

現時点で受けている要望について改めて以下に記載します。

- ① 「実行」や「終了」ボタンなどあらかじめ用意されているボタンの位置の変更
- ② 移動先は「詳細情報」タイトル付きのテーブル（パネル）と「監査情報」タイトル付きのテーブル（パネル）の間

要望を満たせたら一度実行し、生成された Web アプリケーションをエンドユーザーに確認してもらいましょう。
（GeneXus で実行し、生成された画面を確認します）

※会社（Company）のレイアウト 2 補足

移動の希望があったボタンはすべて [Action Group] ノードに含まれているので、このノードを移動することで一括変更できます。

2.1-5. 入力方法の変更

エンドユーザーから社員（Person）の新規登録画面について追加で変更の要望がありました。

「社員性別」の入力がコンボボックス形式になっているが、入力時はラジオボタン形式にし、選択肢は横並びで表示したいと要望しています。

現時点で受けている要望について改めて以下に記載します。

- ① 「社員性別」のデータ入力方式を「ラジオボタン」に変更
- ② 「ラジオボタン」の選択肢は「縦」ではなく、「横」に並べて表示
- ③ この変更は社員（Person）でのデータ入力時のみ適用

要望を満たせたら一度実行し、生成された Web アプリケーションをエンドユーザーに確認してもらいましょう。
（GeneXus で実行し、生成された画面を確認します）

2.1-6. 必須入力の設定

エンドユーザーから社員（Person）の新規登録画面について追加で変更の要望がありました。

「社員姓」、「社員名」、「社員生年月日」については必ず入力する必要があるため、値の入力がない状態で登録できないようにしたいと要望しています。

値の入力がない場合には必須入力であることを示すメッセージを表示することも要望しています。

このメッセージについては、必須であることがわかれば既定のままでも問題ありません。

現時点で受けている要望について改めて以下に記載します。

- ① 「社員姓」、「社員名」、「社員生年月日」を必須入力項目として設定
- ② 値が未入力の場合、必須入力であることを示すメッセージの表示
- ③ メッセージについて既定の内容から変更する必要がない

要望を満たせたら一度実行し、生成された Web アプリケーションをエンドユーザーに確認してもらいましょう。

（GeneXus で実行し、生成された画面を確認します）

2.1-7. 条件付き必須入力の設定

エンドユーザーから社員（Person）の新規登録画面について追加で変更の要望がありました。

「社員住所」については、「社員住所必須」にチェックされている場合のみ必ず入力する必要があります。

そのため、「社員住所必須」がチェックされ、「社員住所」に値の入力がない状態で登録できないようにしたいと要望しています。

上記条件を満たす場合には必須入力であることを示すメッセージを表示することも要望しています。

このメッセージについては、必須であることがわかれば既定のままでも問題ありません。

現時点で受けている要望について改めて以下に記載します。

- ① 「社員住所」を「社員住所必須」がチェックされている場合に必須入力項目として設定
- ② 値が未入力の場合、必須入力であることを示すメッセージの表示
- ③ メッセージについて既定の内容から変更する必要がない
- ④ （「社員住所必須」の既定値は未チェック状態のため、「社員住所」は既定で必須入力ではない）

要望を満たせたら一度実行し、生成された Web アプリケーションをエンドユーザーに確認してもらいましょう。

（GeneXus で実行し、生成された画面を確認します）

2.1-8. 編集不可の設定

エンドユーザーから社員（Person）および会社（Company）の新規登録画面について追加で変更の要望がありました。

監査用の値を格納する各画面の「登録日」、「登録者」、「更新日」、「更新者」の項目はエンドユーザー自身で入力せず、システムから値を入力するように今後設定を行いたいため、手入力できる状態は意図した状態ではありません。

そのため、これらの項目を編集不可に設定したいと要望しています。

現時点で受けている要望について改めて以下に記載します。

- ① 社員（Person）にて、下記の項目を編集不可に設定
「社員登録日」、「社員登録者」、「社員更新日」、「社員更新者」
- ② 会社（Company）にて、下記の項目を編集不可に設定
「会社登録日」、「会社登録者」、「会社更新日」、「会社更新者」
- ③ 現時点でこれらの項目にシステムから値を入力するための実装は不要

要望を満たせたら一度実行し、生成された Web アプリケーションをエンドユーザーに確認してもらいましょう。
（GeneXus で実行し、生成された画面を確認します）

2.1-9. 条件付き非表示の設定

エンドユーザーから社員（Person）の新規登録画面について追加で変更の要望がありました。

「社員お気に入りバンド」については、「社員お気に入りバンド有無」にチェックされている場合のみ表示し、チェックされていない場合には非表示にしたいと要望しています。

また、「社員お気に入りバンド」は初期表示時には非表示の状態にしたいと要望しています。

現時点で受けている要望について改めて以下に記載します。

- ① 「社員お気に入りバンド」を「社員お気に入りバンド有無」がチェックされている場合に表示
- ② 「社員お気に入りバンド」は初期表示時に非表示

要望を満たせたら一度実行し、生成された Web アプリケーションをエンドユーザーに確認してもらいましょう。

（GeneXus で実行し、生成された画面を確認します）

2.1-10. 登録条件の追加

エンドユーザーから社員（Person）の新規登録画面について追加で変更の要望がありました。
この Web アプリケーションで管理する各会社で登録できる社員は年齢が 18 歳以上という条件があります。
万が一 18 歳未満の社員が登録されないように設定を行ってほしいと要望しています。

具体的な要望について以下に記載します。

- ① 社員（Person）で登録できる社員は 18 歳以上
- ② 18 歳未満の社員が入力された場合、登録できない
- ③ 登録できない場合、理由がわかるようにメッセージを画面に表示

要望を満たせたら一度実行し、生成された Web アプリケーションをエンドユーザーに確認してもらいましょう。
（GeneXus で実行し、生成された画面を確認します）

※登録条件の追加補足

WorkWithPlus for Web によって自動生成されるルールの範囲外に追加するルールを「ユーザールール」と呼んでいました。

この課題ではまさにこのユーザールールを追加する必要があります。

GeneXus では、年齢を計算するための関数として「Age」関数が用意されているため、この関数を利用し、年齢を計算、この値が 18 未満であればエラーとなるようにルールを追加します。

2.1-11. 管理項目の追加

エンドユーザーから社員（Person）で管理する項目の追加要望がありました。

この Web アプリケーションで管理する各会社の管理職社員を管理する必要があります。

そのため、管理職であるかどうかをチェックボックスで指定できるように項目を追加してほしいと要望しています。

この項目は、社員（Person）に関連して生成される各画面に表示することも要望しています。

具体的な要望について以下に記載します。

- ① 社員（Person）に「会社管理職」（PersonIsDirector）という項目を追加
- ② 追加する項目はチェックボックス（Boolean 型）で入力できるようにする
- ③ 社員（Person）に関連して自動生成される各画面でも表示できるようにする

要望を満たせたら一度実行し、生成された Web アプリケーションをエンドユーザーに確認してもらいましょう。

（GeneXus で実行し、生成された画面を確認します）



WORKWITHPLUS
FOR WEB

WorkWithPlus for Web トレーニング

2 – トランザクションオブジェクト

への適用

2.2 – List

2.2 – List

2.2-1. 不要な項目の削除または非表示

エンドユーザーが社員（Person）の一覧画面を確認し、項目数が多いため、横にスクロールする必要がある画面となっていることに気が付きました。

改めて確認したところ、一覧に不必要な項目があることが確認できました。

また、特定のユーザーだけが必要とする項目もあるため、既定では非表示にし、必要に応じて表示できるようにしたいと要望がありました。

具体的な要望について以下に記載します。

- ① 次の項目は一覧に不必要です。
「社員お気に入りバンド有無」、「社員住所必須」、「社員登録日」、「社員登録者」、
「社員更新日」、「社員更新者」
- ② 次の項目は既定では非表示にし、必要に応じて表示
「社員番号」、「社員姓」、「社員名」、「社員ニックネーム」、「社員お気に入りバンド」、
「社員住所」、「会社番号」、「会社管理職」
- ③ 上記要望 2 点の結果、表示される項目は次の項目となる
「社員フルネーム」、「社員性別」、「社員生年月日」、「社員趣味」、「社員写真」、「会社名」

要望を満たせたら一度実行し、生成された Web アプリケーションをエンドユーザーに確認してもらいましょう。
(GeneXus で実行し、生成された画面を確認します)

2.2-2. 1 ページあたりの表示件数の変更

エンドユーザーが社員休暇期間（PersonHolidays）の一覧画面を確認し、登録データが多いため、ページングが行われ、このページも多数になっていることを確認しました。

休暇期間情報については、一度により多くのデータを確認できるようにしたいため、1 ページあたり 20 件のデータを表示するように変更を要望しています。

現時点で受けている要望について改めて以下に記載します。

- ① 社員休暇期間（PersonHolidays）の一覧画面は 1 ページあたり「20 件」のデータを表示
- ② 表示件数は「20 件」であり、これは今後も変更要望が出る予定はない

要望を満たせたら一度実行し、生成された Web アプリケーションをエンドユーザーに確認してもらいましょう。
（GeneXus で実行し、生成された画面を確認します）

2.2-3. フィルタの追加

エンドユーザーが社員（Person）の一覧画面を確認し、フィルタが実装されていることを確認しました。
このフィルタに利用できる項目が足りないため、追加の要望がありました。
既定で実装されているフィルタ項目のコンボボックスに追加するのではなく、単独のフィルタとして追加してほしいという要望です。

具体的な要望について以下に記載します。

- ① フィルタはコンボボックスに追加ではなく、単独のフィルタとして追加
- ② 「社員性別」をフィルタとして追加。フィルタのタイプについて既定のものを利用
- ③ 「社員登録日」を「動的な演算子が含まれたフィルタ」として追加。
- ④ 「社員更新日」を「範囲フィルタ」として追加。

要望を満たせたら一度実行し、生成された Web アプリケーションをエンドユーザーに確認してもらいましょう。
(GeneXus で実行し、生成された画面を確認します)

※フィルタの追加補足

もし、「社員更新日」が「範囲フィルタ」以外のフィルタタイプとして追加された場合、追加されたノードを右クリックし、「範囲フィルタ' に変換」を実行することで変更可能です。

2.2-4. 動的なフィルタの変更

エンドユーザーが社員（Person）の一覧画面に対し要望したフィルタの追加を確認したところ、追加したフィルタによって画面のデザインから統一性がなくなったように感じられるとのことで、既定で実装されているコンボボックスに先ほど追加したフィルタを含めるようにしたいと要望しています。

また、そのほかにも追加でフィルタとして追加したい項目があるとのことで、あわせて追加を要望しています。

具体的な要望について以下に記載します。

- ① コンボボックス外のフィルタとして追加した「社員性別」、「社員登録日」、「社員更新日」をコンボボックス内でフィルタとして選択できるように変更
- ② 次の項目をコンボボックス内で選択できるフィルタとして追加
「社員番号」、「社員生年月日」、「社員趣味」
- ③ 「社員生年月日」は「範囲フィルタ」（2つの日付を入力する形）として追加し、その他は既定のタイプ
- ④ 既定で追加される2つを含む8つの選択肢は次の記載順で表示
「社員名称」、「会社名」、「社員番号」、「社員生年月日」、「社員性別」、「社員趣味」、
「社員登録日」、「社員更新日」

要望を満たせたら一度実行し、生成された Web アプリケーションをエンドユーザーに確認してもらいましょう。
（GeneXus で実行し、生成された画面を確認します）

※動的なフィルタの変更補足

コンボボックス内で表示される項目の表示順は階層構造での表示順に基づいています。

そのため、表示順を変更したい場合には、階層構造において Ctrl + 矢印キー上/下にて並び替えを行うことで対応が可能です。

また、範囲フィルタが追加できない場合、直前の課題の「補足」を参照し、2つの日付が入力できる範囲フィルタに設定する場合、[Picker] プロパティが [Separate dates] とする必要がある。

2.2-5. 固定表示のフィルタ追加

エンドユーザーが社員（Person）の一覧画面で、フィルタとして「社員フルネーム」、「会社名」の利用頻度が高いため、毎回コンボボックスで選択することや、両方をフィルタとして利用するための検索対象を増やす操作を省略し、常にフィルタとして利用できる状態にしたいという要望があります。

ただし、初めてフィルタの追加を要望したように見栄えは崩したくないという要望もあります。

現時点で受けている要望について改めて以下に記載します。

- ① 「社員フルネーム」、「会社名」を常に表示される固定されたフィルタに設定する
- ② 見栄えは崩さず、現在のコンボボックスで選択できるフィルタと同じような見栄えにする

要望を満たせたら一度実行し、生成された Web アプリケーションをエンドユーザーに確認してもらいましょう。

（GeneXus で実行し、生成された画面を確認します）

2.2-6. タイトルフィルタのカスタマイズ

エンドユーザーが社員（Person）の一覧画面で、列のタイトルにもフィルタ機能があることを確認しました。
このタイトルフィルタについて、「社員趣味」については表示される候補について検索できる必要はないため、非表示にしてほしいという要望があります。

その他のタイトルフィルタについては既定のままで問題ありません。

現時点で受けている要望について改めて以下に記載します。

- ① 「社員趣味」のタイトルフィルタで検索ボックスを含めない
- ② 他の列で利用できるタイトルフィルタは既定のまま

要望を満たせたら一度実行し、生成された Web アプリケーションをエンドユーザーに確認してもらいましょう。
（GeneXus で実行し、生成された画面を確認します）

2.2-7. 参照画面を呼び出すアクションの追加

エンドユーザーが社員（Person）の一覧画面で、特定の行に表示されたデータを参照するための画面（次の章で扱う機能）を呼び出せる機能があると聞き、これを追加することを要望しています。

この画面を呼び出すために、更新や削除を行うためのアイコンと同じように一覧内にアイコンを追加してほしいと要望しています。

追加する位置は一番左を要望しています。

現時点で受けている要望について改めて以下に記載します。

- ① データ参照画面を呼び出すためのアイコンの追加
- ② アイコンは一覧内の一番左に追加

要望を満たせたら一度実行し、生成された Web アプリケーションをエンドユーザーに確認してもらいましょう。
（GeneXus で実行し、生成された画面を確認します）

※参照画面を呼び出すアクションの追加補足

参照画面を呼び出すスタンダードアクションは「Display」を利用できます。

2.2-8. 削除を実行するアクションの追加

エンドユーザーが会社（Company）の一覧画面で、削除画面を呼び出さずに削除したいと要望しています。ただし、対象の会社を参照する社員がいる場合にはもちろん削除できないようにし、削除ができた場合も削除できない場合も処理が終わった時点でメッセージの表示を要望しています。

具体的な要望について以下に記載します。

- ① 会社（Company）の削除画面を呼び出さずにデータ削除を実行
- ② 現在の削除画面を呼び出す機能は不要
- ③ 今回の対応で追加する機能のアイコンは削除画面を呼び出すアイコンと同一となることを要望
- ④ アクションを実行しても参照する社員がいる場合、削除できない
- ⑤ 処理が完了した際には、結果に依存せずにメッセージを表示
- ⑥ 削除後の一覧のリフレッシュは不要

要望を満たせたら一度実行し、生成された Web アプリケーションをエンドユーザーに確認してもらいましょう。
（GeneXus で実行し、生成された画面を確認します）

※削除を実行するアクションの追加補足

・アイコンの設定

スタンダードアクションの Delete と同様のアイコンでアクションを追加する場合、ユーザーアクションに対し以下のようにプロパティを設定します。

[Control type] : [Image]

[Image Class] : [fa fa-times]

[Image Type] : [Font icon]

・削除処理

ここで実行する削除処理はあらかじめ用意されている [DeleteCompany] プロシージャを呼び出すことで実装できます。そのためには [CompanyId] 項目属性をパラメーターとして受け渡す必要があります。

・削除後の一覧

削除できるレコードであった場合、メッセージでは削除したことが表示されますが、一覧には表示されたままになります。

これは、削除処理を実行しても一覧のリフレッシュ（再読み込み）が自動でかからないためです。

2.2-9. 休暇期間を延長するアクションの追加

エンドユーザーが社員休暇期間（PersonHolidays）の一覧画面で、休暇期間内のレコードの場合、終了日を 1 日延長できる機能を追加したいと要望しています。

また、処理が終わった場合、画面上で延長されていることを確認できることも要望しています。

具体的な要望について以下に記載します。

- ① 社員休暇期間（PersonHolidays）で社員休暇期間終了日を 1 日加算する処理を実行
- ② 修正対象レコードは休暇期間内のもののみ。
- ③ アクションは一覧内に 3 つ目のアイコンとして表示
- ④ 処理完了後の一覧のリフレッシュが必要

要望を満たせたら一度実行し、生成された Web アプリケーションをエンドユーザーに確認してもらいましょう。

（GeneXus で実行し、生成された画面を確認します）

※休暇期間を延長するアクションの追加補足

・アイコンの設定

今回追加するユーザーアクションのアイコンは以下のようにプロパティを設定します。

[Control type] : [Image]

[Image Type] : [Font icon]

[Font icon] : [far fa-calendar-plus]

・日数追加処理

実行する処理はあらかじめ用意されている [AddHolidays] プロシーチャーを呼び出すことで実装できます。

そのためには、[PersonHolidaysId] 項目属性をパラメーターとして受け渡す必要があります。

・一覧のリフレッシュ

処理の最後一覧をリフレッシュするためには、「Grid.Refresh()」と記述する必要があります。

2.2-10. 1 列で表示する会社情報の追加

エンドユーザーから社員（Person）の一覧画面に対しさらに追加の要件がありました。

社員が所属する会社情報について「会社名」が既定で表示されるようになっているが、「会社番号」も同時に確認できるようにしたいと要望しています。

ただし、「会社番号」と「会社名」を表示するために 2 列必要となる点を改善したく、1 列で「会社番号」と「会社名」を表示してほしいという要望です。

また、追加した列の内容は、既定で実装されているエクセル出力や PDF 出力でも同じ出力を含むことを要望しています。

具体的な要望について以下に記載します。

- ① 「会社番号」と「会社名」の内容を含む「会社情報」という列を追加
- ② 「会社名」は既定で非表示にし、「会社情報」を既定で表示
- ③ 既定で実装されているエクセル出力や PDF 出力でも同じ出力を含む
- ④ 項目属性として追加することは考えていない

要望を満たせたら一度実行し、生成された Web アプリケーションをエンドユーザーに確認してもらいましょう。
(GeneXus で実行し、生成された画面を確認します)

※1 列で表示する会社情報の追加補足

グリッドに変数を追加し、[Load Code] プロパティを利用することでエクセル出力や PDF 出力に自動的に変数へ値を格納し、表示内容に含むように WorkWithPlus for Web では機能が備わっています。

「会社番号」と「会社名」を 1 列で表示したい場合、例えば追加した変数のノードが「CompanyInfo」の場合、以下のような記載で実装することができる。

```
&CompanyInfo = CompanyId.ToString() + " " + CompanyName
```



WORKWITHPLUS
FOR WEB

WorkWithPlus for Web トレーニング

2 – トランザクションオブジェクト

への適用

2.3 – View

2.3 – View

2.3-1. 「追加情報」タブの追加

エンドユーザーが社員（Person）の登録済レコードを表示する画面を確認し、追加の要望がありました。現在、この画面を参照すると、「General」というタブで登録画面における「詳細情報」タブの内容が表示されていますが、「追加情報」として別タブに表示している項目が表示されていないことを確認しました。これらの項目も参照したいため、表示するタブの追加を要望しています。

具体的な要望について以下に記載します。

- ① 登録画面（Transaction）の「追加情報」に含まれる下記一部項目を表示するタブの追加
「社員お気に入りバンド」、「社員趣味」、「社員住所」、「社員写真」、「会社番号」、「会社名」、「社員登録日」、「社員登録者」、「社員更新日」、「社員更新者」
- ② デザインについては、「追加情報」と同様の表示
- ③ 新しいタブは「General」と「社員メール」の間に追加

要望を満たせたら一度実行し、生成された Web アプリケーションをエンドユーザーに確認してもらいましょう。（GeneXus で実行し、生成された画面を確認します）

2.3-2. 「監査情報」パネルの追加

エンドユーザーが会社（Company）の登録済レコードを表示する画面を確認し、追加の要望がありました。現在、この画面を参照すると、「General」というタブに「会社登録日」、「会社登録者」、「会社更新日」、「会社更新者」が表示されていないことを確認しました。これらの監査用項目も参照できるべきであるため、「General」タブへタイトル付きのテーブル（パネル）を追加し、監査用項目を表示する変更を要望しています。

具体的な要望について以下に記載します。

- ① 「General」タブにタイトル付きのテーブル（パネル）を追加
- ② 追加したタイトル付きのテーブル（パネル）に下記 4 項目を 1 行 2 列で表示
「会社登録日」、「会社登録者」、「会社更新日」、「会社更新者」

要望を満たせたら一度実行し、生成された Web アプリケーションをエンドユーザーに確認してもらいましょう。
（GeneXus で実行し、生成された画面を確認します）

2.3-3. Autolink の無効化

エンドユーザーが社員（Person）の一覧画面で、登録済レコードを表示する画面の呼び出しに [Display] スタンダードアクションを利用しているため、「社員フルネーム」がリンクである必要はありません。
この社員の一覧画面で「社員フルネーム」のリンクを無効にするように要望しています。

現時点で受けている要望について改めて以下に記載します。

- ① 「社員フルネーム」のリンク無効化
- ② 対象画面は社員（Person）の一覧画面のみ

要望を満たせたら一度実行し、生成された Web アプリケーションをエンドユーザーに確認してもらいましょう。
（GeneXus で実行し、生成された画面を確認します）



WORKWITHPLUS
FOR WEB

WorkWithPlus for Web トレーニング

3 – WorkWithPlus for Web の設定

3.1 – 既定値の設定

3.1 – 既定値の設定

3.1-1. 既定の演算子の利用

エンドユーザーが社員（Person）の一覧画面に対して追加の変更要望があります。

フィルタとして利用する「社員登録日」について選択できる演算子を「指定日付」、「範囲」、「過去」、「今月」の4つに変更したいと要望しています。

この時、「範囲」は開始と終了の日付を入力することを要望しています。

現時点で受けている要望について改めて以下に記載します。

- ① 「社員登録日」の動的な演算子として次の4つとなるように変更
「指定日付」、「範囲」、「過去」、「今月」
- ② 追加する動的な演算子「範囲」については、「範囲ピッカー」ではなく、従来通りの開始と終了 2 つの日付を入力する

要望を満たせたら一度実行し、生成された Web アプリケーションをエンドユーザーに確認してもらいましょう。

（GeneXus で実行し、生成された画面を確認します）

3.1-2. 全画面を対象にしたタイトルフィルタのカスタマイズ

エンドユーザーがすべての一覧画面で列のタイトルフィルタに対する変更要望があります。

Web アプリケーション内すべての日時型項目（DateTime 型）のタイトルフィルタに表示される検索ボックスを開始と終了の 2 つの日時を入力できるように変更することを要望しています。

この変更は今後作成される一覧画面にも適用される変更となります。

現時点で受けている要望について改めて以下に記載します。

- ① 生成されるすべての一覧画面に対するタイトルフィルタの変更
- ② 対象はデータタイプが DateTime 型の項目属性のみ
- ③ 検索ボックスの表示を開始と終了の 2 つの入力欄（＝範囲検索）への変更
- ④ 今後作成される一覧画面にも変更内容を適用

要望を満たせたら一度実行し、生成された Web アプリケーションをエンドユーザーに確認してもらいましょう。

（GeneXus で実行し、生成された画面を確認します）

※全画面を対象にしたタイトルフィルタのカスタマイズ補足

今回の課題では、「WorkWithPlus 設定」のカスタマイズを行う必要があります。

変更を保存した際に表示される「すべての WorkWithPlus パターンインスタンスを適用」ダイアログでは、「はい」をクリックします。

この点については、別途説明を実施しています。



WORKWITHPLUS
FOR WEB

WorkWithPlus for Web トレーニング

3 – WorkWithPlus for Web の設定

3.2 – テンプレート

3.2 – テンプレート

3.2- 1. 自動ルールの追加

エンドユーザーが監査用項目に値を自動入力していないことを思い出しました。
そのため、監査用項目に値を自動入力する変更を要望しています。
この変更については、現在の定義範囲だけでなく、今後のことも見据えた変更を希望しています。
監査用項目の名前については命名規則に基づき一定の名前となることを確認済みです。

具体的な要望について以下に記載します。

- ① Web アプリケーションがもつすべての監査用項目に値を自動で代入
- ② 対象は項目属性名に必ず下記いずれかの文字列を含む（命名規則により）
CreateDate、CreateUser、UpdateDate、UpdateUser
- ③ 日時型の項目には現在の日時、文字列型には「ゲストユーザー」を代入
（セキュリティ機能を有効化していないため、一時的なユーザー名を指定）
- ④ 値の自動入力はデータの新規登録や更新処理の直前に実施
- ⑤ 今後追加される項目属性にも自動で代入を適用

要望を満たせたら一度実行し、生成された Web アプリケーションをエンドユーザーに確認してもらいましょう。
（GeneXus で実行し、生成された画面を確認します）

※自動ルールの追加補足

・ルールの定義

ルールを定義する際、WorkWithPlus for Web では、あらかじめ用意された自動置換されるタグがあり、
今回のように対象の項目属性名を利用したい場合、「<ATT_NAME>」と記述することで対応利用可能です。

・現在の日時取得

現在の日時を取得する場合、「ServerNow()」関数を利用できます。

・自動入力のタイミング指定

GeneXus では、ルールの実行を任意のタイミングにすることができるトリガーイベントがありました。

今回の場合、新規登録時に実行するルールについては、「on BeforeInsert」を指定し、更新時に実行する
ルールについては、「on BeforeUpdate」を指定することで要望を満たせます。

・インスタンスの更新

変更を保存した際に表示される「インスタンスの更新」ダイアログでは、「はい」をクリックします。
この点については、別途説明を実施しています。

・社員（Person）の更新

社員（Person）にルールは追加されたでしょうか？

もし、追加されていない場合、社員（Person）のレイアウトをカスタマイズ（[2.1-2](#)）した際の
操作を確認しましょう。

3.2-2. 実行モードの表示

エンドユーザーがすべての登録画面に対する変更要望があります。

現在、登録画面が表示されていても、データの新規登録か、更新か、削除かは明確に確認できません。

そのため、これらの情報が画面上に表示されるように変更することを希望しています。

この変更は今後作成される登録画面にも適用される変更となります。

具体的な要望について以下に記載します。

- ① 「新規登録」、「更新」、「削除」という文字列を実行しているモードに合わせて表示
- ② 表示位置については、フォームのタイトルに書き足す形で表示を希望
- ③ 今後作成される登録画面にも変更内容を適用

要望を満たせたら一度実行し、生成された Web アプリケーションをエンドユーザーに確認してもらいましょう。

(GeneXus で実行し、生成された画面を確認します)

※実行モードの表示補足

・フォームのタイトル編集コード

フォームのタイトルに文字列を書き足したい場合、下記のイベントコードを「Start」イベントで実行することで実装できます。

Do case

Case &Mode = TrnMode.Insert

Form.Caption += " (新規登録) "

Case &Mode = TrnMode.Update

Form.Caption += " (更新) "

Case &Mode = TrnMode.Delete

Form.Caption += " (削除) "

EndCase

・社員 (Person) の更新

直前の課題同様に社員 (Person) にイベントは追加されたでしょうか？



WORKWITHPLUS
FOR WEB

WorkWithPlus for Web トレーニング

3 – WorkWithPlus for Web の設定

3.3 – インスタンスの更新 / 適用

3.3 – インスタンスの更新 / 適用

3.3-1. 全登録画面のレイアウト変更

エンドユーザーがすべての登録画面で、ボタンの表示位置について変更要望があります。現在は画面の一番下に表示されていますが、画面の一番上に表示したいと要望しています。この変更は今後作成される登録画面にも適用される変更となります。

具体的な要望について以下に記載します。

- ① 「実行」や「終了」ボタンなどあらかじめ用意されているボタンの位置の変更
- ② 移動先は「詳細情報」タイトル付きのテーブル（パネル）の上
- ③ 今後作成される登録画面にも変更内容を適用

要望を満たせたら一度実行し、生成された Web アプリケーションをエンドユーザーに確認してもらいましょう。
（GeneXus で実行し、生成された画面を確認します）

※全登録画面のレイアウト変更補足

・社員（Person）登録画面

社員（Person）登録画面のボタンは移動したでしょうか？

もし、移動していない場合、社員（Person）登録画面のレイアウトをカスタマイズ（[2.1-2](#)）した際の操作を確認しましょう。

・会社（Company）登録画面

会社（Company）登録画面のボタンは移動したでしょうか？

もし、移動していない場合、会社（Company）登録画面のレイアウトをカスタマイズ（[2.1-4](#)）した際の操作を確認しましょう。



WORKWITHPLUS



WORKWITHPLUS
FOR WEB

WorkWithPlus for Web トレーニング

4 – Web パネル

4 – Web パネル

4-1. 固定日数の社員休暇期間（PersonHolidays）登録画面

エンドユーザーは、社員休暇期間（PersonHolidays）の登録簡易化の要望があります。

社員が休暇を取る場合、申請の翌日から 1 週間（7 日間）の休暇を取ることが多いため、登録時の操作簡易化のために自動的にこの期間で指定した社員の休暇が登録されるように機能を追加したいと要望しています。

社員については既に登録されているデータから選択できる必要があります。

ただし、現在実装済みのすべて入力可能な画面も維持したい。

具体的な要望について以下に記載します。

- ① 社員休暇期間（PersonHolidays）を登録するための新規画面の作成
- ② 対象社員は、コンボボックス（動的コンボボックス）による選択
- ③ 社員休暇期間理由はこの画面で手入力
- ④ この画面では休暇期間は登録時の日付の翌日（+1 日）から 1 週間後（+8 日）固定
- ⑤ この画面は社員休暇期間（PersonHolidays）の一覧画面に追加するボタンで呼び出す
- ⑥ 一覧画面に追加するボタンはグリッド外の「追加」ボタンに並び、同様のデザインを適用

要望を満たせたら一度実行し、生成された Web アプリケーションをエンドユーザーに確認してもらいましょう。

（GeneXus で実行し、生成された画面を確認します）

※固定日数の社員休暇期間（PersonHolidays）登録画面補足

- ・社員休暇期間（PersonHolidays）登録画面外からの登録
社員休暇期間（PersonHolidays）トランザクションは取り込み時にビジネスコンポーネントとして設定済
- ・社員休暇期間開始日、社員休暇期間終了日の設定
画面上から入力しないため、イベントにて直接コードを追記する必要があります。
[Events] エレメントの [Enter] イベントとして既定の登録コードが生成されているため、この自動生成コード前に日付を指定するコードを追記します。
- ・一覧画面からの呼び出し
呼び出し時には、パラメーターとして実行モードの指定が求められる場合、「TrnMode.Insert」を指定し、PersonHolidaysId の指定が求められる場合、空の値（0）を指定する必要があります。
- ・同様のデザインを適用
既定で追加されている「追加」ボタンなどは [Action Group] ノードに含まれている。ここで追加するユーザーアクションも同じ [Action Group] ノードへ追加することで統一したデザインとすることができる。

4-2. 会社（Company）と社員（Person）の同時登録

エンドユーザーは、会社（Company）の登録に関して要望があります。

一部の会社（Company）を登録する際には、必ず管理職となる社員（Person）を登録できる機能が必要です。この時登録する社員については必要最低限の項目だけ入力できれば問題ありませんが、会社または社員の登録が失敗した場合には、どちらも登録せずに処理を終了させます。

また、会社と社員の入力画面はタブ形式で表示することを要望しています。

具体的な要望について以下に記載します。

- ① 会社（Company）と社員（Person）を同時に登録できる新規画面の作成
- ② 会社（Company）の登録時に入力したい項目は次の項目
会社名、会社詳細
- ③ 社員（Person）の登録時に入力したい項目は次の項目
社員姓、社員名、社員性別、社員生年月日
- ④ 社員（Person）の登録時に内部的に値を割り当てたい項目は次の項目
会社番号、会社管理職
- ⑤ 会社と社員の入力画面はタブ形式で表示
- ⑥ 入力された会社と社員どちらかが登録失敗の場合、どちらも登録しない
- ⑦ この画面は会社（Company）の一覧画面に追加するボタンで呼び出す

要望を満たせたら一度実行し、生成された Web アプリケーションをエンドユーザーに確認してもらいましょう。

（GeneXus で実行し、生成された画面を確認します）

※会社（Company）と社員（Person）の同時登録補足

- ・会社（Company）登録画面外からの登録
会社（Company）トランザクションは取り込み時にビジネスコンポーネントとして設定済
- ・社員（Person）登録画面外からの登録
社員（Person）トランザクションは取り込み時にビジネスコンポーネントとして設定済
- ・会社番号の取得
社員（Person）の会社番号に割り当てる値は、会社（Company）をビジネスコンポーネントとして登録した場合、登録処理後に「&Comp;any.CompanyId」で取得できる。（&Company は Company ビジネスコンポーネント）

4-3. 会社一覧画面の新規作成

エンドユーザーから会社の一覧について新しい要望があります。

会社の一覧に所属する社員数とその中で休暇期間中の社員が何名いるのかを表示したいという要望です。

ただし、これまで利用していた会社の一覧画面はそのまま利用したいため、別の画面として作成を要望しています。

具体的な要望について以下に記載します。

- ① この会社の一覧で表示したい項目は次の通り
会社番号、会社名、社員数、休暇社員数
- ② 既存の会社の一覧は編集せず、新規に作成した一覧画面としたい
- ③ 実装されている機能については既存の会社一覧と同様を希望
(本課題では見た目だけそろえ、実際の機能としての実装は不要です)
- ④ 社員数や休暇社員数についてはこの画面のみで必要となるため、項目として定義する必要はない

要望を満たせたら一度実行し、生成された Web アプリケーションをエンドユーザーに確認してもらいましょう。

(GeneXus で実行し、生成された画面を確認します)

※会社一覧画面の新規作成補足

要望を満たすために利用できる SDT オブジェクトおよびデータプロバイダーオブジェクトはあらかじめナレッジベースに取り込んでいました。それぞれ以下の名称です。

SDT オブジェクト : CompanyListSDT

データプロバイダーオブジェクト : CompanyListDP



WORKWITHPLUS
FOR WEB

WorkWithPlus for Web トレーニング

5 – PDF レポート

5 – PDF レポート

5-1. 社員休暇期間一覧出力

エンドユーザーから選択した社員に関連した休暇期間の一覧を PDF として出力してほしいと要望しています。
この PDF は社内で保管されるため、一定の見栄えを要求しています。
また、出力については社員（Person）の一覧画面で表示される各行にアイコンを追加し、これをクリックすることで出力できることを要望しています。

具体的な要望について以下に記載します。

- ① PDF のヘッダー情報として出力したい項目は次の通りとなり、1 行あたり 2 項目表示したい
社員番号、社員フルネーム、会社名、会社管理職
- ② PDF の明細情報として出力したい項目は次の通り
社員休暇期間理由、社員休暇期間開始日、社員休暇期間終了日
- ③ 明細情報の「社員休暇期間開始日」と「社員休暇期間終了日」の間に「～」を表示
- ④ PDF のフッター情報として出力日を右寄せで出力
- ⑤ PDF は社員（Person）の一覧画面で、各行に追加するアイコンのクリックで表示
- ⑥ 社員（Person）の一覧画面でクリックした行の社員に関連するデータを表示

要望を満たせたら一度実行し、生成された Web アプリケーションをエンドユーザーに確認してもらいましょう。
(GeneXus で実行し、生成された画面を確認します)

※社員休暇期間一覧出力補足

- ・日付の右寄せ出力
[Location in father table] プロパティ群に含まれる [Align] プロパティで指定可能
- ・一覧画面の行へアイコンを追加
[Grid] タイプノードに [UserAction] ノードを追加する必要がある。また、アイコンで表示するためには、WorkWithPlus for Web によってあらかじめ取り込まれている画像 [ActionExportReport] を利用できる。
このアイコンで表示する点においては下記のプロパティを設定することで実装可能
[Control type] : [Image]
[Image] : [ActionExportReport]
[Image Class] : [Image]



WORKWITHPLUS



WORKWITHPLUS
FOR WEB

WorkWithPlus for Web トレーニング

6 - セキュリティ

6 - セキュリティ

6-1. セキュリティ機能の有効化

エンドユーザーが、この Web アプリケーションのすべての機能をだれでも利用できる状況から、権限に基づく機能のみを利用できるように制限をかけたいと要望しています。

この制限は機能（画面）単位だけでなく、項目についても権限を与えたいと要望しています。

具体的な要望について以下に記載します。

- ① セキュリティ機能（GAM）を権限が管理できるレベル（「承認」）で有効化
- ② ユーザーが任意のタイミングでログアウトできるようにログアウト操作の実装
- ③ 次の項目に対する編集権限の付与を可能とする
社員姓、社員名、社員趣味
- ④ 次の項目に対する表示権限の付与を可能とする
社員番号、社員姓、社員名、社員フルネーム、社員ニックネーム、社員性別、社員生年月日、
会社管理職、社員登録日、社員登録者、社員更新日、社員更新者

要望を満たせたら一度実行し、生成された Web アプリケーションをエンドユーザーに確認してもらいましょう。
（GeneXus で実行し、生成された画面を確認します）

※セキュリティ機能の有効化補足

- ・「デザイン システム ウィザード」の再実行
GAM に基づくセキュリティ機能を実装する場合、「デザイン システム ウィザード」を実行します。
実行時に表示されるドキュメントを開くかの確認ダイアログは「いいえ」のクリックで問題ありません。
- ・ベースのデザインシステム選択
「デザイン システム ウィザード」の初期画面（ステップ 1）となる「ベースのデザインシステムを選択」では、「使用中のデザインシステム」を選択します。
- ・ログアウト操作の追加
ステップ 5 で「ログアウトアクション」というチェックボックスにチェックすることで追加できます。
- ・再インポート方法
「デザイン システム ウィザード」で設定を完了し、「デザインシステムをインポート」をクリックすると、インポート方法について確認するダイアログが表示されます。
このダイアログでは、既定値の「必要なオブジェクトのみをインポート」のままインポートをクリックします。
- ・GAM 有効化後の実行
「デザイン システム ウィザード」による再インポート完了後、「すべてリビルド」を実施する必要があります。
- ・GAM の初期ユーザー
GAM によって下記の通り初期ユーザーが作成されています。
メールアドレス（ユーザーID） : admin
パスワード : admin123

6-2. 監査用項目への自動割り当て値更新

エンドユーザーが、権限に基づく利用制限をかけることによって利用者の「ユーザー名」が取得可能となったことから、以前実装していた監査用項目の「登録者」および「更新者」に自動入力する値を変更することを要望しています。

現時点で受けている要望について改めて以下に記載します。

- ① 実装済みの自動入力処理 ([3.2-1](#)) を変更
- ② 監査用項目のうち、「登録者」、「更新者」へ割り当てる値の変更
- ③ 割り当てる値はログインしたユーザーの「ユーザー名」

要望を満たせたら一度実行し、生成された Web アプリケーションをエンドユーザーに確認してもらいましょう。
(GeneXus で実行し、生成された画面を確認します)

※監査用項目への自動割り当て値更新補足

GAM の機能を利用し、ログインしたユーザー名を取得したい場合、「GAMUser.GetName()」と記載することで取得可能。

6-3. ユーザーの追加

エンドユーザーからユーザーの作成方法がわからないと連絡を受け、登録を代行することになりました。
2種類のロールを作成し、それぞれのロールを紐づけるユーザーの作成を代行します。
(2つのロール、2つのユーザー作成)

登録する内容は以下の通りです。(指定内容以外は既定値のまま)

・ロール 1

- ① 名前は「Visitor」
- ② 紐づける権限（アクセス許可）は以下の通り
 - company_FullControl
 - companyww_Execute
 - gamhome_Execute
 - home_Execute
 - person_Execute
 - person_Update
 - personholidays_FullControl
 - personholidaysww_Execute
 - personww_Execute

・ロール 2

- ① 名前は「LowPerm」
- ③ 紐づける権限（アクセス許可）は以下の通り（太字は「ロール 1」と重複）
 - **company_FullControl**
 - **companyww_Execute**
 - **gamhome_Execute**
 - **home_Execute**
 - **person_Execute**
 - **person_Update**
 - **personholidays_FullControl**
 - **personholidaysww_Execute**
 - **personww_Execute**
 - PersonCreateDate_Attribute
 - PersonFirstName_Attribute
 - PersonHobby_Edit_Attribute
 - PersonLastName_Attribute

(次ページへ続く)

・ユーザー1

- ① ユーザー名は「VisitorUser」
- ② メールアドレスは「visitoruser@test.jp」
- ③ パスワード、パスワード設定は「vu123」
- ④ 紐づけるロールは「Visitor」

・ユーザー2

- ① ユーザー名は「TestUser」
- ② メールアドレスは「testuser@test.jp」
- ③ パスワード、パスワード設定は「tu123」
- ④ 紐づけるロールは「LowPerm」

要望を満たせたら各ユーザーでアプリケーションにログインしてもらい、ユーザーの要望している制御が行われていることをエンドユーザーに確認してもらいましょう。

（各ユーザーでログインし、正しい制御となっていることを確認します）

※ユーザーの追加補足

・ロールへの権限（アクセス許可）紐づけ

権限（アクセス許可）を紐づける場合、必ず「アプリケーション」フィルタを指定する必要があります。
Web アプリケーションに関する権限（アクセス許可）を追加する場合、ナレッジベース名と同じ値（既定値）を選択します。

・ロールのコピー

似た設定内容のロールを作成する場合、一覧画面に用意された「コピー」というアクションを活用できます。
コピーして作成されたロールを必要に応じて修正します。

・ユーザーのメールアドレス

ユーザー追加時に指定するメールアドレスはメールフォーマット（xx@yy.zz）にのっとっていない場合、登録することができません。これは GeneXus による制限です。

・ユーザーのパスワード

ユーザー追加時に指定する「パスワード」と「パスワード設定」については、1 度目の入力と、入力内容の確認用の 2 度目の入力欄です。そのため、同じ値を入力する必要があります。



WORKWITHPLUS
FOR WEB

WorkWithPlus for Web トレーニング

7 - 監査

7 - 監査

7-1. 監査機能の有効化

エンドユーザーが、現在の監査用項目では、下記のケースに対応できないことに気が付きました。

- ・新規登録時に入力されていたデータの内容
- ・データが複数回更新された際の各更新日時および内容
- ・データが削除されたことの記録

現在の監査項目はそのまま一般ユーザーに参照できるように残しておき、上記のような監査用データを管理者が参照できるように残したいと要望しています。

対象はこの Web アプリケーションでデータを管理しているテーブルすべてとしたいとのことです。

具体的な要望について以下に記載します。

- ① Web アプリケーションに監査項目とは別に監査用データを残せる機能を追加
- ② すべてのテーブルを対象
- ③ 今後追加されるテーブルも自動で対象とする
- ④ 記録される監査用データには、ログインしているユーザーがわかるようにする
- ⑤ 記録する操作はこの Web アプリケーションから画面を通した操作のみを対象とする

要望を満たせたら一度実行し、生成された Web アプリケーションをエンドユーザーに確認してもらいましょう。
(GeneXus で実行し、生成された画面を確認します)

※監査機能の有効化補足

- ・「Audit」トランザクションオブジェクトの定義
既定で用意されている「AuditTransaction」プロシージャオブジェクトに記載の手順では、利用しているセキュリティタイプに基づき「Audit」トランザクションオブジェクトに定義する項目を変更するように指示が記載されている。ただし、「Audit」トランザクションオブジェクトは既にインポート済みです。
- ・「Audit」トランザクションオブジェクト自身の監査
もし、「Audit」トランザクションオブジェクトに WorkWithPlus for Web を適用する場合、
「Audit」トランザクションオブジェクト自身に対する監査機能は無効化する必要がある。
- ・「AuditTransaction」プロシージャオブジェクトのコード
前述の通りセキュリティタイプに基づく定義の差分について記載があるが、実際に値を代入するコードは各セキュリティタイプに対応した記載がない。
今回はあらかじめ用意した「Audit」トランザクションオブジェクトを利用するため、Source エlement 39 行目は以下とする必要がある。
`&Audit.AuditUserName = GAMUser.GetName()`