



GeneXus BPM Suite

Custom Client part 2

Version: GeneXus™ 17

Instructor | GeneXus Training

VOICE: Carolina Torrado

カスタムクライアント

- GeneXus の WebPanelとプロシージャーを使用して開発
- Gxflow API を使用してエンジンに接続
- 通常の Gxflow クライアントのすべての機能が含まれています
- カスタマイズ可能（例::
 1. ユーザーの受信トレイ グリッドまたはその他のページに新しい列を追加または非表示にする
 2. Gxflow コード間にアクションまたはカスタムコードを追加する
 3. UX を変更する
 4. カスタムクライアントの一部またはすべてをアプリケーションに統合する

<http://wiki.genexus.jp/hwikibypageid.aspx?11364>

GXflow カスタムクライアントは GeneXus によって提供され、ナレッジベースにインポートしてプロジェクトの一部として使用または変更することができます。

このコースの前半で説明した GXflow 標準クライアントの多くの機能を実装するために必要なすべての WebPanelとプロシージャが含まれています。

GXflowカスタムクライアントを使用すると、GXflowにログインし、ユーザーの受信トレイを実行し、送信トレイを確認し、プロセス定義、作業項目、インスタンスを管理し、実行を監視することもできます。

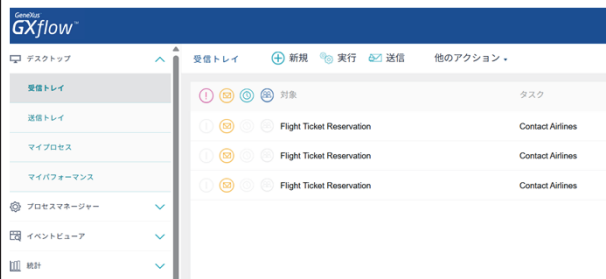
これらのWebPanelにプログラムされたソースコードを見ると、GXflow APIを使用してGXflowエンジンに接続し、それぞれのケースで必要な情報を取得し、各オブジェクトに対して異なる関数を実行する方法がわかります。

この章では、カスタムクライアントのWebPanelをいくつか見て、そのコードとその変更方法を理解します。

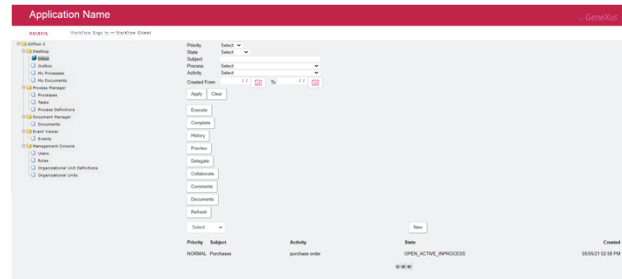
カスタムクライアントは様々な目的で変更できます。
例えば、情報の追加や非表示、カスタムクライアントを
様々なパネルに統合、様々なパネルにアクションを追加し
てコードを埋め込むなどです。このウェブパネルのUXは
すべて、プロジェクトに合わせて変更・調整可能です。

カスタムクライアントを使用して、GXflowのユーザーイ
ンターフェースを使用せずに、独自のUXインターフェー
スを使用して、プロジェクトの一部としてGXflowのクラ
イアントをプロジェクトに統合します。

カスタムクライアント



標準クライアント



カスタムクライアント

カスタムクライアントはデスクトップフォルダ（受信トレイ、送信トレイ、マイプロセス、マイドキュメント）に含まれており、標準クライアントと同じ情報とアクションを提供します。

プロセスマネージャフォルダには、プロセスマネージャセクションからよく使用されるプロセスページ、タスク、プロセス定義が含まれています。

ドキュメントマネージャフォルダには、標準クライアントと同じドキュメント管理ページが含まれています。

イベントビューアフォルダには、標準クライアントと同じイベントマネージャページが含まれています。

管理コンソールフォルダには、標準クライアントと同じユーザー管理、ロール管理、組織単位定義、組織単位ページが含まれています。

統計、バックエンド、サーバー設定、ライセンスマネージャは、カスタムクライアントサンプルには実装されていません。

- WorkflowClient
 - WorkflowDialogs
 - WorkFlowAssign
 - ユーザーに作業項目を割り当てるためのWebPanel
 - WorkFlowComments
 - プロセスインスタンスまたは作業項目にコメントを入力するためのWebPanel
 - WorkFlowDocumentVersions
 - ドキュメントのバージョンを管理できるWebPanel
 - WorkFlowNewDocument
 - 新しいドキュメントを作成するためのWebPanel
 - WorkFlowOrganizationalUnitDefinitionDialog
 - 新しい組織単位定義を追加するためのWebPanel
 - WorkFlowOrganizationalUnitDialog
 - 新しい組織単位を追加するためのWebPanel
 - WorkFlowOrganizationalUnitMembers
 - 組織単位のユーザーを管理できるWebPanel
 - WorkFlowRenameDocument
 - 現在のプロセスインスタンスに関連付けられているドキュメントの名前を変更できるWebPanel
 - WorkFlowRoleDialog
 - 新しいロールを作成するためのWebPanel
 - WorkFlowRoleUsers
 - 1つのロールのユーザーを管理できるWebPanel
 - WorkFlowSelectActivity
 - アクティビティを選択するためのWebPanel
 - WorkFlowUploadDocument
 - ファイルをアップロードするためのWebPanel
 - WorkFlowUserDialog
 - ユーザーを追加/削除/変更するための WebPanel
 - WorkFlowUserOrganizationalUnits
 - ユーザーの組織単位を管理できるWebPanel
 - WorkFlowUserRoles
 - ユーザーのロールを管理できるWebPanel
 - WorkFlowWorkWithDocuments
 - 作業項目のドキュメントを管理できるWebPanel

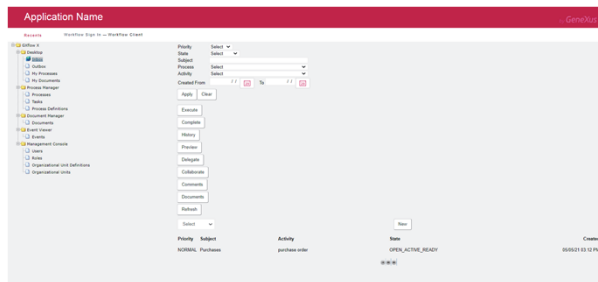
WorkflowEntryPoints
WorkflowDocuments
ドキュメントを管理できるWebコンポーネント
WorkflowEvents
イベント履歴を表示できるWebコンポーネント
WorkflowHistory
プロセス履歴を表示できるWebコンポーネント
WorkflowInbox
ユーザーが実行する必要があるタスクを管理できる
Webコンポーネント
WorkflowOrganizationalUnitDefinitions
組織単位の定義を管理できるWebコンポーネント
WorkflowOrganizationalUnits
組織単位を管理できるWebコンポーネント
WorkflowProcessDefinitions
プロセス定義を管理できるWebコンポーネント
WorkflowProcesses
プロセスを管理できるWebコンポーネント
WorkflowRoles
ロールを管理できるWebコンポーネント
WorkflowSettings
設定を管理できる Web コンポーネント
WorkflowTasks
タスクを管理できる Web コンポーネント
WorkflowUsers
ユーザーを管理できる Web コンポーネント

WorkflowHelpers
WorkflowBuildApplicationUrl
アプリケーションURLを構築するプロセス
WorkflowDocumentDownloader
ドキュメントをダウンロードできるプロセス
WorkflowFilters
グリッドでフィルターを使用できる
Webコンポーネント
WorkflowPaging
グリッドでページングを使用できる
Web コンポーネント

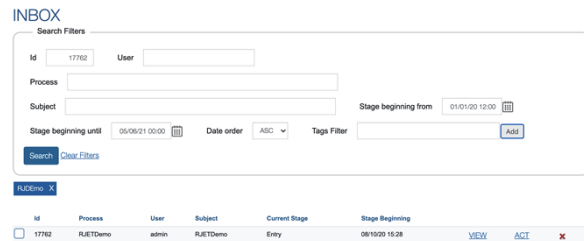
WorkflowClient
クライアントを一元管理する WebPanel.

WorkflowSignIn
ワークフロークライアントにログインできる
WebPanel.

カスタムクライアント



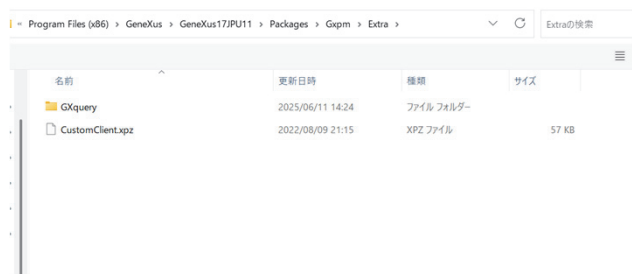
カスタムクライアント既定の受信トレイ




カスタマイズされた受信トレイ

カスタムクライアントの主な用途は、同じUXを使用してGXflowページをプロジェクトに組み込むことです。例えば、このページは、アプリケーションの他の部分と一致するルック&フィールと動作でカスタマイズされた受信トレイを表示します。アプリケーションのユーザーは、アプリケーションの他の部分との違いを感じることはありません。

カスタムクライアント



1. CustomClient.xpzはGeneXusインストールフォルダの「packages/Gxpm/extra」フォルダ内にあります。

 インポートする前に、このナレッジベースにビジネスプロセスダイアグラムが作成されていることを確認してください。

CustomClient.xpz というファイルがあり、GeneXusのインストールフォルダの次の配下に配布されています：

<GeneXusインストールディレクトリ>

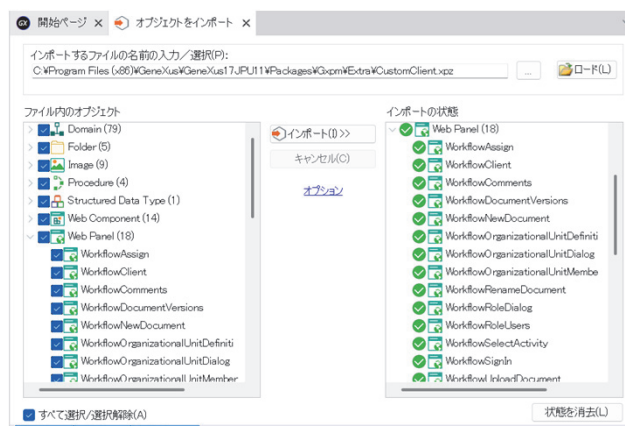
¥Packages¥GXPM¥Extra

インポートする前に、このナレッジベースに少なくとも1つのビジネスプロセスダイアグラムが作成されていることを確認してください。

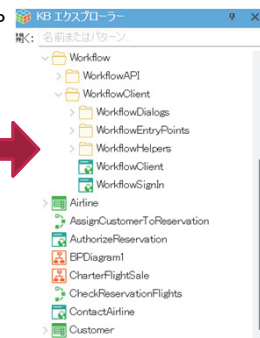
これは、モデルにワークフローデータ型が含まれていることを確認するために必要です。

そうでない場合、プロジェクトのコンパイル時にエラーが発生します。

オブジェクトのインポート - カスタムクライアント

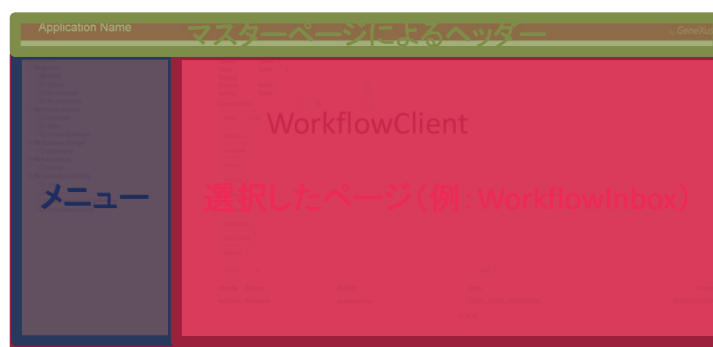


- ナレッジマネージャのインポート機能を使用して xpz をインポートします。



ワークフローカスタムクライアント – ワークフロークライアント

GXflowカスタムクライアントの主なオブジェクトは、WorkflowClientと呼ばれるWebPanelです



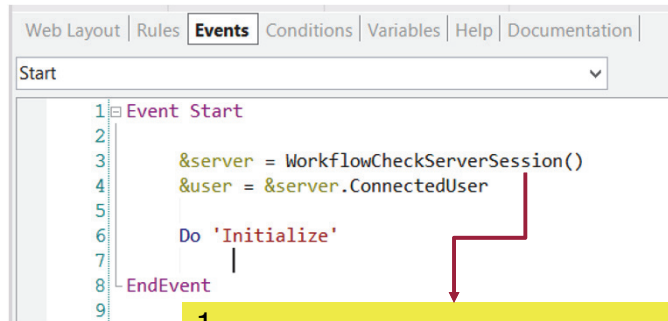
Workflow Inbox – 実行画面およびレイアウト

The screenshot displays the 'Workflow Sign In - Workflow Client' interface. On the left is a sidebar menu with categories like Desktop, Outbox, My Processes, My Documents, Process Manager, Process Definitions, Task, Document Manager, Documents, Event Viewer, Management Console, Users, Roles, Organizational Unit Definitions, and Organizational Units. The main area contains filters for Priority, State, Subject, Process, and Activity, along with 'Created From' and 'To' fields. Action buttons include Execute, Complete, History, Preview, Delegate, Collaborate, Comments, Documents, and Refresh. A table at the bottom shows workflow items with columns for Priority, WfWorkitem, Subject, Activity, State, and Created. The right-hand panel shows a component view with a table of workflow items and a 'New' button.

Priority	WfWorkitem	Subject	Activity	State	Created
NORMAL	Purchases		purchase order	OPEN_AC	

このオブジェクトにあらかじめ定義されているイベントについて取り上げていきます。

セッションの検証



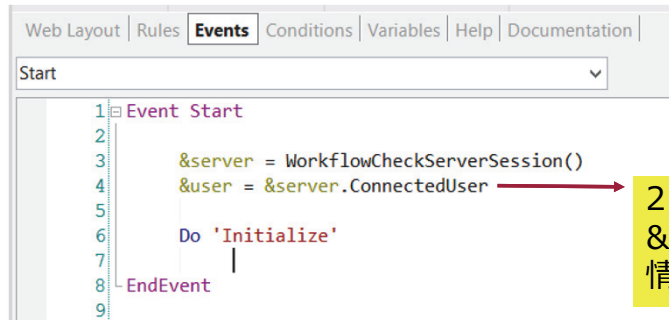
- すべてのカスタムクライアントオブジェクトは、Startイベントで、workflow sessionの検証を行う
- &server 変数 :
WorkflowServer データタイプ
- &user 変数 :
WorkflowUser データタイプ

1.
WorkflowCheckServerSession
プロシージャーを利用し、Web
セッションから&Server変数を
読み込む

カスタムクライアントとして取り込んだすべてのサンプルオブジェクトは、有効なセッションがあるかどうかを確認します。

そのためには、WorkflowCheckServerSessionと呼ばれるプロシージャーを使用します。

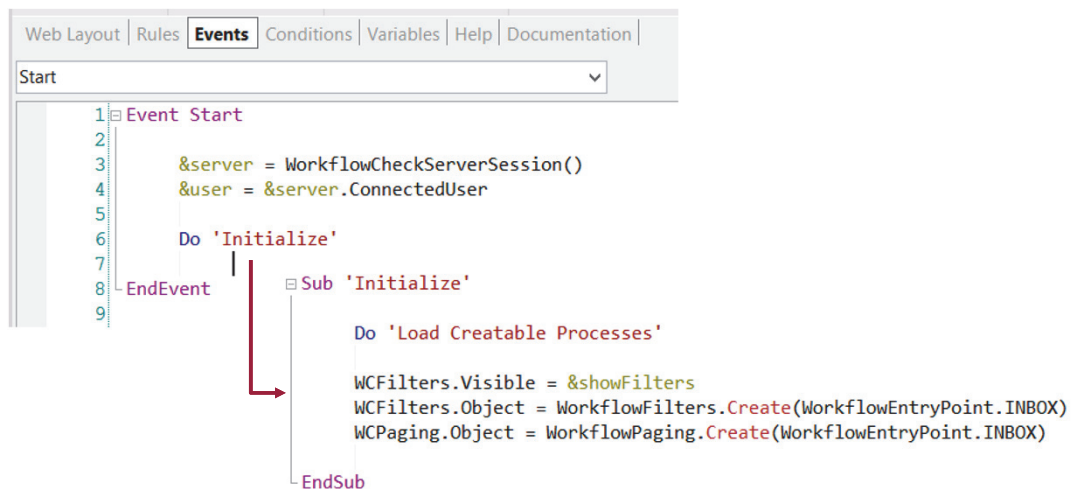
セッションの検証



```
1 Event Start
2
3     &server = WorkflowCheckServerSession()
4     &user = &server.ConnectedUser
5
6     Do 'Initialize'
7         |
8 EndEvent
9
```

2.
&server の接続情報からユーザー
情報を読み込み

セッションの検証



The screenshot shows the GeneXus development environment with the 'Events' tab selected. The 'Start' event is defined with the following code:

```
1 Event Start
2
3   &server = WorkflowCheckServerSession()
4   &user = &server.ConnectedUser
5
6   Do 'Initialize'
7   |
8 -EndEvent
9
```

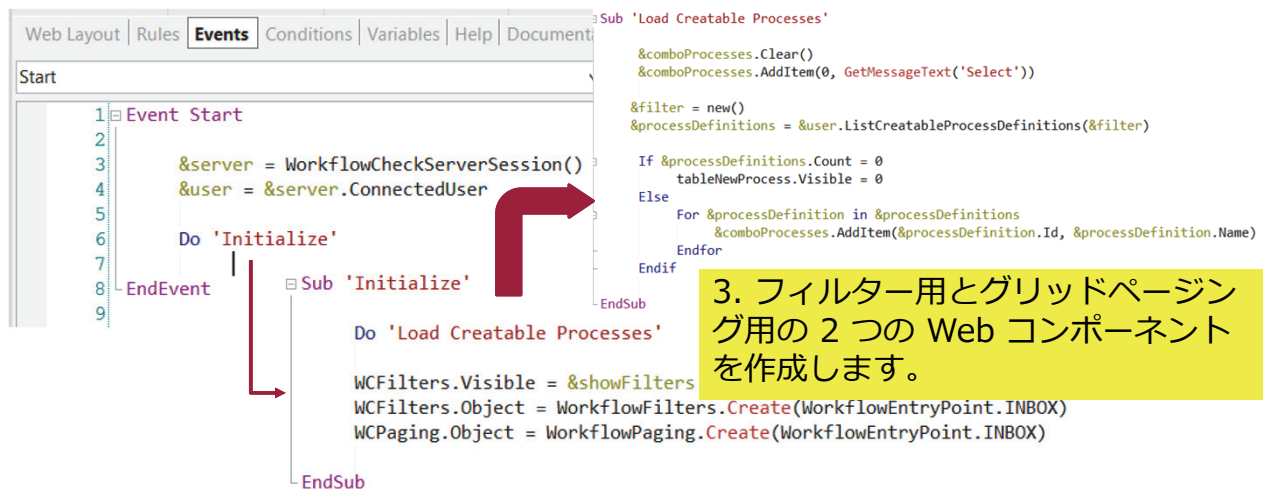
The 'Initialize' sub-routine is defined as follows:

```
Sub 'Initialize'
  Do 'Load Creatable Processes'
  WCFilters.Visible = &showFilters
  WCFilters.Object = WorkflowFilters.Create(WorkflowEntryPoint.INBOX)
  WCPaging.Object = WorkflowPaging.Create(WorkflowEntryPoint.INBOX)
EndSub
```

A red arrow points from the 'Do 'Initialize'' line in the event code to the 'Sub 'Initialize'' definition.

その後、ユーザーの受信トレイの場合は、Webページを初期化します。「Initialize」サブルーチンを使用します。

セッションの検証



プロセス定義のコンボボックスの内容をロードするには：

- 1)まず、ユーザーが作成する権限を持つプロセス定義のコンボボックスを読み込みます。
- 2)その後、フィルターを表示するためのWebコンポーネントとグリッドページングコントロールを表示するためのWebコンポーネントの2つを作成します。

セッションの検証- WorkflowCheckServerSession

```

1
2 &server.Load(&webSession.Get(WorkflowWebSession.SessionHandle))
3 If &server.Error.Code > 0
4     WorkflowSignIn.Link()
5 Endif
6

```

名前	タイプ	Is Collection	デスクリプション
Variables			
Standard Variables			
server	WorkflowServer	<input type="checkbox"/>	server
session	WorkflowSession	<input type="checkbox"/>	session
webSession	WebSession	<input type="checkbox"/>	web Session

- WebセッションからセッションハンドラIDを取得し、この接続をworkflowServer変数にロードします。
- 読み込みエラーの場合は、WebPanel「WorkflowSignIn」にリダイレクトします。
- セッションをロードした後、ログインしたユーザーがサーバーに接続されます。

Workflow Inbox – リフレッシュとフィルター

```
10 Event Grid.Refresh
11   Do 'Load Filters'
12     &workitems = &user.GetWorklistOrderBy(&filter, WorkflowOrder.CREATED_DESC)
13   EndEvent
14
15 Sub 'Load Filters'
16   &filter.Load(&session.Get(WorkflowEntryPoint.INBOX + '!Filter'))
17   &page = Val(&session.Get(WorkflowEntryPoint.INBOX + '!Page'))
18   If &page > 1
19     &filter.Start = (&page - 1) * WorkflowPaging.SIZE
20   Else
21     &filter.Start = 0
22   Endif
23   &filter.Limit = WorkflowPaging.SIZE
24 EndSub
```

フィルターは
WorkflowFiltersというWeb
コンポーネントにあります。
フィルターはWebSession
にSDT形式で保存されます。

Workflow Filters – リフレッシュとフィルター

WorkflowFilters

Web Layout | Rules | Events | Conditions | Variables | Help | Documentation

選択されているアクショングループはありません。

MainTable

Priority	&cmbPriority		
State	&cmbState		
Subject	&subject		
Process	&cmbProcessDefinition		
Activity	&cmbActivity		
Name	&name		
User	&cmbUser		
Event Type	&cmbEventType		
Ended From	&endedFrom	To	&endedTo
Created From	&createdFrom	To	&createdTo
Apply			

```

104 Sub 'Load Filters'
105   &filter.Load(&session.Get(&entryPoint + '!Filter'))
106   &cmbPriority      = &filter.Priority
107   &cmbState        = &filter.State
108   &subject         = &filter.Subject
109   &name            = &filter.Name
110   &createdFrom     = &filter.CreatedFrom
111   &createdTo       = &filter.CreatedTo
112   &endedFrom       = &filter.EndedFrom
113   &endedTo         = &filter.EndedTo
114   &cmbEventType    = &filter.EventType
115   &cmbProcessDefinition = &filter.ProcessDefinition.Name
116   &cmbActivity     = &filter.Activity.Name
117   &cmbUser         = &filter.User.Id
118   Do Case
119     Case &entryPoint = WorkflowEntryPoint.OUTBOX
120       If &filter.EndedFrom.IsEmpty()
121         &endedFrom = &Today
122       Endif
123       If &filter.EndedTo.IsEmpty()
124         &endedTo = &Today
125       Endif
126     Case &entryPoint = WorkflowEntryPoint.MY_PROCESSES Or &entryPoint = WorkflowEntryPoint.PROCESSES
127       Or &entryPoint = WorkflowEntryPoint.TASKS Or &entryPoint = WorkflowEntryPoint.DOCUMENTS
128       Or &entryPoint = WorkflowEntryPoint.MY_DOCUMENTS Or &entryPoint = WorkflowEntryPoint.EVENTS
129       If &filter.CreatedFrom.IsEmpty()
130         &createdFrom = &Today
131       Endif
132       If &filter.CreatedTo.IsEmpty()
133         &createdTo = &Today
134       Endif
135     EndCase
136 EndSub

```

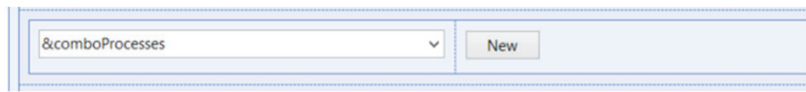
Workflow Inbox – リフレッシュとロード

```
10 ⊢ Event Grid.Refresh
11
12     Do 'Load Filters'
13         &workitems = &user.GetWorklistOrderBy(&filter, WorkflowOrder.CREATED_DESC)
14
15     EndEvent
16
17 ⊢ Event Grid.Load
18
19     For &workitem in &workitems
20
21         &id      = &workitem.Id
22         &activity = &workitem.Activity.Name
23         &state   = WorkflowWorkitemState.Convert(&workitem.State)
24         &subject = &workitem.ProcessInstance.Subject
25         &priority = WorkflowPriority.Convert(&workitem.Priority)
26         &created = &workitem.Created
27
28         Grid.Load()
29
30     Endfor
31
32 EndEvent
33
```

ユーザーのワークリストを取得します。(ワークリストにはユーザーが実行する必要があるすべての作業項目が含まれます。)

リストの各項目について、その内容を受信トレイのグリッドにリストします。

Workflow Inbox – 「新規」アクション



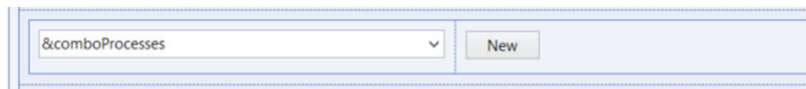
```
104 Sub 'New'
105   If &comboProcesses > 0
106     &processDefinition.Load(&comboProcesses)
107     &processInstance = &processDefinition.CreateInstance()
108     &error = &processDefinition.Error
109     If &error.Code > 0
110       Do 'Error'
111     Else
112       &processInstance.Owner = &user
113       &processInstance.Start()
114       &error = &processInstance.Error
115       If &error.Code > 0
116         Do 'Error'
117       Endif
118     Endif
119     Commit
120   Endif
121 EndSub

270 Sub 'Error'
271   msg(&error.Message)
272 EndSub
```

コンボ ボックスで選択された
プロセス ID を使用して、プ
ロセス定義を
&processDefinition 変数に読
み込みます。

ロードされたプロセス定義
の新しいプロセスインスタ
ンスを作成します

Workflow Inbox – 「新規」アクション



```
104 Sub 'New'
105   If &comboProcesses > 0
106     &processDefinition.Load(&comboProcesses)
107     &processInstance = &processDefinition.CreateInstance()
108     &error = &processDefinition.Error
109     If &error.Code > 0
110       Do 'Error'
111     Else
112       &processInstance.Owner = &user
113       &processInstance.Start()
114       &error = &processInstance.Error
115       If &error.Code > 0
116         Do 'Error'
117       Endif
118     Endif
119     Commit
120   Endif
121 EndSub
```

```
270 Sub 'Error'
271   msg(&error.Message)
272 EndSub
```

プロセスインスタンスのプロセス所有者を割り当てます

プロセスの実行を開始する

Workflow Inbox – 「実行」アクション

```
38 Event 'Execute'
39   &action = WorkflowAction.EXECUTE
40   Do 'Button Pressed'
41 EndEvent
42
78 Sub 'Button Pressed'
79   &workitem.Load(&id)
80   Do Case
81     Case &action = WorkflowAction.EXECUTE
82       Do 'Execute'
83     Case &action = WorkflowAction.COMPLETE
84       Do 'Complete'
85     Case &action = WorkflowAction.PREVIEW
86       Do 'Preview'
87     Case &action = WorkflowAction.DELEGATE
88       Do 'Delegate'
89     Case &action = WorkflowAction.COLLABORATE
90       Do 'Collaborate'
91     Case &action = WorkflowAction.VIEW_HISTORY
92       Do 'View History'
93     Case &action = WorkflowAction.ADD_COMMENTS
94       Do 'Add Comments'
95     Case &action = WorkflowAction.VIEW_DOCUMENTS
96       Do 'Documents'
97   EndCase
98   Commit
99   If &error.Code > 0
100     Do 'Error'
101   Endif
102 EndSub
```

```
155 Sub 'Execute'
156   Do 'Set In-Process State'
157   If &error.Code = 0
158     &app = &workitem.Activity.Application
159     If Not &app.IsEmpty()
160       Do 'Open App'
161     Else
162       Do 'Documents'
163     Endif
164   Endif
165 EndSub
```

```
Sub 'Set In-Process State'
  Do 'Take'
  If &error.Code = 0
    &workitem.ChangeState(WorkflowWorkitemState.OPEN_ACTIVE_INPROCESS)
    &error = &workitem.Error
  Endif
EndSub
```

次ページ

Workflow Inbox – 「実行」アクション

```
145 Sub 'Take'
146   Do Case
147     Case &workitem.State = WorkflowWorkitemState.OPEN_ACTIVE_READY
148       &workitem.Assign(&user)
149       &error = &workitem.Error
150     Case &workitem.State = WorkflowWorkitemState.OPEN_ACTIVE_ASSIGNED
151       If &workitem.Participant.Id = &user.Id
152         //Do nothing
153       Else
154         If &workitem.Participant.Id = !'N/A' //Assigned to a role or a list of users
155           &workitem.Reassign(&workitem.Participant, &user)
156           &error = &workitem.Error
157         Else
158           &error.Code = 203 //The task is already assigned to another user
159         Endif
160       Endif
161     Otherwise
162       &error.Code = 200 //Invalid transition
163     EndCase
164   EndSub
165 EndSub
166
```

Workflow Inbox – 「実行」アクション

```

38 Event 'Execute'
39   &action = WorkflowAction.EXECUTE
40   Do 'Button Pressed'
41 EndEvent
42
78 Sub 'Button Pressed'
79   &workitem.Load(&id)
80   Do Case
81     Case &action = WorkflowAction.EXECUTE
82       Do 'Execute'
83     Case &action = WorkflowAction.COMPLETE
84       Do 'Complete'
85     Case &action = WorkflowAction.PREVIEW
86       Do 'Preview'
87     Case &action = WorkflowAction.DELEGATE
88       Do 'Delegate'
89     Case &action = WorkflowAction.COLLABORATE
90       Do 'Collaborate'
91     Case &action = WorkflowAction.VIEW_HISTORY
92       Do 'View History'
93     Case &action = WorkflowAction.ADD_COMMENTS
94       Do 'Add Comments'
95     Case &action = WorkflowAction.VIEW_DOCUMENTS
96       Do 'Documents'
97   EndCase
98   Commit
99   If &error.Code > 0
100     Do 'Error'
101   Endif
102 EndSub

```

```

155 Sub 'Execute'
156   Do 'Set In-Process State'
157   If &error.Code = 0
158     &app = &workitem.Activity.Application
159     If Not &app.IsEmpty()
160       Do 'Open App'
161     Else
162       Do 'Documents'
163     Endif
164   Endif
165 EndSub

```

```

258 Sub 'Documents'
259   If &workitem.Activity.canWorkWithDocuments = True
260     If &workitem.State <> WorkflowWorkitemState.OPEN_ACTIVE_INPROCESS
261       Do 'Set In-Process State'
262     Endif
263     If &error.Code = 0
264       &window.Object = WorkflowWorkWithDocuments.Create(&workitem.Id)
265       &window.Open()
266     Endif
267   Else
268     msg('Operation not allowed')
269   Endif
270 EndSub

```

```

277 Sub 'Open App'
278   If Not &app.IsEmpty()
279     &app = WorkflowBuildApplicationUrl(&app, &workitem)
280     &window.Url = &app
281     &window.Autosize = False
282     &window.Width = WorkflowWindowSize.APP_WIDTH
283     &window.Height = WorkflowWindowSize.APP_HEIGHT
284     &window.Open()
285   Endif
286 EndSub

```

Diagram illustrating the execution flow of the 'Execute' action in the Workflow Inbox. The flow starts with the 'Execute' event, which triggers the 'Button Pressed' sub-process. This sub-process contains a 'Case' statement that branches based on the action type. For the 'EXECUTE' action, it calls the 'Execute' sub-process. The 'Execute' sub-process then checks for an error. If there is no error, it calls the 'Open App' sub-process. If there is an error, it calls the 'Documents' sub-process. The 'Open App' sub-process checks if the application is empty. If not empty, it builds the application URL and opens the application window. If empty, it calls the 'Documents' sub-process. The 'Documents' sub-process checks if the user can work with documents. If yes, it sets the state to 'OPEN_ACTIVE_INPROCESS' and creates a new document object, opening the document window. If no, it displays an error message.

Workflow Inbox – 「送信」アクション

```

43 Event 'Complete'
44   &action = WorkflowAction.COMPLETE
45   Do 'Button Pressed'
46 EndEvent
47
78 Sub 'Button Pressed'
79   &workitem.Load(&id)
80   Do Case
81     Case &action = WorkflowAction.EXECUTE
82       Do 'Execute'
83     Case &action = WorkflowAction.COMPLE
84       Do 'Complete'
85     Case &action = WorkflowAction.PREVJ
86       Do 'Preview'
87     Case &action = WorkflowAction.DELEG
88       Do 'Delegate'
89     Case &action = WorkflowAction.COLL
90       Do 'Collaborate'
91     Case &action = WorkflowAction.VIEW
92       Do 'View History'
93     Case &action = WorkflowAction.ADD
94       Do 'Add Comments'
95     Case &action = WorkflowAction.VIEW
96       Do 'Documents'
97   EndCase
98   Commit
99   If &error.Code > 0
100     Do 'Error'
101   Endif
102 EndSub

```

```

77 Sub 'Complete'
78   If &workitem.State = WorkflowWorkitemState.OPEN_ACTIVE_INPROCESS
79     &workitem.Complete()
80     &error = &workitem.Error
81     If &error.Code > 0
82       Do Case
83         Case &error.Code = WorkflowError.OPTIONALS_SELECTION_REQUIRED
84           &window.Object = WorkflowSelectActivity.Create(&workitem.Id, WorkflowSelectionMode.OPTIONALS, WorkflowAction.COMPLETE)
85           &window.Open()
86           &error = new()
87         Case &error.Code = WorkflowError.ADHOC_SELECTION_REQUIRED
88           &window.Object = WorkflowSelectActivity.Create(&workitem.Id, WorkflowSelectionMode.ADHOC, WorkflowAction.COMPLETE)
89           &window.Open()
90           &error = new()
91         Case &error.Code = WorkflowError.COMMENTS_REQUIRED
92           &window.Object = WorkflowComments.Create(&workitem.Id, WorkflowObjectType.WORKITEM, False)
93           &window.Open()
94           &error = new()
95       EndCase
96     Else
97       &error.Code = 204 // The task has not been processed yet
98     Endif
99   EndSub

```

Workflow Inbox – 「履歴」アクション

```
48 Event 'History'
49   &action = WorkflowAction.VIEW_HISTORY
50   Do 'Button Pressed'
51 EndEvent

78 Sub 'Button Pressed'
79   &workitem.Load(&id)
80   Do Case
81     Case &action = WorkflowAction.EXECUTE
82       Do 'Execute'
83     Case &action = WorkflowAction.COMPLETE
84       Do 'Complete'
85     Case &action = WorkflowAction.PREVIEW
86       Do 'Preview'
87     Case &action = WorkflowAction.DELEGATE
88       Do 'Delegate'
89     Case &action = WorkflowAction.COLLABORATE
90       Do 'Collaborate'
91     Case &action = WorkflowAction.VIEW_HISTORY
92       Do 'View History'
93     Case &action = WorkflowAction.ADD_COMMENTS
94       Do 'Add Comments'
95     Case &action = WorkflowAction.VIEW_DOCUMENTS
96       Do 'Documents'
97   EndCase
98   Commit
99   If &error.Code > 0
100     Do 'Error'
101   Endif
102 EndSub
...
```

240 Sub 'View History'

241 &window.Object = WorkflowHistory.Create(&workitem.ProcessInstanceId)

242 &window.Open()

243 EndSub

...

Workflow History – 実行画面

Workflow History						
Query	Skip	Undo	History			
Subject	Activity	State	Participant	Created	Ended	
Purchases	purchase order	completed	Workflow Administrator	05/10/21 12:49 PM	05/13/21 09:30 AM	
Purchases	Authorization	completed	Workflow Administrator	05/13/21 09:30 AM	05/13/21 09:30 AM	
	Administration Manager			05/13/21 09:30 AM	05/13/21 09:30 AM	
Purchases	Purchasing Manager	completed	Workflow Administrator	05/13/21 09:30 AM	05/13/21 09:30 AM	

Workflow History – レイアウト

The layout consists of two main sections within a container. The top section is a light red rectangular area containing four buttons: "Query", "Skip", "Undo", and "History". The bottom section is a light green rectangular area labeled "GRID" in its top-left corner. It contains a table with 7 columns and 2 rows. The first row contains the column headers: "WfWorkItem", "Subject", "Activity", "State", "Participant", "Created", and "Ended". The second row contains the corresponding data field labels: "&id", "&subject", "&activity", "&state" (with a dropdown arrow), "&participant", "&created", and "&ended".

WfWorkItem	Subject	Activity	State	Participant	Created	Ended
&id	&subject	&activity	&state ▼	&participant	&created	&ended

Workflow History – イベント

Web Layout * Rules Events * Conditions Variables Help Documentation

```
1 param(in: &processInstanceId);  
2  
3 Event Start  
4     &server = WorkflowCheckServerSession()  
5     &user = &server.ConnectedUser  
6     &processInstance.Load(&processInstanceId)  
7 EndEvent  
8  
9 Event Grid.Refresh  
10     &workitems = &processInstance.Workitems  
11 EndEvent  
12  
13 Event Grid.Load  
14     For &workitem in &workitems  
15         &id = &workitem.Id  
16         &activity = &workitem.Activity.Name  
17         &state = WorkflowWorkitemState.Convert(&workitem.State)  
18         &subject = &workitem.ProcessInstance.Subject  
19         &participant = &workitem.Participant.Name  
20         &created = &workitem.Created  
21         &ended = &workitem.Ended  
22     Endfor  
23     Grid.Load()  
24 EndEvent
```

Start イベント :

- 1.有効なセッションがあるか確認する
- 2.プロセスインスタンスID変数からプロセスインスタンスをロードする

Grid.Refresh イベント :

- 3.ワークフロープロセスインスタンス変数のすべての作業項目を取得します。workflowWorkitemsのリストを返します。

Grid.Load イベント :

- 4.リストの各作業項目をグリッドにロードします

Workflow Inbox – 「削除」アクション

```
58 Event 'Delegate'
59   &action = WorkflowAction.DELEGATE
60   Do 'Button Pressed'
61 EndEvent
62

78 Sub 'Button Pressed'
79   &workitem.Load(&id)
80   Do Case
81     Case &action = WorkflowAction.EXECUTE
82       Do 'Execute'
83     Case &action = WorkflowAction.COMPLETE
84       Do 'Complete'
85     Case &action = WorkflowAction.PREVIEW
86       Do 'Preview'
87     Case &action = WorkflowAction.DELEGATE
88       Do 'Delegate'
89     Case &action = WorkflowAction.COLLABORATE
90       Do 'Collaborate'
91     Case &action = WorkflowAction.VIEW_HISTORY
92       Do 'View History'
93     Case &action = WorkflowAction.ADD_COMMENTS
94       Do 'Add Comments'
95     Case &action = WorkflowAction.VIEW_DOCUMENTS
96       Do 'Documents'
97   EndCase
98   Commit
99   If &error.Code > 0
100     Do 'Error'
101   Endif
102 EndSub

241 Sub 'Delegate'
242   If &workitem.Activity.canDelegate = True
243     &window.Object = WorkflowAssign.Create(&workitem.Id, WorkflowAction.DELEGATE)
244     &window.Open()
245   Else
246     msg('Operation not allowed')
247   Endif
248 EndSub
```



Videos

training.genexus.com

Documentation

wiki.genexus.com

Certifications

training.genexus.com/certifications