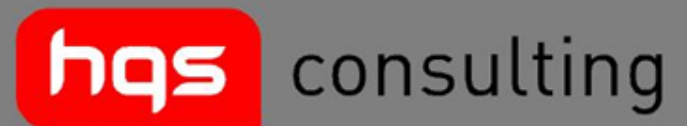


Component Interface



TÓPICOS

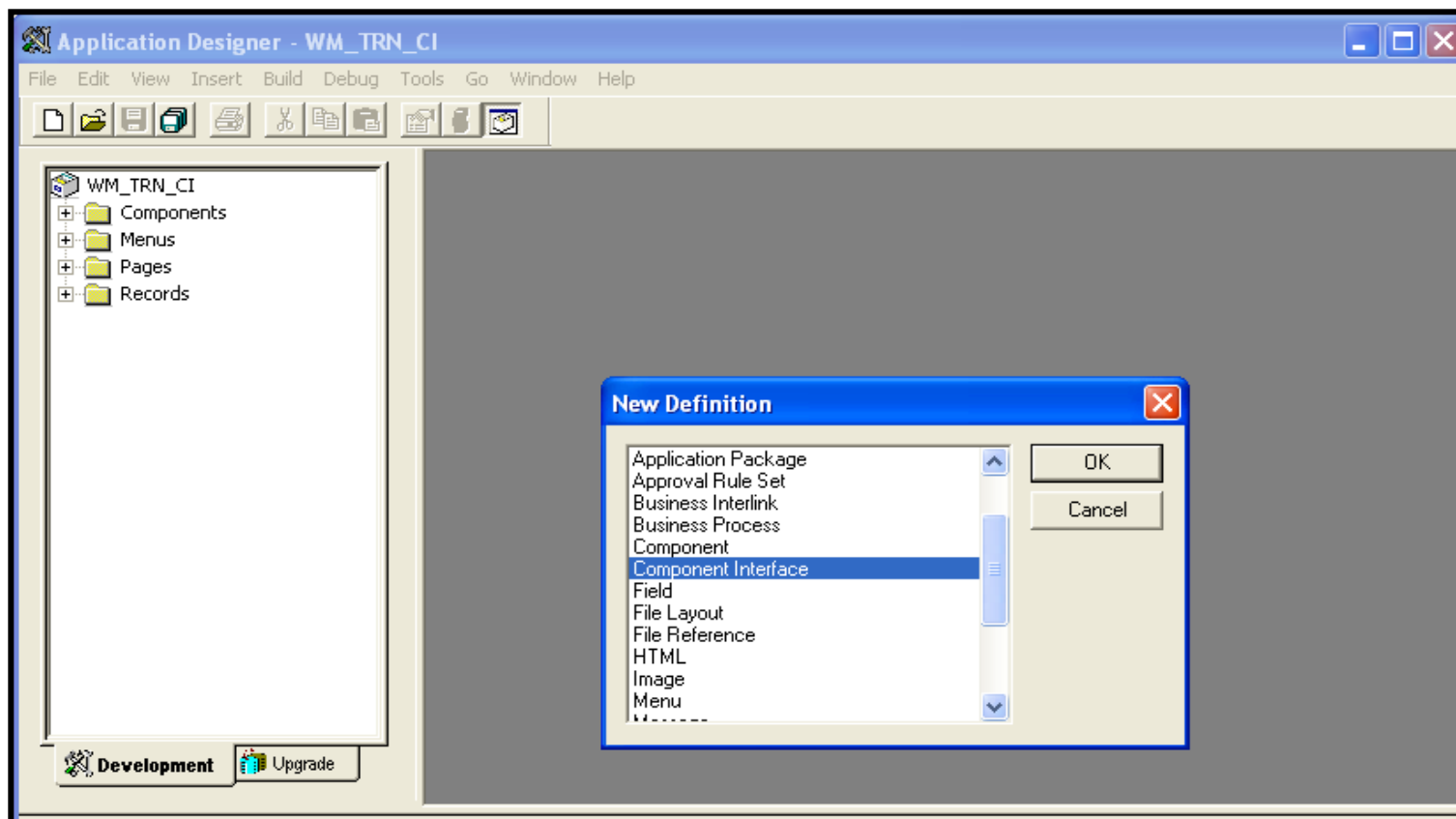
- **Introdução**
- **Criando o Component Interface**
- **Segurança no Component Interface**
- **Trabalhando com o Component Interface**
- **Resultado**

INTRODUÇÃO

- O que faz um Component Interface?
Um component interface permite à um PeopleCode ou programa externo acessar e modificar dados de um componente.
- Por que usar Component Interface?
O Component Interface executa toda a lógica de negócio construída no componente. Como resultado, fornece um nível confiável das validações de dados (Maior do que uma simples instrução SQL)

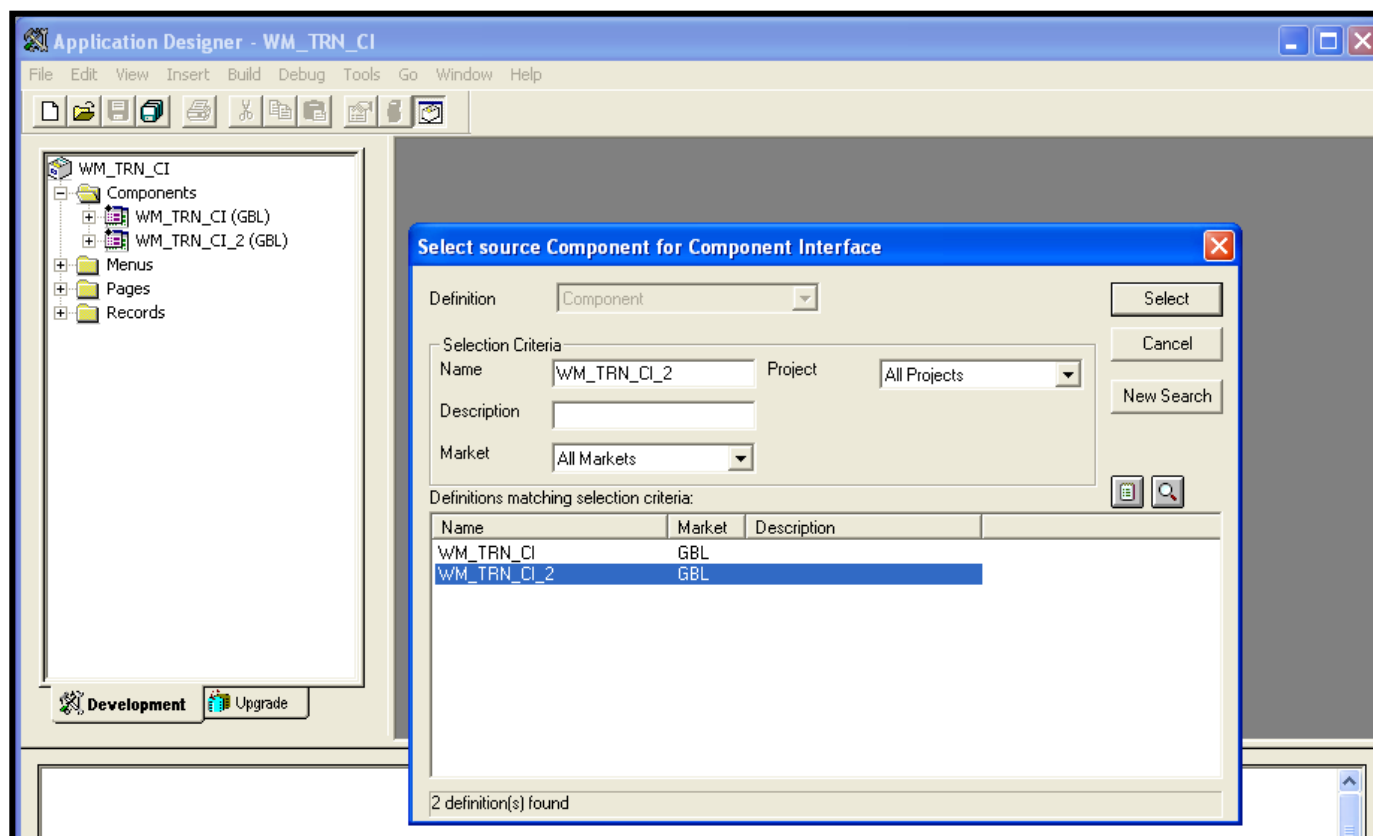
Criando o Component Interface

O Component Interface é criado no Application Designer, tal como uma definição de Record ou field: File > New (Ctrl + N)



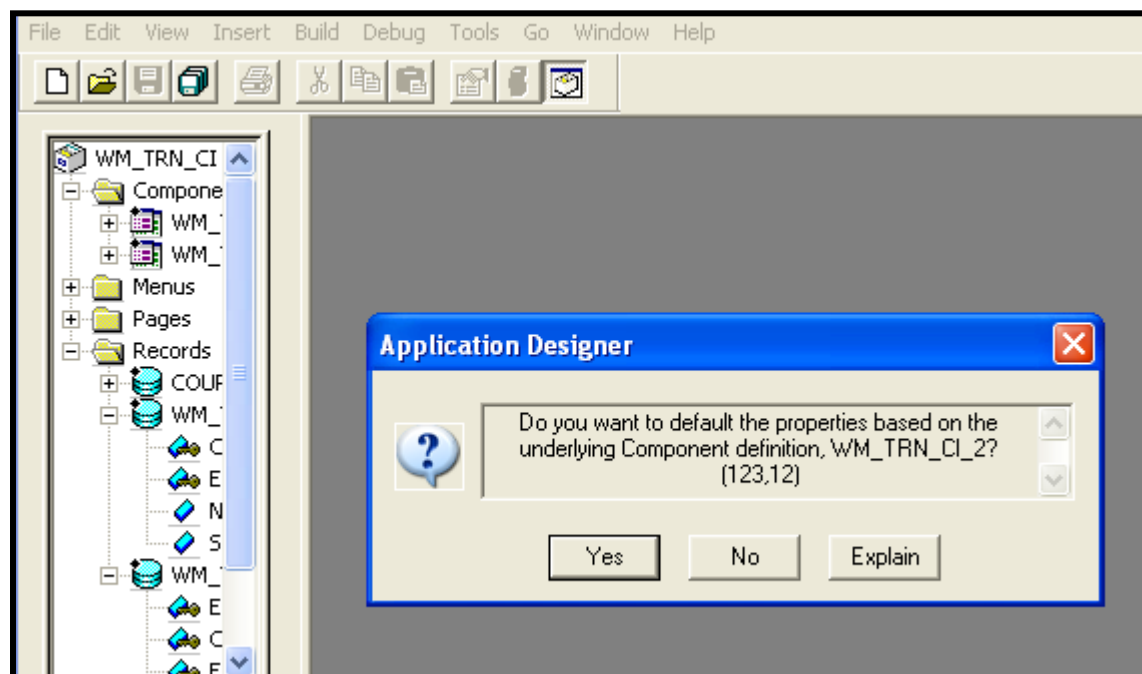
Criando o Component Interface

Em seguida, selecionamos o componente que desejamos mapear. Neste exemplo usaremos o componente WM_TRN_CI_2:



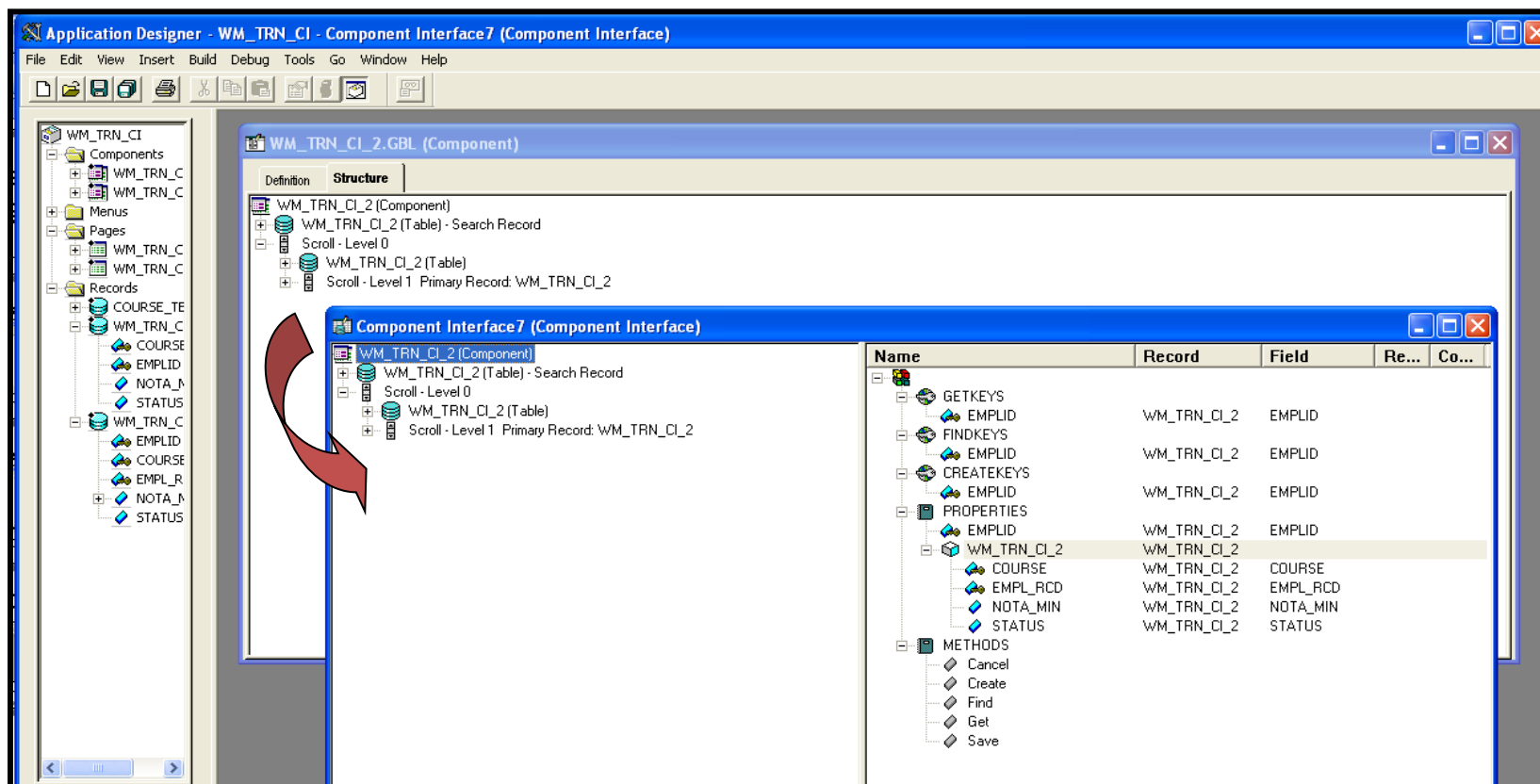
Criando o Component Interface

O Application Designer perguntará se você quer mapear todas as records e fields do component. A resposta sim construirá o Component Interface com as propriedades padrão:



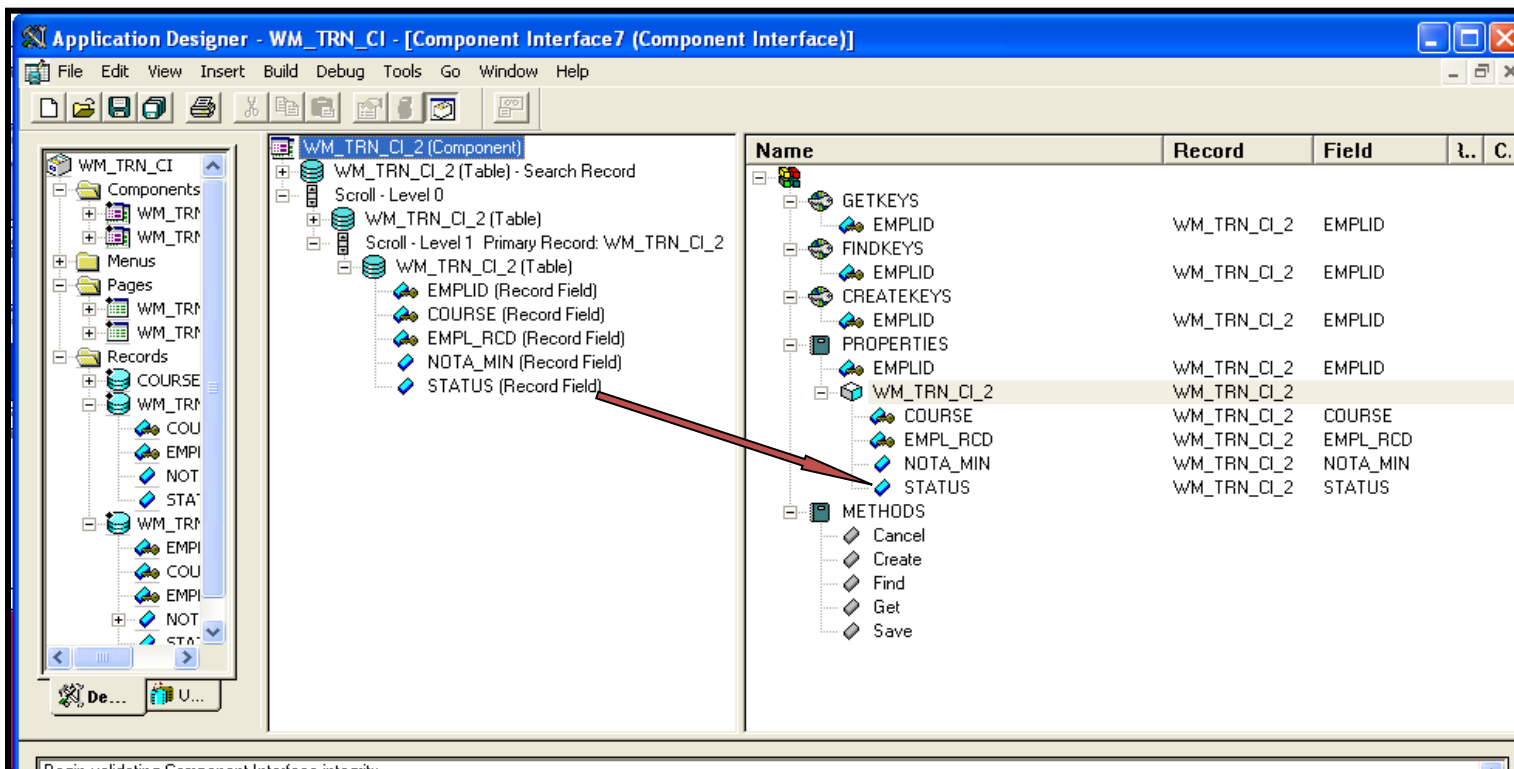
Criando o Component Interface

Observe que o Component Interface é dividido em duas partes. Do lado esquerdo está a estrutura idêntica à do Component:



Criando o Component Interface

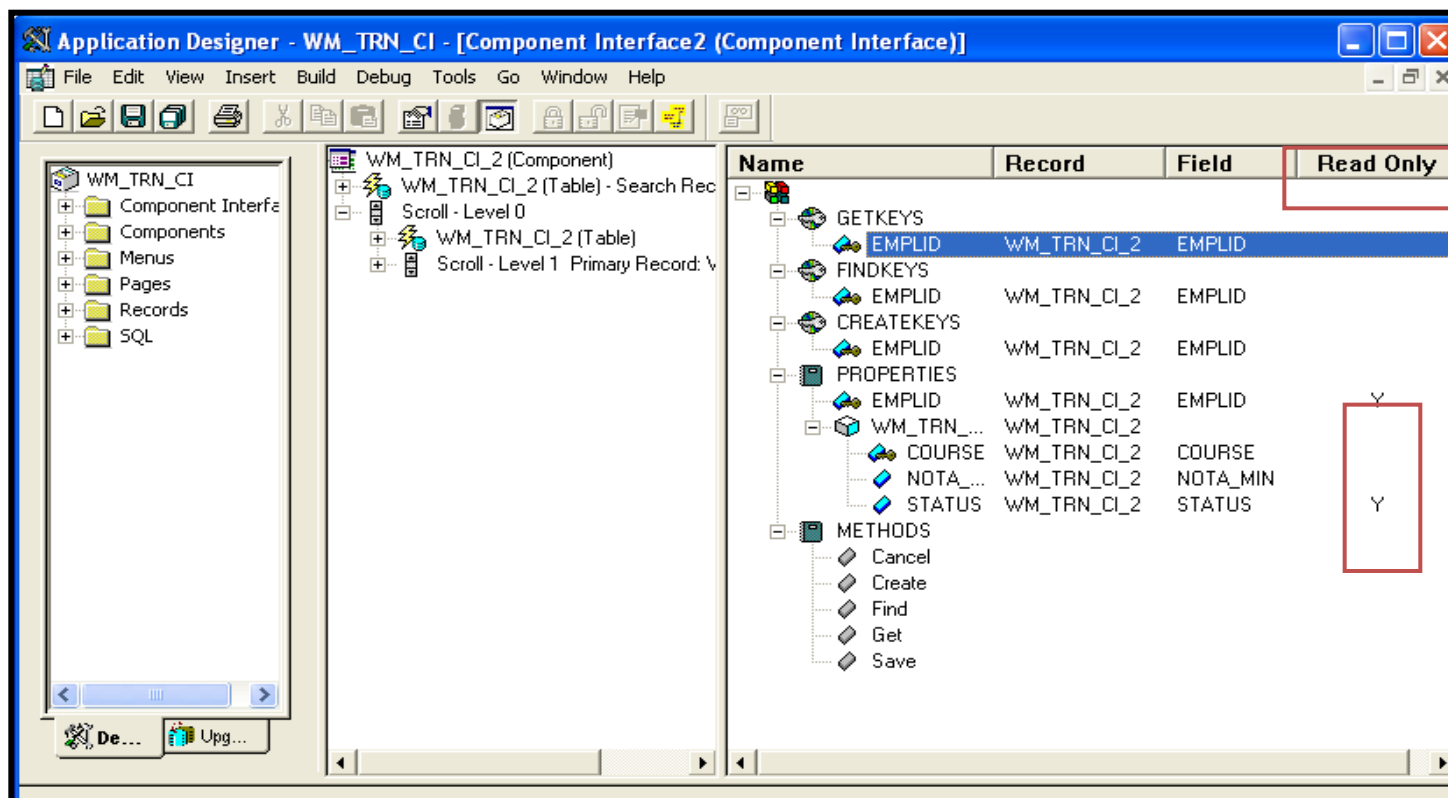
Do lado direito está o mapa das records / fields (*PROPERTIES*) e as ações possíveis de serem executadas nesse component (*METHODS*). Os três primeiros grupos (*Keys*) contêm os campos *search* component.



Name	Record	Field
GETKEYS	WM_TRN_CI_2	EMPLID
FINDKEYS	WM_TRN_CI_2	EMPLID
CREATEKEYS	WM_TRN_CI_2	EMPLID
PROPERTIES	WM_TRN_CI_2	EMPLID
WM_TRN_CI_2	WM_TRN_CI_2	STATUS
COURSE	WM_TRN_CI_2	COURSE
EMPL_RCD	WM_TRN_CI_2	EMPL_RCD
NOTA_MIN	WM_TRN_CI_2	NOTA_MIN
STATUS	WM_TRN_CI_2	STATUS

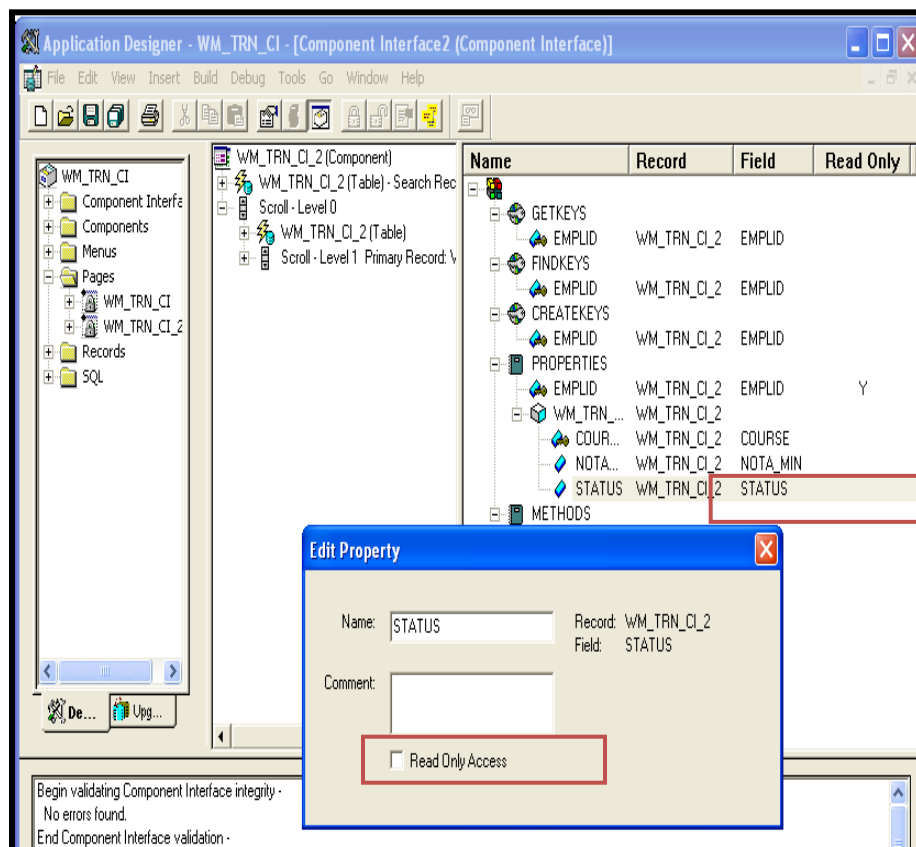
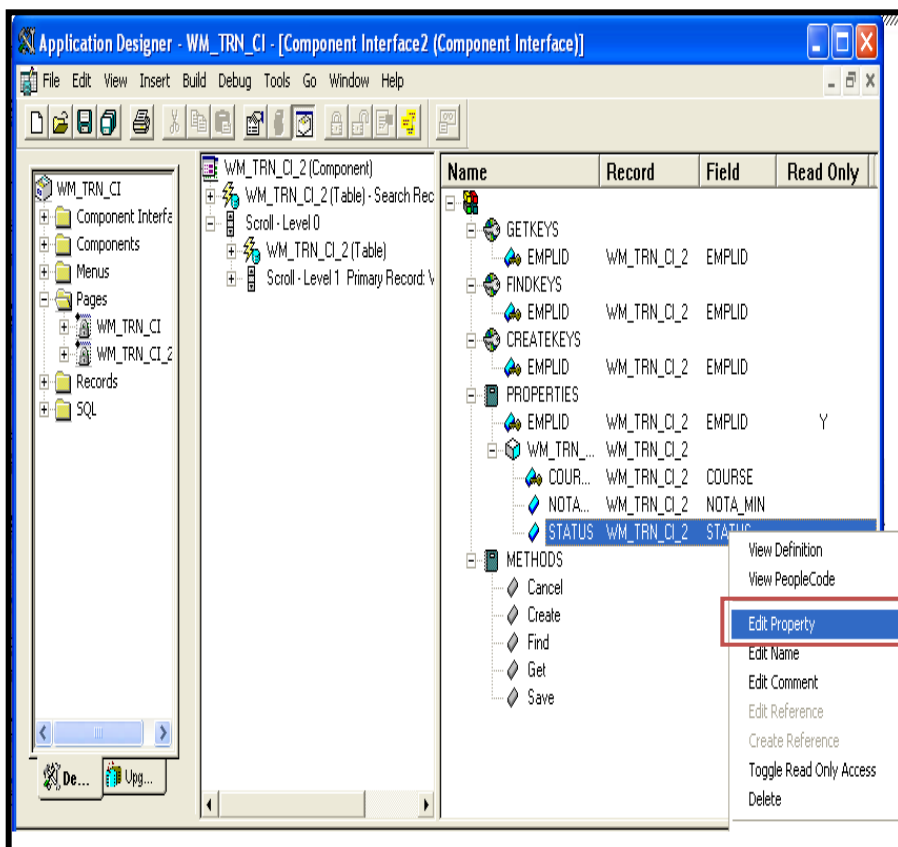
Criando o Component Interface

Observe que alguns campos possuem a propriedade “*Read Only*” (Atribuída aos campos usados como “*Display Only*” na página):



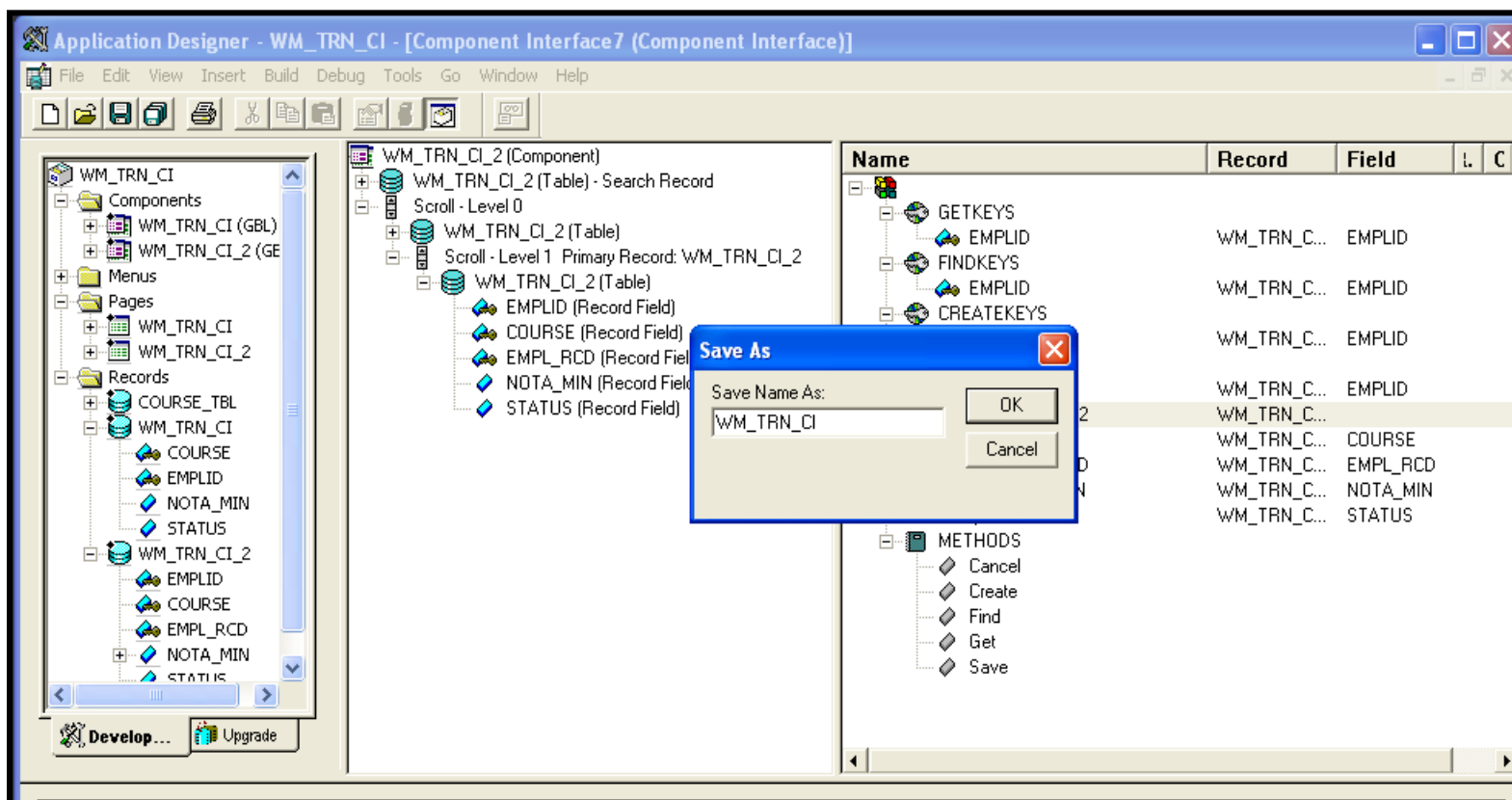
Criando o Component Interface

Caso seja necessário atribuir algum valor à esses campos, temos que desmarcar a opção “*Read Only*” no Component Interface:



Criando o Component Interface

Agora, podemos salvar nosso Component Interface: File > Save (Ctrl +S)



Segurança no Component Interface

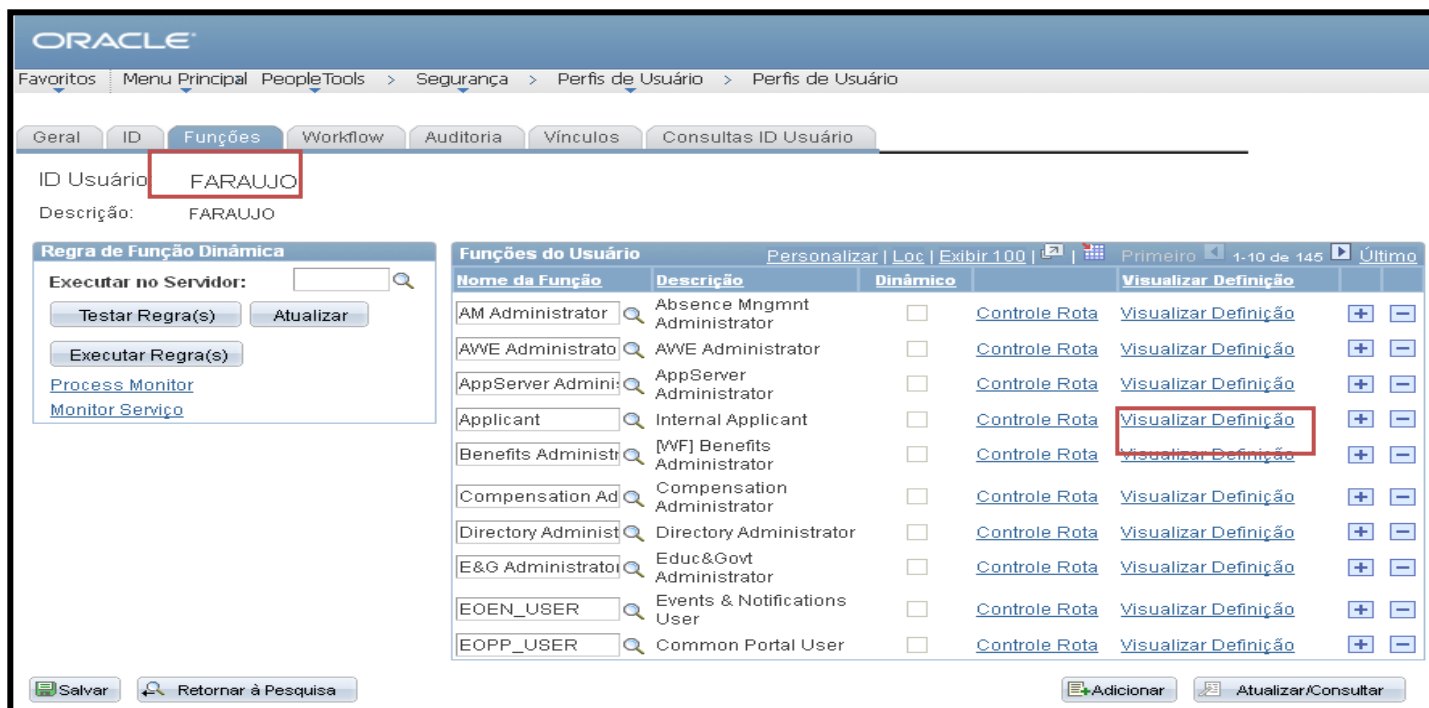
- Para podermos ter acesso ao Component Interface, antes de qualquer coisa, devemos inseri-lo em uma lista de permissão.
Neste exemplo iremos acrescentá-lo na lista de permissão HCCPRS1050 (Que dá ao usuário acesso à página que utilizará o Component Interface).

Segurança no Component Interface

Podemos ir direto na lista de permissão, ou abrir o Perfil de usuário:

Menu Principal > People Tools > Segurança > Perfis de Usuário > Perfis de Usuário.

Na aba Funções, escolhemos a função e clicamos em VISUALIZAR DEFINIÇÃO:



ORACLE

Favoritos | Menu Principal | PeopleTools > Segurança > Perfis de Usuário > Perfis de Usuário

Geral | ID | **Funções** | Workflow | Auditoria | Vínculos | Consultas ID Usuário

ID Usuário: FARAUJO

Descrição: FARAUJO

Regra de Função Dinâmica

Executar no Servidor:

Testar Regra(s) Atualizar

Executar Regra(s)

[Process Monitor](#)

[Monitor Serviço](#)

Funções do Usuário Personalizar | Loc | Exibir 100 | Primeiro 1-10 de 145 Último


Nome da Função	Descrição	Dinâmico	Controle Rota	Visualizar Definição
AM Administrator	Absence Mngmnt Administrator	<input type="checkbox"/>	Controle Rota	Visualizar Definição
AWE Administrato	AWE Administrator	<input type="checkbox"/>	Controle Rota	Visualizar Definição
AppServer Admini	AppServer Administrator	<input type="checkbox"/>	Controle Rota	Visualizar Definição
Applicant	Internal Applicant	<input type="checkbox"/>	Controle Rota	Visualizar Definição
Benefits Administ	[WF] Benefits Administrator	<input type="checkbox"/>	Controle Rota	Visualizar Definição
Compensation Ad	Compensation Administrator	<input type="checkbox"/>	Controle Rota	Visualizar Definição
Directory Administ	Directory Administrator	<input type="checkbox"/>	Controle Rota	Visualizar Definição
E&G Administrato	Educ&Govt Administrator	<input type="checkbox"/>	Controle Rota	Visualizar Definição
EOEN_USER	Events & Notifications User	<input type="checkbox"/>	Controle Rota	Visualizar Definição
EOPP_USER	Common Portal User	<input type="checkbox"/>	Controle Rota	Visualizar Definição

Salvar Retornar à Pesquisa

Adicionar Atualizar/Consultar

Segurança no Component Interface

Na aba Lista de Permissão, escolhemos a lista e clicamos em VISUALIZAR DEFINIÇÃO:



ORACLE

Favoritos | Menu Principal | PeopleTools > Segurança > Perfis de Usuário > Perfis de Usuário

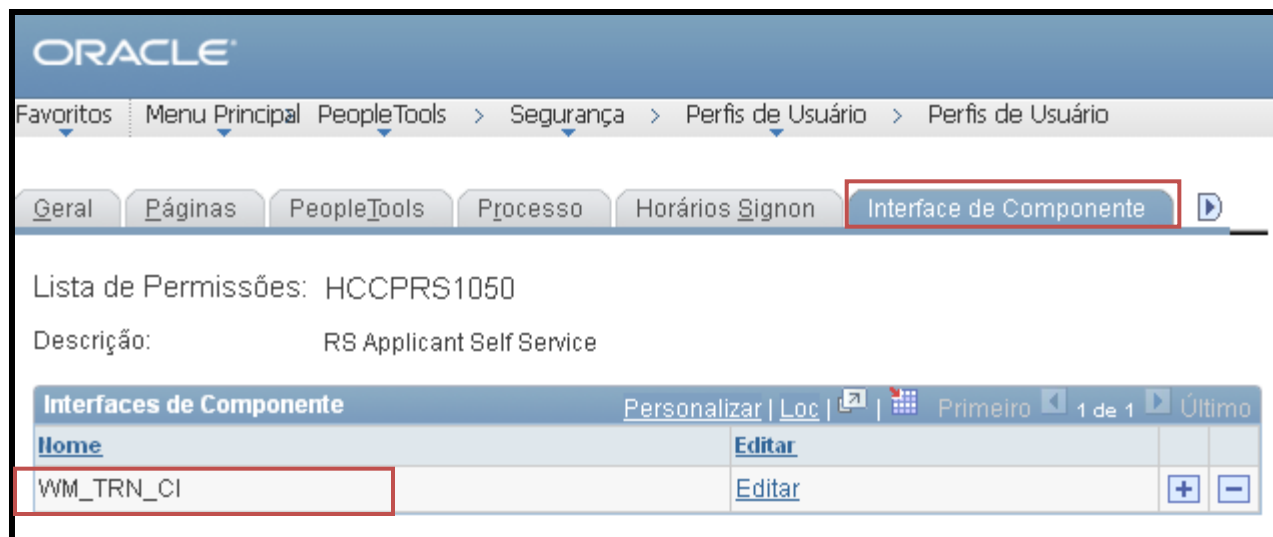
[Listas de Permissões](#)
[Membros](#)
[Membros Dinâmicos](#)
[Workflow](#)
[Atribuição de Função](#)
[Vínculos](#)

Função: Applicant
Descrição: Internal Applicant

Listas de Permissões		Personalizar	Loc	Exibir Tudo	Primeiro	1-3 de 3	Último
Lista de Permissões	Descrição				Visualizar Definição		
HCCPRS1050	RS Applicant Self Service				Visualizar Definição	+	-
HCSPCMPINT	Standard Comp Intf Permissions				Visualizar Definição	+	-
HCSPWEBLIB	Standard WebLib Permissions				Visualizar Definição	+	-

Segurança no Component Interface

Então, podemos adicioná-lo na aba Interface de Componente. Se você salvar neste ponto, o People Tools removerá ele da lista de permissão por que ainda não foi dado o acesso. Para isso, clicamos no Link Editar.



ORACLE®

Favoritos | Menu Principal | PeopleTools > Segurança > Perfis de Usuário > Perfis de Usuário

Geral | Páginas | PeopleTools | Processo | Horários Signon | **Interface de Componente** | ▶

Lista de Permissões: HCCPRS1050

Descrição: RS Applicant Self Service

Interfaces de Componente		Personalizar	Loc	Primeiro	1 de 1	Último
Home	Editar					
WM_TRN_CI	Editar					

Segurança no Component Interface

Após clicar em Acesso Completo, podemos dar Ok e salvar. Temos agora, então, permissão para acessar nosso Component Interface.



ORACLE

Favoritos | Menu Principal | PeopleTools > Segurança > Perfis de Usuário > Perfis de Usuário

Permissões da Interface de Componente

WM_TRN_CI

Permissão Interf Compon	Personalizar Loc  Primeiro 1-5 de 5 Último
<u>Método</u>	<u>'Acesso Método</u>
Get	Sem Acesso 
Create	Sem Acesso 
Save	Sem Acesso 
Cancel	Sem Acesso 
Find	Sem Acesso 

OK Cancelar

Acesso Completo

Sem Acesso

Trabalhando com o Component Interface

Agora que criamos o component Interface e o incluímos em uma lista de permissão válida, podemos utilizá-lo para nosso objetivo.

Como exemplo, utilizamos o componente da seguinte página:



The screenshot displays a web application interface. At the top, there is a tab labeled "Componente 1". Below the tab, there is a form with the following fields:

- ID Associação**: A text input field containing the value "1".
- Teste CI**: A sub-form with a header bar containing navigation links: "Loc", "Exibir Tudo", "Primeiro", "1 de 4", and "Último".
- Código do Curso**: A text input field containing the value "1".
- Nº Registro do Funcionário**: A text input field containing the value "1".
- Nota Minima**: A text input field containing the value "20,00".
- Status**: A dropdown menu with the value "Ativo" selected.

Trabalhando com o Component Interface

Nosso objetivo, neste exemplo, será atualizar os dados da referida página, cada vez que modificarmos a página abaixo, de acordo com o Emplid e o Curso:

Component 2

Código do Curso 1
Descrição

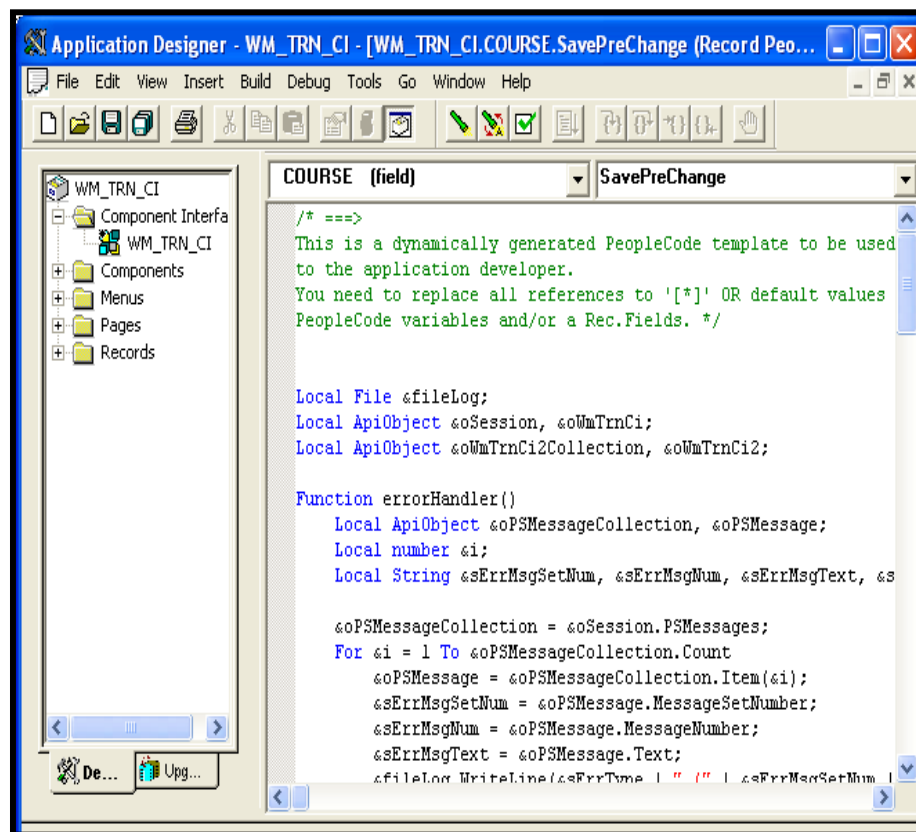
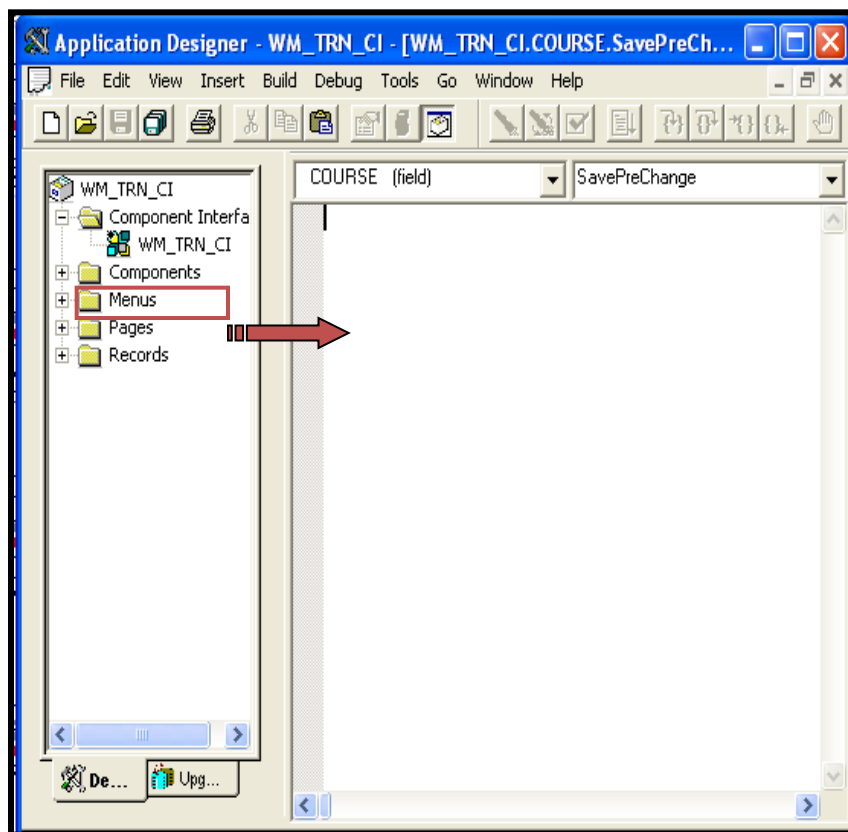
Personalizar Loc  Primeiro 1-3 de 3 Último					
	ID Funcionário	Nota Minima	Status		
1	<input type="text" value="1"/>	<input type="text"/>	<input type="text" value="v"/>	<input type="button" value="+"/>	<input type="button" value="-"/>
2	<input type="text" value="2"/>	<input type="text"/>	<input type="text" value="v"/>	<input type="button" value="+"/>	<input type="button" value="-"/>
3	<input type="text" value="3"/>	<input type="text"/>	<input type="text" value="v"/>	<input type="button" value="+"/>	<input type="button" value="-"/>

Trabalhando com o Component Interface

Com o Component Interface criado será possível acessar e modificar os dados das tabelas do component mapeado. Como o CI engloba todas as regras de negócio, antes de serem modificados, os dados serão validados de acordo com as mesmas.

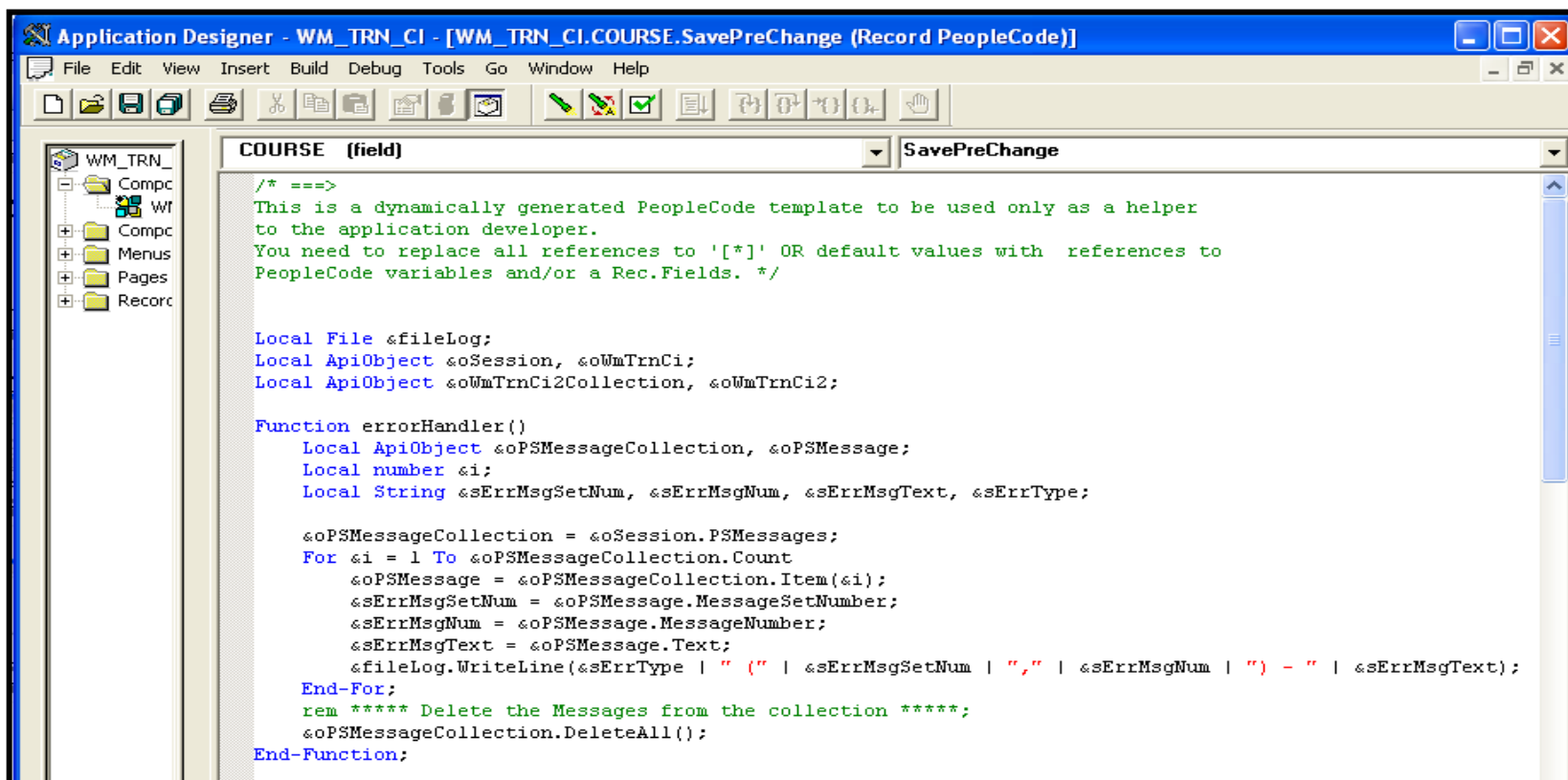
Trabalhando com o Component Interface

O Component Interface permite gerar um “Modelo” de PeopleCode, no qual substituiremos a referencia [*] por valores de campos. Você pode simplesmente arrastá-lo e soltá-lo no programa.



Trabalhando com o Component Interface

Esta primeira parte, declara as variáveis usadas e cria uma função para exibir mensagens de erro, quando necessário.



The screenshot shows the Oracle Application Designer window titled 'Application Designer - WM_TRN_CI - [WM_TRN_CI.COURSE.SavePreChange (Record PeopleCode)]'. The left pane shows a project tree with folders for 'Comp', 'WF', 'Comp', 'Menus', 'Pages', and 'Records'. The main editor displays the 'COURSE' field with the 'SavePreChange' function. The code is as follows:

```

/* ==>
This is a dynamically generated PeopleCode template to be used only as a helper
to the application developer.
You need to replace all references to '['*']' OR default values with references to
PeopleCode variables and/or a Rec.Fields. */

Local File &fileLog;
Local ApiObject &oSession, &oWmTrnCi;
Local ApiObject &oWmTrnCi2Collection, &oWmTrnCi2;

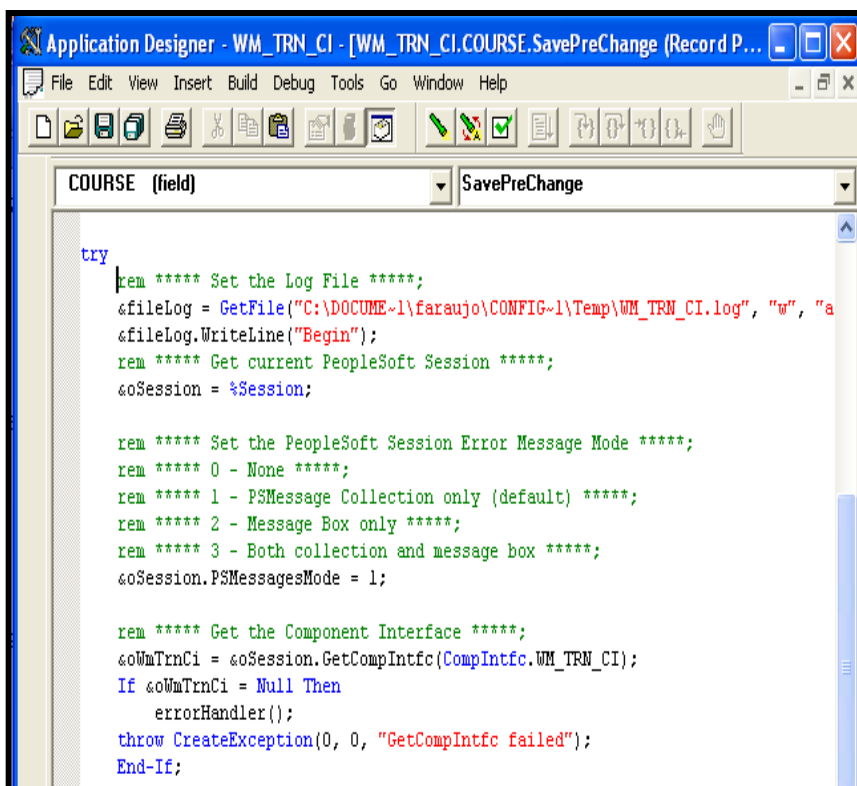
Function errorHandler()
    Local ApiObject &oPSMessageCollection, &oPSMessage;
    Local number &i;
    Local String &sErrMsgSetNum, &sErrMsgNum, &sErrMsgText, &sErrType;

    &oPSMessageCollection = &oSession.PSMessages;
    For &i = 1 To &oPSMessageCollection.Count
        &oPSMessage = &oPSMessageCollection.Item(&i);
        &sErrMsgSetNum = &oPSMessage.MessageSetNumber;
        &sErrMsgNum = &oPSMessage.MessageNumber;
        &sErrMsgText = &oPSMessage.Text;
        &fileLog.WriteLine(&sErrType | " (" | &sErrMsgSetNum | "," | &sErrMsgNum | ") - " | &sErrMsgText);
    End-For;
    rem ***** Delete the Messages from the collection *****;
    &oPSMessageCollection.DeleteAll();
End-Function;

```

Trabalhando com o Component Interface

Então, há o comando *TRY*, no qual está inserido todo o restante do código. A primeira parte cria um arquivo de log. A segunda parte define a forma que as mensagens serão apresentadas. Em seguida é feita uma validação para iniciar o Component Interface. No nosso caso, não escreveremos um arquivo de log.

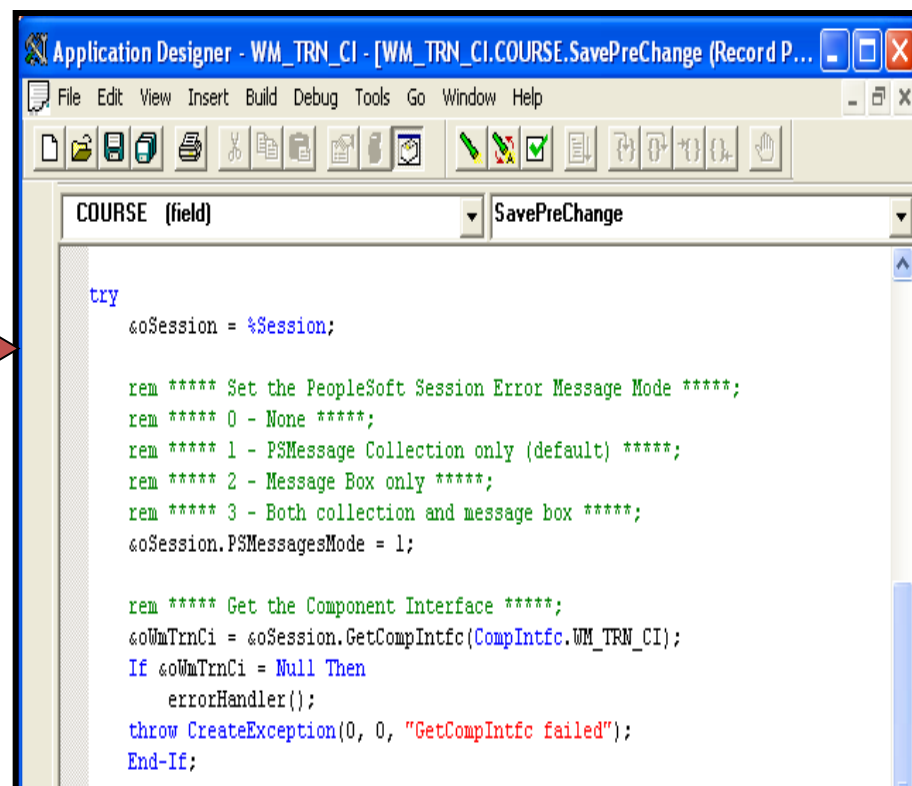


```

try
rem ***** Set the Log File *****;
&fileLog = GetFile("C:\DOCUME~1\faraujo\CONFIG~1\Temp\WM_TRN_CI.log", "w", "a");
&fileLog.WriteLine("Begin");
rem ***** Get current PeopleSoft Session *****;
&oSession = %Session;

rem ***** Set the PeopleSoft Session Error Message Mode *****;
rem ***** 0 - None *****;
rem ***** 1 - PSMessage Collection only (default) *****;
rem ***** 2 - Message Box only *****;
rem ***** 3 - Both collection and message box *****;
&oSession.PSMessagesMode = 1;

rem ***** Get the Component Interface *****;
&oWmTrnCt = &oSession.GetCompIntfc(CompIntfc.WM_TRN_CI);
If &oWmTrnCt = Null Then
    errorHandler();
throw CreateException(0, 0, "GetCompIntfc failed");
End-If;
    
```

```

try
&oSession = %Session;

rem ***** Set the PeopleSoft Session Error Message Mode *****;
rem ***** 0 - None *****;
rem ***** 1 - PSMessage Collection only (default) *****;
rem ***** 2 - Message Box only *****;
rem ***** 3 - Both collection and message box *****;
&oSession.PSMessagesMode = 1;

rem ***** Get the Component Interface *****;
&oWmTrnCt = &oSession.GetCompIntfc(CompIntfc.WM_TRN_CI);
If &oWmTrnCt = Null Then
    errorHandler();
throw CreateException(0, 0, "GetCompIntfc failed");
End-If;
    
```

Trabalhando com o Component Interface

Abaixo, indicamos as ações possíveis do component Interface, com *TRUE* . Em seguida, inserimos o valor da chave com a qual desejamos entrar no component e validamos a mesma.

```

Application Designer - WM_TRN_CI - [WM_TRN_CI.COURSE.SavePreChange (Record P...
File Edit View Insert Build Debug Tools Go Window Help
COURSE (field) SavePreChange

rem ***** Set the Component Interface Mode *****;
&oWmTrnCi.InteractiveMode = False;
&oWmTrnCi.GetHistoryItems = True;
&oWmTrnCi.EditHistoryItems = False;

rem ***** Set Component Interface Get/Create Keys *****;
&oWmTrnCi.EMPLID = [*];

rem ***** Execute Get *****;
If Not &oWmTrnCi.Get() Then
    rem ***** No rows exist for the specified keys.*****;
    errorHandler();
    throw CreateException(0, 0, "Get failed");
End-If;

rem ***** Begin: Get/Set Component Interface Properties *****;
rem ***** Get/Set Level 0 Field Properties *****;
&fileLog.WriteLine("&oWmTrnCi.EMPLID = " | &oWmTrnCi.EMPLID);
rem &oWmTrnCi.EMPLID = [*];
  
```



```

Application Designer - WM_TRN_CI - [WM_TRN_CI.COURSE.SavePreChange (Record P...
File Edit View Insert Build Debug Tools Go Window Help
COURSE (field) SavePreChange

rem ***** Set the Component Interface Mode *****;
&oWmTrnCi.InteractiveMode = True;
&oWmTrnCi.GetHistoryItems = True;
&oWmTrnCi.EditHistoryItems = True;

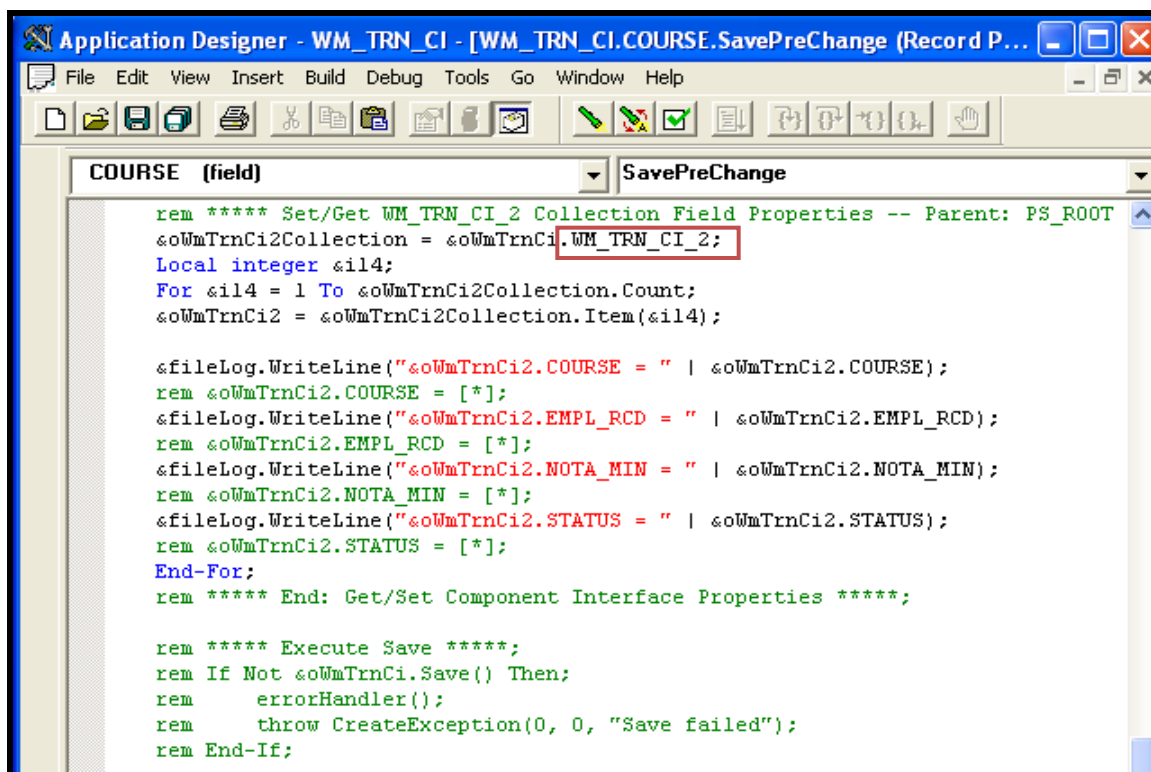
rem ***** Set Component Interface Get/Create Keys *****;
&oWmTrnCi.EMPLID = WM_TRN_CI.EMPLID;

rem ***** Execute Get *****;
If Not &oWmTrnCi.Get() Then
    rem ***** No rows exist for the specified keys.*****;
    errorHandler();
    throw CreateException(0, 0, "Get failed");
End-If;

rem ***** Begin: Get/Set Component Interface Properties *****;
  
```

Trabalhando com o Component Interface

- Nesta parte, há todas as tabelas e campos do Component Interface. No nosso caso, a única tabela mapeada é a: `wm_trn_ci_2`. Como trata-se de uma grid, há um *FOR* que passa por todas as linhas.



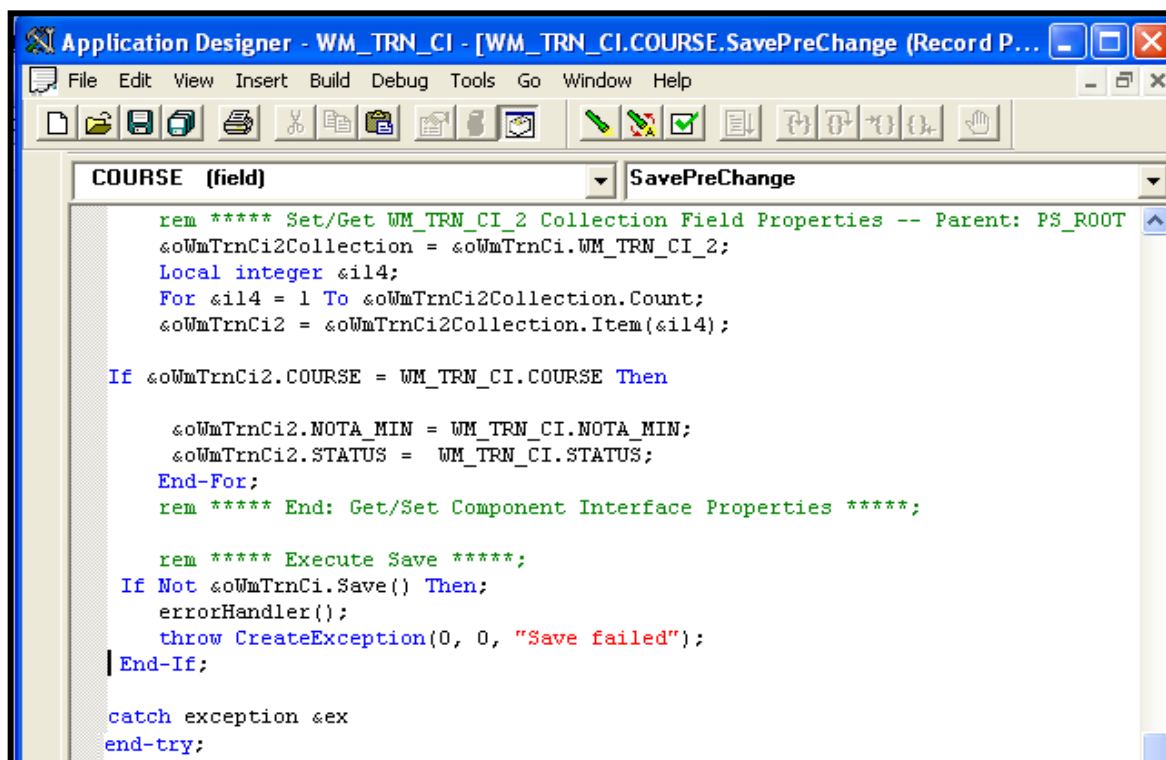
```
Application Designer - WM_TRN_CI - [WM_TRN_CI.COURSE.SavePreChange (Record P...
File Edit View Insert Build Debug Tools Go Window Help
COURSE (field) SavePreChange
rem ***** Set/Get WM_TRN_CI_2 Collection Field Properties -- Parent: PS_ROOT
&oWmTrnCi2Collection = &oWmTrnCi.WM_TRN_CI_2;
Local integer &i14;
For &i14 = 1 To &oWmTrnCi2Collection.Count;
&oWmTrnCi2 = &oWmTrnCi2Collection.Item(&i14);

&fileLog.WriteLine("&oWmTrnCi2.COURSE = " | &oWmTrnCi2.COURSE);
rem &oWmTrnCi2.COURSE = [*];
&fileLog.WriteLine("&oWmTrnCi2.EMPL_RCD = " | &oWmTrnCi2.EMPL_RCD);
rem &oWmTrnCi2.EMPL_RCD = [*];
&fileLog.WriteLine("&oWmTrnCi2.NOTA_MIN = " | &oWmTrnCi2.NOTA_MIN);
rem &oWmTrnCi2.NOTA_MIN = [*];
&fileLog.WriteLine("&oWmTrnCi2.STATUS = " | &oWmTrnCi2.STATUS);
rem &oWmTrnCi2.STATUS = [*];
End-For;
rem ***** End: Get/Set Component Interface Properties *****;

rem ***** Execute Save *****;
rem If Not &oWmTrnCi.Save() Then;
rem     errorHandler();
rem     throw CreateException(0, 0, "Save failed");
rem End-If;
```


Trabalhando com o Component Interface

Então, para cada linha, onde o course de uma página for igual ao da outra (De acordo com o objetivo do exemplo), o campo Nota_min e status é atualizado e salvo em seguida:



```
Application Designer - WM_TRN_CI - [WM_TRN_CI.COURSE.SavePreChange (Record P...
File Edit View Insert Build Debug Tools Go Window Help
COURSE (field) SavePreChange

rem ***** Set/Get WM_TRN_CI_2 Collection Field Properties -- Parent: PS_ROOT
&oWmTrnC2Collection = &oWmTrnC2.WM_TRN_CI_2;
Local integer &i14;
For &i14 = 1 To &oWmTrnC2Collection.Count;
&oWmTrnC2 = &oWmTrnC2Collection.Item(&i14);

If &oWmTrnC2.COURSE = WM_TRN_CI.COURSE Then

    &oWmTrnC2.NOTA_MIN = WM_TRN_CI.NOTA_MIN;
    &oWmTrnC2.STATUS = WM_TRN_CI.STATUS;
End-For;
rem ***** End: Get/Set Component Interface Properties *****;

rem ***** Execute Save *****;
If Not &oWmTrnC2.Save() Then;
    errorHandler();
    throw CreateException(0, 0, "Save failed");
End-If;

catch exception &ex
end-try;
```

Trabalhando com o Component Interface

As informações são atualizadas, de acordo com as alterações da página que utiliza o Component Interface. Por isso o nosso PeopleCode está inserido no evento SavePosChange.

Resultado

Estes são os valores inseridos na página do component mapeado:

The screenshot displays a web application interface for 'Componente 2'. It features a form titled 'Teste CI' with a navigation bar at the top containing links like 'Loc', 'Exibir Tudo', 'Primeiro', '1 de 6', and 'Último'. The form contains several input fields: 'ID Associação' with the value '1', 'Código do Curso' with the value '1', 'Nota Mínima' with the value '10,00', and 'Status' with a dropdown menu set to 'Ativo'. A red oval highlights the 'Nota Mínima' and 'Status' fields, and a red rectangle highlights the 'Código do Curso' field.

Resultado

Component 1

Código do Curso 1 Descrição

ID Funcionário	Nota Mínima	Status		
1	10,00	Ativo	+	-

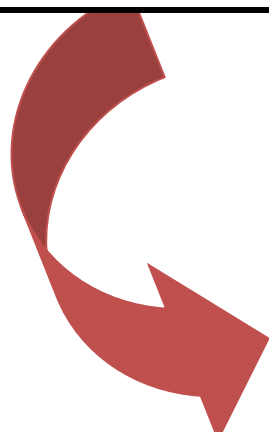
Ao mudar o valor do campo nota mínima na página que utiliza o Component Interface:

Component 1

Salvo

Código do Curso 1 Descrição

ID Funcionário	Nota Mínima	Status		
1 1	20,00	Ativo	+	-



Resultado

O Component é atualizado, de acordo com o emplid e curso.

Componente 2

ID Associação 1

Teste CI

[Loc](#) | [Exibir Tudo](#) | [Primeiro](#) | 1 de 6 | [Último](#)

Código do Curso

1

+

-

Nota Minima

20,00

Status

Ativo

▼

29





hqs