

Criação de parceiro utilizando a camada BOL

17/08/2015 10:00

Olá Zumbizada animada!

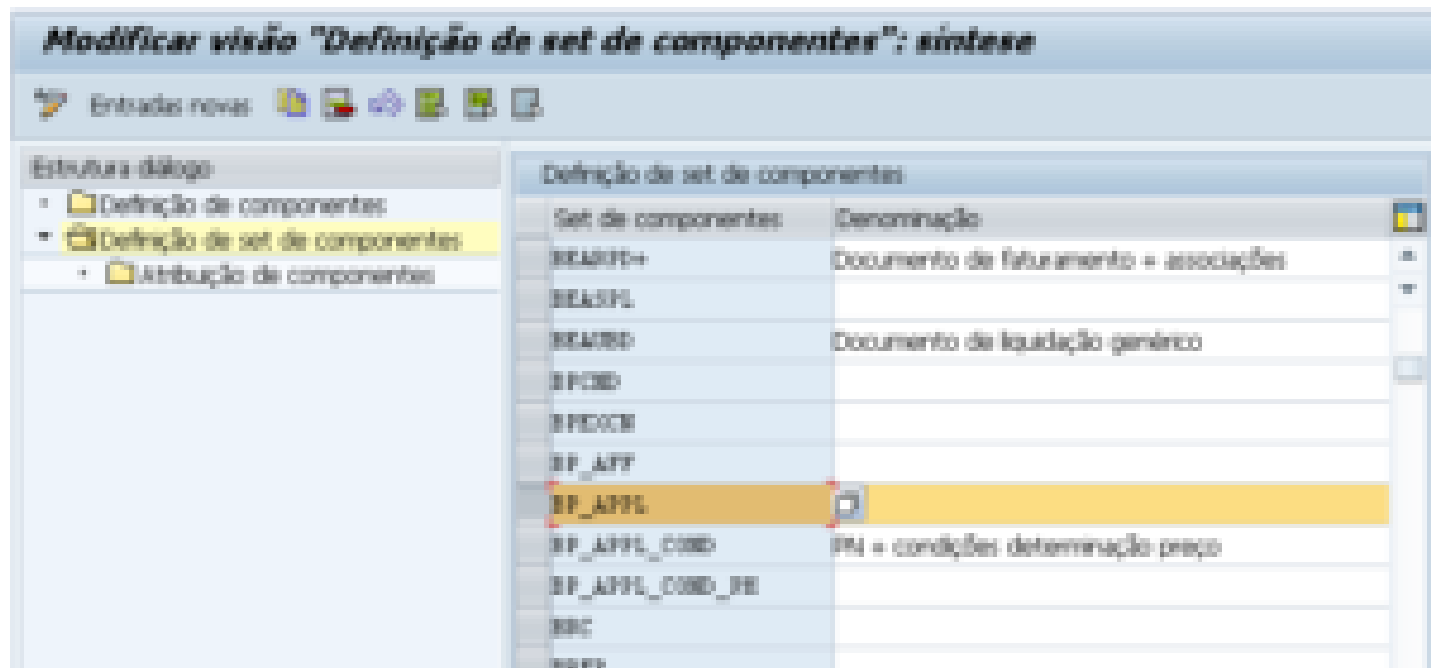
Quero mostrar um exemplo de como utilizar a camada BOL em um report, para isto vou mostrar o exemplo da criação de um Parceiro (Business Partner).

Se você tem alguma dúvida de como a camada BOL funciona, dá uma olhada no post [Relacionamento BOL](#).

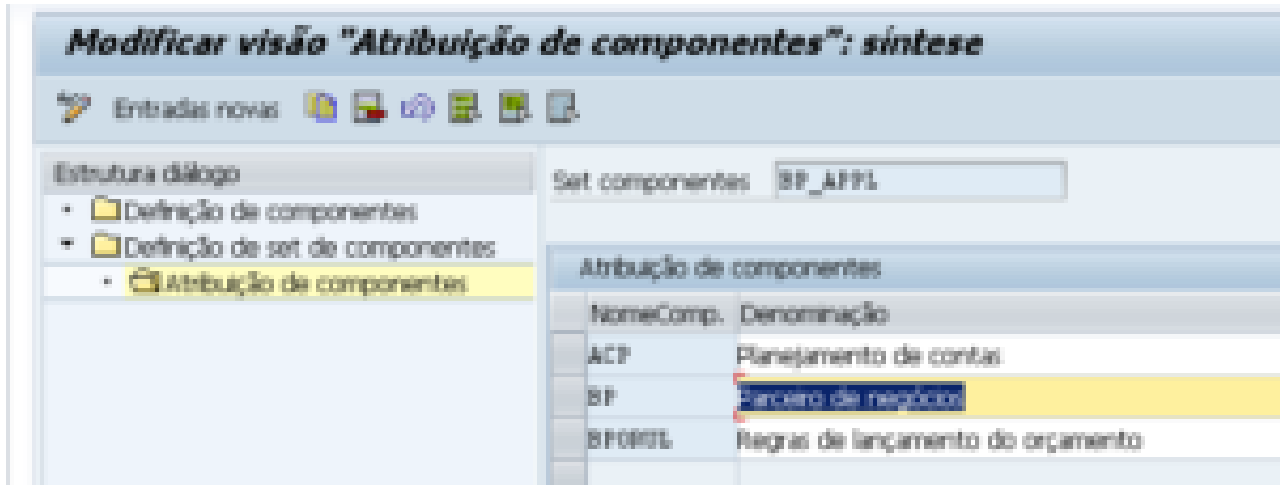
Primeiro precisamos pegar a instância da classe CL_CRM_BOL_CORE, é a classe responsável pelo controle da camada BOL e iniciar o set de componentes.

O set de componentes é o grupo de objetos BOL que pode ser utilizados durante uma execução. O set de componente pode ser visualizado no seguinte caminho:

SPRO->Customer Relationship Management->CRM: componentes válidos para todas as aplicações->Interaction Layer/Object Layer genérico->Configurações globais



Como vamos fazer a criação de um parceiro, podemos utilizar o BP_APPL



Dentro do BP_APPL tem a atribuição de Parceiro de Negócios.

Simbora... sem xurumelas...

A classe CL_CRM_BOL_CORE é uma classe singleton, precisamos pegar a instância e iniciar o set de componentes que iremos utilizar.

```
DATA: o_core TYPE REF TO cl_crm_bol_core.

o_core = cl_crm_bol_core=>get_instance( ).
o_core->start_up( 'BP_APPL' ).
```

Com isso a camada do BOL já pode ser utilizada.

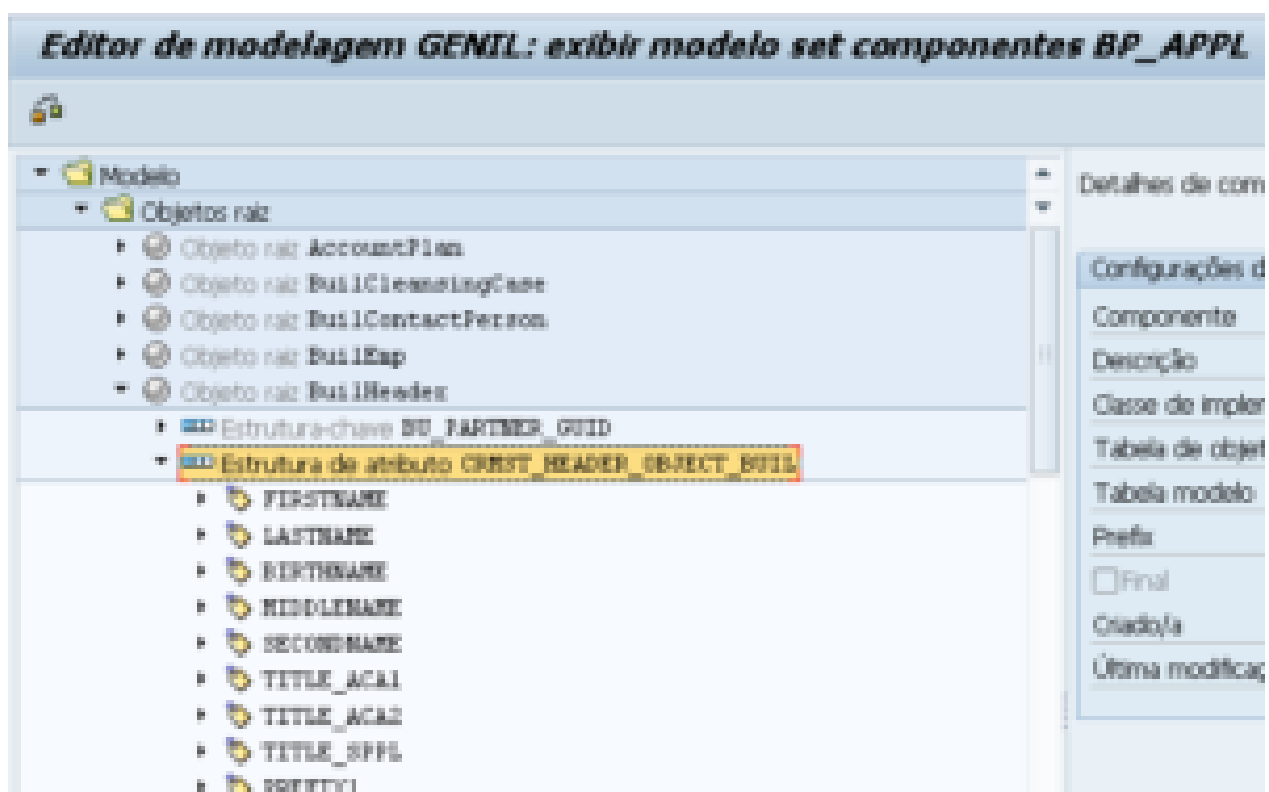
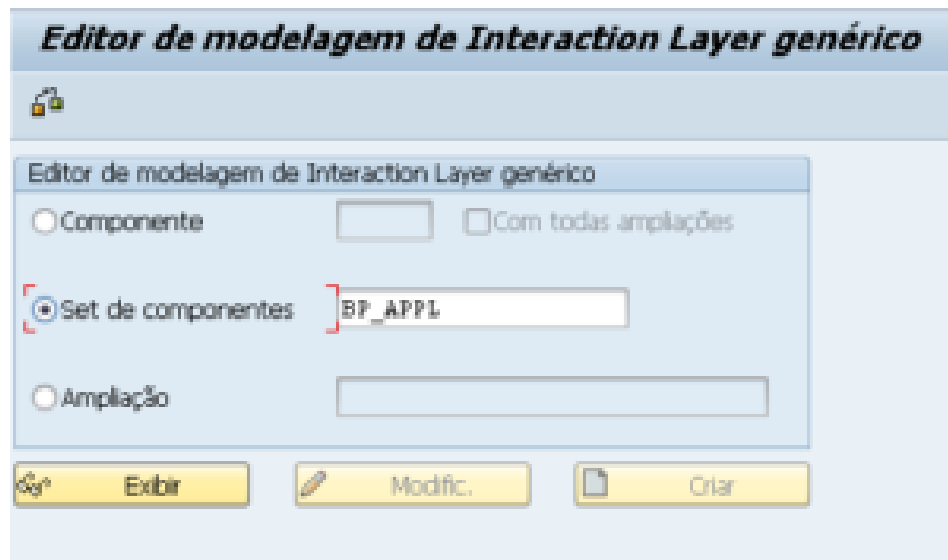
Podemos fazer a criação do objeto do Parceiro, que no BOL se chama BuilHeader.

A classe CL_CRM_BOL_ENTITY é responsável pela manipulação do objeto BOL e seus relacionamentos.

```
DATA: o_builheader TYPE REF TO cl_crm_bol_entity.

o_builheader = o_core->get_entity_factory( 'BuilHeader' )->create( it_params ).
```

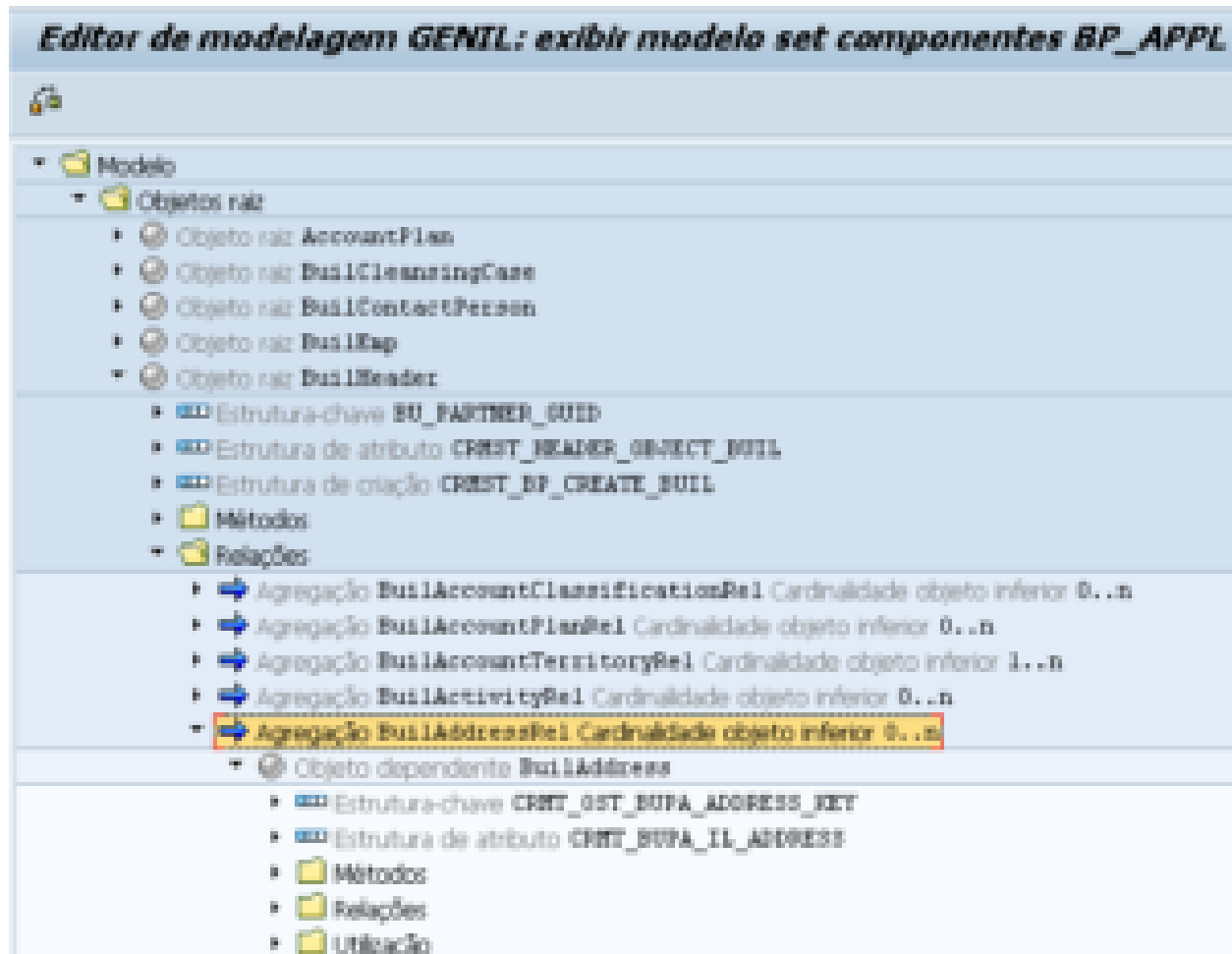
O it_params é uma tabela do tipo crmt_name_value_pair_tab, nesta tabela podemos passar alguns atributos com valores do nosso parceiro, os atributos possíveis podem ser visualizado na GENIL_MODEL_BROWSER.



Assim, criamos o cabeçalho do nosso Parceiro.

Com o [Relacionamento BOL](#) podemos fazer a criação dos outros objetos relacionados ao Parceiro.

Como por exemplo o Endereço, que utiliza o Relacionamento BuilAddressRel.



Após preencher todos os objetos, temos que chamar o método MODIFY da classe CL_CRM_BOL_CORE, neste momento as BAPIs serão executadas.

Porem, nada ainda sera gravado na base de dados, precisamos chamar o SAVE e o COMMIT do BOL, como seria se fosse feito com as chamadas das BAPIs.

Segue um exemplo:

```
REPORT ztestem_bp.
```

```
*****
```

```
*** Objeto (O_XX) ***
```

```
*****
```

```
DATA: o_builheader TYPE REF TO cl_crm_bol_entity,  
        o_endereco TYPE REF TO cl_crm_bol_entity,  
        o_email TYPE REF TO cl_crm_bol_entity,  
        o_role TYPE REF TO cl_crm_bol_entity,  
        o_tel TYPE REF TO cl_crm_bol_entity,  
        o_cel TYPE REF TO cl_crm_bol_entity,  
        o_note TYPE REF TO cl_crm_bol_entity,  
        o_core TYPE REF TO cl_crm_bol_core.
```

```
*****
```

```
*** Tabela Interna (IT_XX) ***
```

```
*****
```

```
DATA: it_params TYPE crmt_name_value_pair_tab.
```

```
*****
```

```
*** Work Area (WA_XX) ***
```

```

*****
DATA: wa_params TYPE crmt_name_value_pair.

*****
*** Variavel Local (VL_XX) ***
*****
DATA: vl_bp_number TYPE string.

"Tipo de pessoa
wa_params-name = 'BP_CATEGORY'.
wa_params-value = 1.
APPEND wa_params TO it_params.

"Tipo de pessoa
wa_params-name = 'FIRSTNAME'.
wa_params-value = 'bla'.
APPEND wa_params TO it_params.

"Tipo de pessoa
wa_params-name = 'LASTNAME'.
wa_params-value = 'silva'.
APPEND wa_params TO it_params.

"Pegar a instância do BOL
o_core = cl_crm_bol_core=>get_instance( ).
o_core->start_up( 'BP_APPL' ).

"Criação Cabeçalho
o_builheader = o_core->get_entity_factory( 'BuilHeader' )->create( it_params ).

"Criação da Role do tipo de Cliente
o_role = o_builheader->create_related_entity( 'BuilRolesRel' ).
o_role->set_property_as_string( iv_attr_name = 'PARTNERROLE'
                               iv_value     = 'CRM000'      ).

"Criação do Endereço
o_endereco = o_builheader->create_related_entity( 'BuilAddressRel' ).
o_endereco->set_property_as_string( iv_attr_name = 'POSTL_COD1'
                                   iv_value     = '99990-030'      ).
o_endereco->set_property_as_string( iv_attr_name = 'STREET'
                                   iv_value     = 'Rua do teste'    ).
o_endereco->set_property_as_string( iv_attr_name = 'HOUSE_NO'
                                   iv_value     = '123'             ).
o_endereco->set_property_as_string( iv_attr_name = 'COUNTRY'
                                   iv_value     = 'BR'              ).
o_endereco->set_property_as_string( iv_attr_name = 'TIME_ZONE'
                                   iv_value     = 'BRAZIL'          ).
o_endereco->set_property_as_string( iv_attr_name = 'STANDARDADDRESS'
                                   iv_value     = 'X'               ).

"Criação do E-mail vinculado ao endereço
o_email = o_endereco->create_related_entity( 'BuilAddressEMailRel' ).
o_email->set_property_as_string( iv_attr_name = 'E_MAIL'
                                iv_value     = 'dump@bol.com'     ).

"Criação do telefone vinculado ao endereço
o_tel = o_endereco->create_related_entity( 'BuilAddressPhoneRel' ).
o_tel->set_property_as_string( iv_attr_name = 'TELEPHONE'
                              iv_value     = '99999999999'      ).
o_tel->set_property_as_string( iv_attr_name = 'R_3_USER'
                              iv_value     = '1'                ).

"Criação do celular vinculado ao endereço
o_cel = o_endereco->create_related_entity( 'BuilAddressPhoneRel' ).

```

```

o_cel->set_property_as_string( iv_attr_name = 'TELEPHONE'
                               iv_value     = '8888888888' ).
o_cel->set_property_as_string( iv_attr_name = 'R_3_USER'
                               iv_value     = '3' ).

"Criação de uma nota de texto (Observação)
o_note ?= o_builheader->create_related_entity( iv_relation_name = 'BuilNotesRel' ).

o_note->set_property_as_string( iv_attr_name = 'TEXT_ID'
                               iv_value     = '0001' ).
o_note->set_property_as_string( iv_attr_name = 'LANGU'
                               iv_value     = 'PT' ).
o_note->set_property_as_string( iv_attr_name = 'CONC_FORMATTED_LINES'
                               iv_value     = 'teste texto teste' ).

"Gerar e gravar o Cliente
o_core->modify( ).
o_core->get_transaction( )->save( ).
o_core->get_transaction( )->commit( ).

"Após a gravação, exibir em tela o número do Parceiro gerado
vl_bp_number = o_builheader->get_property_as_string( 'BP_NUMBER' ).

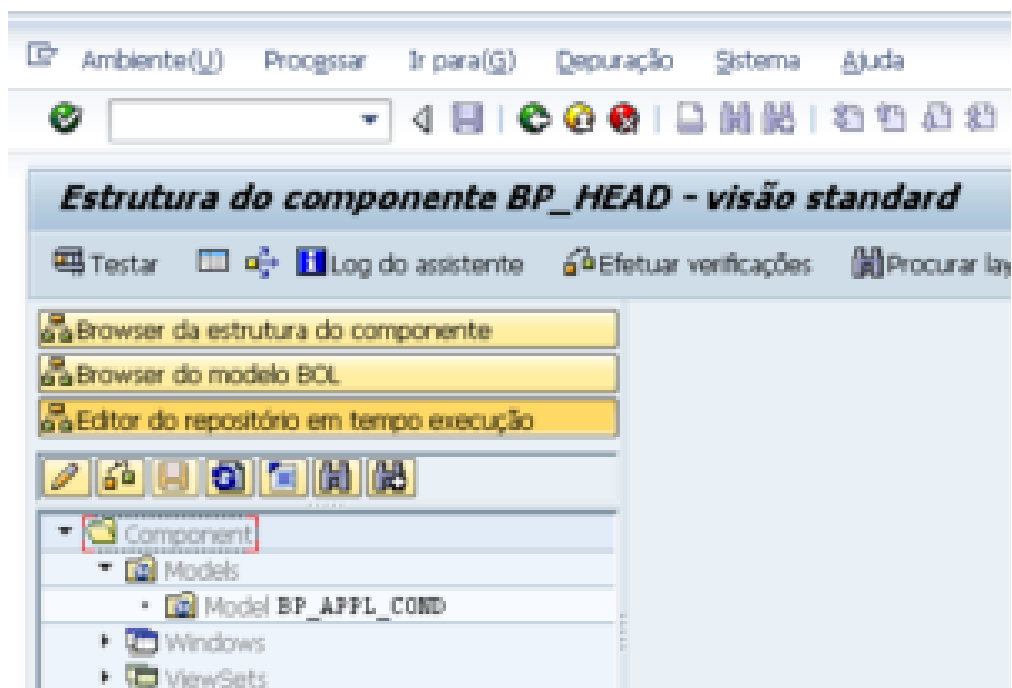
WRITE vl_bp_number.

```

O cliente pode ser visto na transação BP.

Porque quando fazemos o desenvolvimento via componente de CRM não precisamos pegar a instancia da CL_CRM_BOL_CORE e nem passar o set de componentes?

Na verdade quando fazemos um componente baseado em BOL, temos que fazer a configuração do *Models*, neste momento informamos qual sera o set de componentes, assim, o standard sabe quais são os objetos que poderão ser utilizados.



Abraços a todos!

Comentários

Aurelio — 05/10/2020 14:58

Mauro, estamos criando um cliente com a BAPI `bapi_bupa_create_from_data` mas nao estamos conseguindo preencher o campo `icmstaxpay`, tem alguma dica?

Ambiente S4 Hana.

Transação BP

William Favarin — 19/08/2015 08:53

Parabéns!

Muito bacana o artigo.

Bacana mostrar as formas que CRM trabalha, diferente de chamar as BAPI's igual no ECC.