

Criação de Novas Regras de Importação

Produto	:	TOTVS SST® – Import – 6.3.27.14			
Chamado	:	TEVMOB	Data da publicação	:	03/04/14
País(es)	:	<i>Brasil</i>	Banco(s) de Dados	:	<i>Oracle</i>
Cliente	:	General Motors			

Importante

Esta melhoria depende de execução conforme **Procedimentos para Implementação**.

Criação de novas regras de importação para a GM, incorporadas através da configuração de IdentType = 10 na cfg01. Para impedir a criação automática de setores/funções que não estão na base PersonalMed e não atualizar estes dados de funcionários se não estiverem cadastrados.

Procedimento para Implementação

Importante

Antes de executar a atualização é recomendável realizar o backup do banco de dados bem como dos arquivos do Sistema(executáveis, dlls e arquivos de configuração):

Realizar a atualização antes no ambiente de homologação e, posterior a devida validação, no ambiente de produção.

Execute o script que segue anexo, na base de dados de **homologação** do Personal Med.

OBS: o script gera um log em C:\, sendo assim, solicitamos que após a atualização, o log seja encaminhado para análise.

Faça o download dos aplicativos e dll's que constam nos links em anexo e descompacte-os no diretório do Personal Med.

OBS: os aplicativos e dll's devem ser atualizados em todos os micros que possuem o Personal Med instalado.

Atenção

Recomendamos que o chamado relacionado a esse boletim seja encerrado após a homologação.

Atualizações do update/compatibilizador

1. Alteração de **Campo** nas Tabelas.

Operação	Local/Arquivo	Versão
Alteração	pmwe\Pmed\Manutencao\Importacao\lmpObjects.pas	-
Alteração	pmwe\PMed\Manutencao\Importacao\frm_Main.pas	-
Alteração	pmwe\PMed\Manutencao\Importacao\dtm_Importacao.pas	-
Alteração	pmwe_libs\PersonalMessages.pas	-
Alteração	pmwe\PMed\PPP2\dtm_PPPRiscos.pas	-

Procedimento para Configuração

1. Executar scripts:

1. Limpeza de duplicados na base:

a. Excluir duplicados no PPP: **Atualizacao_Duplicados_PPP_GM.sql**

```

DECLARE
  PppCodBase numeric(9);
  nome varchar(120);
  tabela varchar(20);
BEGIN
  FOR DUPLICADOS IN (select clini_00.name, ppp001.Pcod, count(ppp001.pcod) TotReg from ppp001, clini_00 Where clini_00.pcod =
  ppp001.pcod group by
    clini_00.name, ppp001.pcod having count(ppp001.pcod) > 1 order by clini_00.name) LOOP
    nome := DUPLICADOS.NAME;
    tabela := 'Nada';
    PppCodBase := 0;
    select min(PPPCOD) into PppCodBase from PPP001 WHERE PPP001.PCOD = DUPLICADOS.PCOD;
    --Dbms_Output.put_line('PPPCOD: ' || PppCodBase || ' da Tabela: PPP009 DO FUNCIONÁRIO');
    IF PppCodBase <> 0 THEN
      FOR FUNCIONARIO IN (SELECT PPP001.PPPCOD FROM PPP001 WHERE PPP001.PCOD = DUPLICADOS.PCOD ORDER BY
      PPP001.PPPCOD) LOOP
        IF FUNCIONARIO.PPPCOD > PppCodBase THEN
          FOR P002 IN (SELECT PPPCOD FROM PPP002 WHERE PPP002.PPPCOD = FUNCIONARIO.PPPCOD) LOOP
            EXECUTE IMMEDIATE 'UPDATE PPP002 SET PPP002.PPPCOD = ' || PppCodBase || ' WHERE PPP002.PPPCOD = ' ||
            P002.PPPCOD;
            COMMIT;
            tabela := 'PPP002';
          END LOOP;
          FOR P003 IN (SELECT PPPCOD FROM PPP003 WHERE PPP003.PPPCOD = FUNCIONARIO.PPPCOD) LOOP
            EXECUTE IMMEDIATE 'UPDATE PPP003 SET PPP003.PPPCOD = ' || PppCodBase || ' WHERE PPP003.PPPCOD = ' ||
            P003.PPPCOD;
            COMMIT;
            tabela := 'PPP003';
          END LOOP;
          FOR P004 IN (SELECT PPPCOD FROM PPP004 WHERE PPP004.PPPCOD = FUNCIONARIO.PPPCOD) LOOP
            EXECUTE IMMEDIATE 'UPDATE PPP004 SET PPP004.PPPCOD = ' || PppCodBase || ' WHERE PPP004.PPPCOD = ' ||
            P004.PPPCOD;
            COMMIT;
            tabela := 'PPP004';
          END LOOP;
          FOR P005 IN (SELECT PPPCOD FROM PPP005 WHERE PPP005.PPPCOD = FUNCIONARIO.PPPCOD) LOOP
            EXECUTE IMMEDIATE 'UPDATE PPP005 SET PPP005.PPPCOD = ' || PppCodBase || ' WHERE PPP005.PPPCOD = ' ||
            P005.PPPCOD;
            COMMIT;
            tabela := 'PPP005';
          END LOOP;
          FOR P006 IN (SELECT PPPCOD FROM PPP006 WHERE PPP006.PPPCOD = FUNCIONARIO.PPPCOD) LOOP
            EXECUTE IMMEDIATE 'UPDATE PPP006 SET PPP006.PPPCOD = ' || PppCodBase || ' WHERE PPP006.PPPCOD = ' ||
            P006.PPPCOD;
            COMMIT;
            tabela := 'PPP006';
          END LOOP;
          FOR P007 IN (SELECT PPPCOD FROM PPP007 WHERE PPP007.PPPCOD = FUNCIONARIO.PPPCOD) LOOP
            EXECUTE IMMEDIATE 'UPDATE PPP007 SET PPP007.PPPCOD = ' || PppCodBase || ' WHERE PPP007.PPPCOD = ' ||
            P007.PPPCOD;
            COMMIT;
            tabela := 'PPP007';
          END LOOP;
          FOR P008 IN (SELECT PPPCOD FROM PPP008 WHERE PPP008.PPPCOD = FUNCIONARIO.PPPCOD) LOOP
            EXECUTE IMMEDIATE 'UPDATE PPP008 SET PPP008.PPPCOD = ' || PppCodBase || ' WHERE PPP008.PPPCOD = ' ||
            P008.PPPCOD;
            COMMIT;
          END LOOP;
        END LOOP;
      END LOOP;
    END LOOP;
  END LOOP;

```

```

        tabela := 'PPP008';
    END LOOP;

    FOR P009 IN (SELECT PPPCOD FROM PPP009 WHERE PPP009.PPPCOD = FUNCIONARIO.PPPCOD) LOOP
        EXECUTE IMMEDIATE 'UPDATE PPP009 SET PPP009.PPPCOD = ' || PppCodBase || ' WHERE PPP009.PPPCOD = ' ||
P009.PPPCOD;
        COMMIT;
        tabela := 'PPP009';
    END LOOP;

    FOR P011 IN (SELECT PPPCOD FROM PPP011 WHERE PPP011.PPPCOD = FUNCIONARIO.PPPCOD) LOOP
        EXECUTE IMMEDIATE 'UPDATE PPP011 SET PPP011.PPPCOD = ' || PppCodBase || ' WHERE PPP011.PPPCOD = ' ||
P011.PPPCOD;
        COMMIT;
        tabela := 'PPP011';
    END LOOP;
    FOR P013 IN (SELECT PPPCOD FROM PPP013 WHERE PPP013.PPPCOD = FUNCIONARIO.PPPCOD) LOOP
        EXECUTE IMMEDIATE 'UPDATE PPP013 SET PPP013.PPPCOD = ' || PppCodBase || ' WHERE PPP013.PPPCOD = ' ||
P013.PPPCOD;
        COMMIT;
        tabela := 'PPP013';
    END LOOP;

    EXECUTE IMMEDIATE 'DELETE FROM PPP001 WHERE PPP001.PPPCOD = ' || FUNCIONARIO.PPPCOD;
    COMMIT;
    tabela := 'PPP001';

    END IF;
    END LOOP;
    END IF;
    END LOOP;
EXCEPTION
    WHEN OTHERS THEN
        Dbms_Output.put_line('Erro ao atualizar PPPCOD: ' || PppCodBase || ' da Tabela: ' || tabela || ' DO FUNCIONÁRIO ' || nome);
    -- ROLLBACK;
END;
```

b. Correcao_Exames_Laboratoriais_Duplicados_GM.sql

```

/*
PCOD Origem: 105064, PCOD Destino: 105048. Existe(m) 16 resultado(s) semelhante(s) na aleta Exames Laboratoriais. Estas fichas não
foram unificadas, verifique.
PCOD Origem: 105662, PCOD Destino: 103338. Existe(m) 5 resultado(s) semelhante(s) na aleta Exames Laboratoriais. Estas fichas não
foram unificadas, verifique.
PCOD Origem: 101552, PCOD Destino: 101551. Existe(m) 3 resultado(s) semelhante(s) na aleta Exames Laboratoriais. Estas fichas não
foram unificadas, verifique.
PCOD Origem: 101580, PCOD Destino: 101213. Existe(m) 2 resultado(s) semelhante(s) na aleta Exames Laboratoriais. Estas fichas não
foram unificadas, verifique.
PCOD Origem: 106935, PCOD Destino: 106934. Existe(m) 1 resultado(s) semelhante(s) na aleta Exames Laboratoriais. Estas fichas não
foram unificadas, verifique.
PCOD Origem: 105065, PCOD Destino: 105054. Existe(m) 12 resultado(s) semelhante(s) na aleta Exames Laboratoriais. Estas fichas não
foram unificadas, verifique.
PCOD Origem: 104526, PCOD Destino: 104493. Existe(m) 13 resultado(s) semelhante(s) na aleta Exames Laboratoriais. Estas fichas não
foram unificadas, verifique.
PCOD Origem: 70090, PCOD Destino: 69973 . Existe(m) 2 resultado(s) semelhante(s) na aleta Exames Laboratoriais. Estas fichas não foram
unificadas, verifique.
PCOD Origem: 105574, PCOD Destino: 105505. Existe(m) 13 resultado(s) semelhante(s) na aleta Exames Laboratoriais. Estas fichas não
foram unificadas, verifique.
PCOD Origem: 104884, PCOD Destino: 104883. Existe(m) 1 resultado(s) semelhante(s) na aleta Exames Laboratoriais. Estas fichas não
foram unificadas, verifique.
PCOD Origem: 105063, PCOD Destino: 105052. Existe(m) 13 resultado(s) semelhante(s) na aleta Exames Laboratoriais. Estas fichas não
foram unificadas, verifique.
PCOD Origem: 110229, PCOD Destino: 106531. Existe(m) 4 resultado(s) semelhante(s) na aleta Exames Laboratoriais. Estas fichas não
foram unificadas, verifique.
```

PCOD Origem: 93397, PCOD Destino: 70477 . Existe(m) 14 resultado(s) semelhante(s) na aleta Exames Laboratoriais. Estas fichas não foram unificadas, verifique.

PCOD Origem: 101637, PCOD Destino: 101523. Existe(m) 2 resultado(s) semelhante(s) na aleta Exames Laboratoriais. Estas fichas não foram unificadas, verifique.

PCOD Origem: 104853, PCOD Destino: 104852. Existe(m) 1 resultado(s) semelhante(s) na aleta Exames Laboratoriais. Estas fichas não foram unificadas, verifique.

*/

/*

Transferência de exames laboratoriais para funcionários duplicados

*/

DECLARE

TYPE OrigemPcod IS TABLE OF INTEGER(9);

TYPE DestinoPcod IS TABLE OF INTEGER(9);

Destinos DestinoPcod :=

DestinoPcod(105048,103338,101551,101213,106934,105054,104493,69973,105505,104883,105052,106531,70477,101523,104852);

Origens OrigemPcod :=

OrigemPcod(105064,105662,101552,101580,106935,105065,104526,70090,105574,104884,105063,110229,93397,101637,104853);

TotReg numeric(9);

BEGIN

FOR i IN Origens.First .. Origens.Last

LOOP

FOR Origem IN (SELECT * FROM CLINI_13 C13 WHERE C13.PCOD = Origens(i)) loop

SELECT COUNT(CLINI_13.PCOD) INTO TOTREG FROM CLINI_13 WHERE PCOD = Destinos(i) AND EXAMDATE =

Origem.Examdate AND LABCOD = Origem.Labcod

AND LEXAMCOD = Origem.LEXAMCOD;

IF TOTREG = 0 THEN

EXECUTE IMMEDIATE 'UPDATE CLINI_13 SET PCOD = ' || Origem.Pcod || ' WHERE PCOD = ' || Origens(i) ||

' AND LABCOD = ' || Origem.Labcod || ' AND LEXAMCOD = ' || Origem.Lexamcod;

COMMIT;

END IF;

END LOOP;

END LOOP;

FOR i IN Origens.First .. Origens.Last

LOOP

FOR Origem IN (SELECT * FROM CLINI_13 C13 WHERE C13.PCOD = Origens(i)) loop

EXECUTE IMMEDIATE 'DELETE FROM CLINI_13 WHERE PCOD = ' || Origem.Pcod;

COMMIT;

END LOOP;

END LOOP;

END;

2. Alterar o SQL duplicados.sql na pasta do PersonalMed pelo sql de duplicados_GM.sql

```
select ckeep.pcod pcoddestino,
```

```
cdel.pcod pcodorigem
```

```
from clini_00 ckeep,
```

```
clini_00 cdel
```

```
where ((ckeep.registration is not null
```

```
and cdel.registration is not null
```

```
and ckeep.pcod < cdel.pcod
```

```
and substr(ckeeep.registration, 1, 1) = substr(cdel.registration, 1, 1)
```

```
and substr(ckeeep.registration, 3, 8) = substr(cdel.registration, 3, 8))
```

```
or
```

```
((ckeep.registration is null and ckeep.status = 2) and
```

```
(cdel.registration is null and cdel.status = 2) and
```

```
(ckeep.Financeregister = cdel.Financeregister) and
```

```
(ckeep.pcod < cdel.pcod))
```

```
)
```

```
order by ckeep.pcod
```

3. Executar duplicados.exe
4. Executar Import.exe

Procedimento para Utilização

1. No módulo **Import** executar importação.

Novas regras = IdentType = 10 (tabela Cfg01)

O objetivo da atividade é customizar o comportamento da interface que carrega as informações do PeopleSoft para o PersonalMed, a fim de evitar que sejam carregados dados inconsistentes ou duplicados.

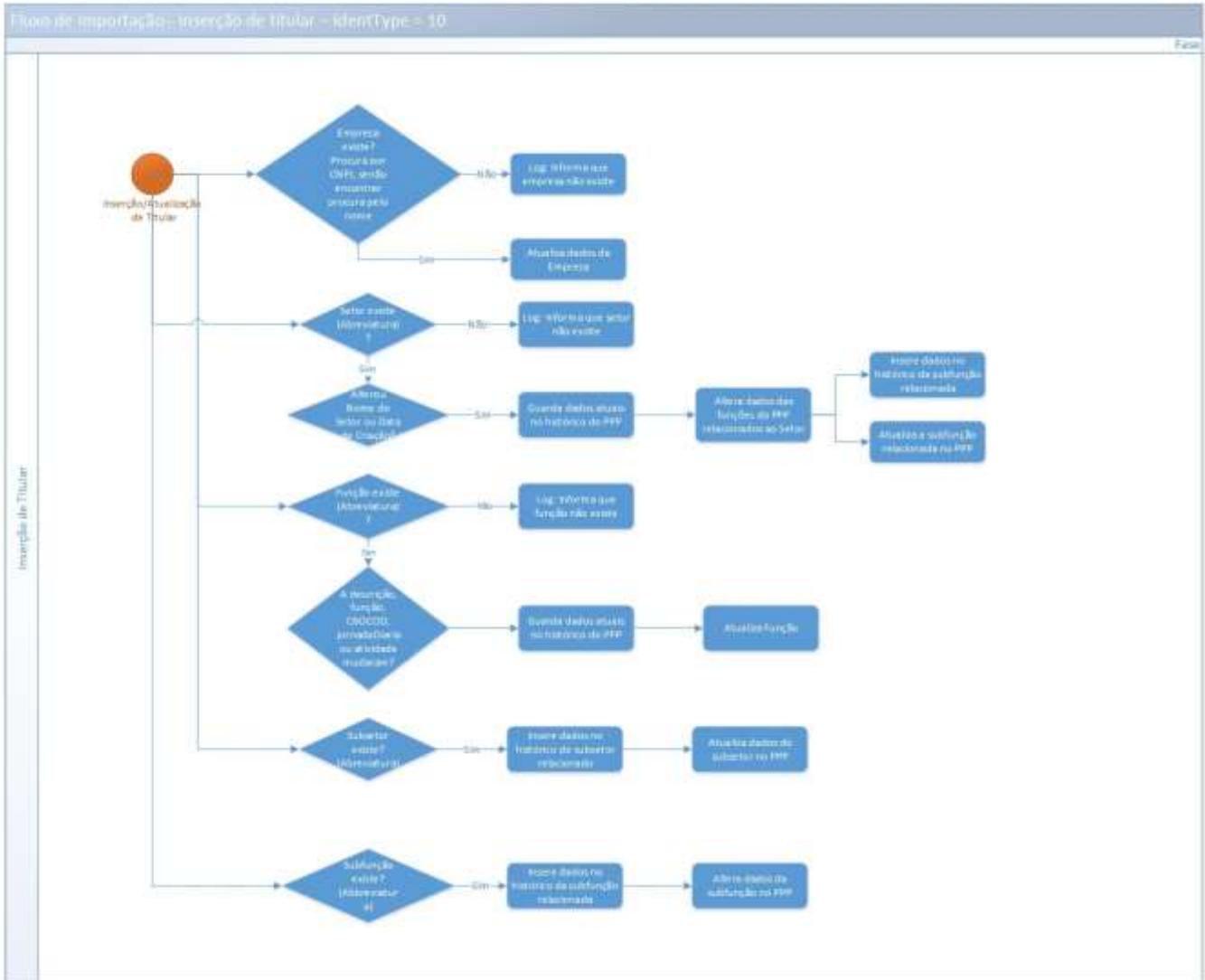
1. Não permitir a criação de novas empresas, setores e funções, barrar todas as diferenças dos registros e guardar num relatório de inconsistências para análise da área responsável na GM.
O novo procedimento definido pela GM é que a área responsável deverá incluir todas as empresas, setores e funções manualmente no sistema (Como fazem no Peoplesoft e no Mainframe).

As inconsistências deverão ser tratadas no PeopleSoft e a GM terá de executar, novamente, a interface para que o ajuste se torne válido.

Definições:

- 1.1 Se Código do Setor diferente de existente na tabela de Setores, branco ou nulo
Inclui dados pessoais do funcionário
Não cria novo setor
Inclui informação de Setor no relatório de inconsistências (Matrícula Funcionário, Código, Abreviatura e Descrição)
- 1.2 Se Abreviatura do Setor diferente de existente na tabela de Setores, branco ou nulo
Inclui dados pessoais do funcionário
Não cria novo setor
Inclui informação de Setor no relatório de inconsistências (Matrícula Funcionário, Código, Abreviatura e Descrição)
- 1.3 Se Descrição do Setor diferente de existente na tabela de Setores, branco ou nulo
Inclui dados pessoais do funcionário
Não cria novo setor
Inclui informação de Setor no relatório de inconsistências (Matrícula Funcionário, Código, Abreviatura e Descrição)
- 1.4 Se Código da Função diferente de existente na tabela de Setores, branco ou nulo
Inclui dados pessoais do funcionário
Não cria novo setor
Inclui informação de Setor no relatório de inconsistências (Matrícula Funcionário, Código, Abreviatura e Descrição)
- 1.5 Se Abreviatura da Função diferente de existente na tabela de Setores, branco ou nulo
Inclui dados pessoais do funcionário
Não cria novo setor
Inclui informação de Setor no relatório de inconsistências (Matrícula Funcionário, Código, Abreviatura e Descrição)
- 1.6 Se Descrição da Função diferente de existente na tabela de Setores, branco ou nulo
Inclui dados pessoais do funcionário
Não cria novo setor
Inclui informação de Setor no relatório de inconsistências (Matrícula Funcionário, Código, Abreviatura e Descrição)
- 1.7 Se Empresa diferente de existente na tabela de Empresas, branco ou nulo
Inclui dados pessoais do funcionário
Não cria nova Empresa
Inclui informação de Empresa no relatório de inconsistências (Matrícula Funcionário, Código e Descrição)

Fluxo do Processo



Informações Técnicas

Tabelas Utilizadas	-
Funções Envolvidas	Import.exe
Sistemas Operacionais	Windows®