

Configuração de cores e tamanhos em PHC por campos de utilizador

Português

Antes de se iniciar a configuração do driver, devem ser criados dois novos Stored Procedures.

- SPMSS_AlterFromACL

```
CREATE PROCEDURE [dbo].[SPMSS_AlterFromACL]
(
    @DCLEXR VARCHAR(30),
    @DCLTPD VARCHAR(10),
    @DCLSER VARCHAR(4),
    @DCLNDC INT,
    @DCLLIN INT,
    @Source VARCHAR(2000),
    @Rank INT
)
AS
BEGIN
    DECLARE @Pos INT
    DECLARE @SepCount INT
    DECLARE @PosStart INT
    DECLARE @PosEnd INT
    SET @Pos = 1
    SET @SepCount = 0
    SET @PosStart = 1
    SET @PosEnd = 0

    WHILE @Pos <= LEN(@Source)
    BEGIN
        IF SUBSTRING(@Source, @Pos, 1) = CHAR(7)
        BEGIN
            SET @SepCount = @SepCount + 1
            IF @Rank > @SepCount
                SET @PosStart = @Pos + 1
        END
        IF @SepCount = @Rank
        BEGIN
            SET @PosEnd = @Pos
            BREAK
        END
        SET @Pos = @Pos + 1
    END
    IF @posend = 0
        SET @posend = LEN(@Source)+1

    UPDATE MSDCL SET DCLACL = (SELECT STUFF(DCL.DCLACL, @PosStart, 1, 'S')
FROM MSDCL DCL (NOLOCK)
    WHERE DCL.DCLEXR = MSDCL.DCLEXR AND DCL.DCLTPD = MSDCL.DCLTPD AND
    DCL.DCLSER = MSDCL.DCLSER AND DCL.DCLNDC = MSDCL.DCLNDC AND DCL.DCLLIN =
    MSDCL.DCLLIN)
    WHERE MSDCL.DCLEXR = @DCLEXR AND MSDCL.DCLTPD = @DCLTPD AND
    MSDCL.DCLSER = @DCLSER AND MSDCL.DCLNDC = @DCLNDC AND MSDCL.DCLLIN = @DCLLIN
END
GO
```

- SPMSS_SubsRefCorTamanho

```
CREATE PROCEDURE [dbo].[SPMSS_SubsRefCorTamanho]
AS
BEGIN
    SET NOCOUNT ON;

    DECLARE @ref VARCHAR(60)
    DECLARE @DCLEXR VARCHAR(30)
    DECLARE @DCLTPD VARCHAR(10)
    DECLARE @DCLSER VARCHAR(4)
    DECLARE @DCLNDC INT
    DECLARE @DCLLIN INT
    DECLARE @DCLACL VARCHAR(2000)

    DECLARE @refCorTam VARCHAR(60)
    DECLARE @RefFinal VARCHAR(60)
    DECLARE @DescFinal VARCHAR(100)

    DECLARE @ErrorMessage NVARCHAR(4000)
    DECLARE @ErrorSeverity INT
    DECLARE @ErrorState INT

    DECLARE curRef CURSOR FOR
    SELECT ISNULL(dclart, ''), DCLEXR, DCLTPD, DCLSER, DCLNDC, DCLLIN,
    DCLACL FROM MSDCL(nolock) WHERE DCLSYNCR = 'N'
    --GROUP BY dclart
    OPEN curRef

    FETCH NEXT FROM curRef INTO
    @ref,@DCLEXR,@DCLTPD,@DCLSER,@DCLNDC,@DCLLIN,@DCLACL

    WHILE @@FETCH_STATUS = 0 BEGIN
        BEGIN TRY

            SET @RefFinal = ''
            SET @RefFinal = (SELECT ISNULL(ST.ref, '') FROM ST
            WHERE (RTRIM(ST.u_refpai) + '.' + RTRIM(ST.u_refcor) + '.' +
            RTRIM(ST.u_reftam)) = @ref)
            SET @DescFinal = (SELECT ISNULL(ST.design, '') FROM
            ST WHERE (RTRIM(ST.u_refpai) + '.' + RTRIM(ST.u_refcor) + '.' +
            RTRIM(ST.u_reftam)) = @ref)

            IF ((@RefFinal <> '') AND (@ref <> @RefFinal))
                UPDATE MSDCL SET dclart = @RefFinal, DCLDSA =
            @DescFinal WHERE DCLEXR = @DCLEXR AND DCLTPD = @DCLTPD AND DCLSER = @DCLSER
            AND DCLNDC = @DCLNDC AND DCLLIN = @DCLLIN AND DCLART = @ref AND DCLSYNCR =
            'N'

            -- Verifica se é um artigo com cor e tamanho
            SET @refCorTam = dbo.ExtractFromACL(@DCLACL, 15)
            IF (@refCorTam = 'F')
                EXEC SPMSS_AlterFromACL
            @DCLEXR,@DCLTPD,@DCLSER,@DCLNDC,@DCLLIN,@DCLACL,15

        END TRY
        BEGIN CATCH
            SELECT @ErrorMessage = ERROR_MESSAGE(),
            @ErrorSeverity = ERROR_SEVERITY(), @ErrorState = ERROR_STATE()
        END CATCH
    END WHILE
END
```

```
                                RAISERROR (@ErrorMessage, @ErrorSeverity,  
@ErrorState)  
  
                                GOTO FIM  
                                END CATCH  
  
                                FETCH NEXT FROM curRef INTO  
@ref,@DCLEXR,@DCLTPD,@DCLSER,@DCLNDC,@DCLLIN,@DCLACL  
                                END  
                                FIM:  
                                CLOSE curRef  
                                DEALLOCATE curRef  
END  
GO
```

Configuração do Driver

Após a criação dos Stored Procedures anteriores, vai ser necessário efetuar as seguintes configurações no driver:

- Tabela Artigos (ART):

- SQL 1 (Artigos que não utilizam grelhas)

Mapeamentos

Alterar o mapeamento dos campos em baixo.

Campo	Descrição	Tabela ERP	Camp ERP
ARTTRD	Tipo Recolha Dados	%DEFAULT%	'S'
ARTTGR	Tem grelha	%DEFAULT%	'N'

Filtros

Acrescentar o filtro de baixo aos existentes.

and st.u_refpai = "

- SQL 2 (Todos os Artigos com cores e tamanhos)

Mapeamentos

Duplicar o SQL 1 e alterar o mapeamento dos seguintes campos.

Campo	Descrição	Tabela ERP	Camp ERP
ARTCOD	Código	%DEFAULT%	RTRIM(st.u_refpai) + '.' + RTRIM(st.u_refcor) + '.' + RTRIM(st.u_reftam)
ARTDSC	Descrição	%OPTIONS%	<u>Campo a validar:</u> st.u_refpai <u>Se igual a...:</u> " Então: st.design <u>Senão:</u> st.u_descpai
ARTTRD	Tipo de recolha de dados	%DEFAULT%	'F'
ARTTGR	Tem grelha	%DEFAULT%	'N'

Filtros

Alterar o filtro:

and st.u_refpai = "

Para:

and st.u_refpai <> "

- SQL 3 (Artigos pai)

Mapeamentos

Duplicar o SQL 2 e alterar o mapeamento dos seguintes campos.

Campo	Descrição	Tabela ERP	Camp ERP
ARTCOD	Código	%DEFAULT%	RTRIM(st.u_refpai)
ARTDSC	Descrição	st	u_descpai
ARTTRD	Tipo de recolha de dados	%DEFAULT%	'G'
ARTTGR	Tem grelha	%DEFAULT%	'S'

Filtros

Acrescentar aos filtros existentes:

and st.ref = (select top 1 st2.ref from st st2 (NOLOCK) where st.u_refpai = st2.u_refpai)

- Tabela Actualização de Stock (ARTEX1):

- SQL 1 (Artigos que não utilizam grelhas)

Filtros

Acrescentar o filtro de baixo aos existentes.

and st.u_refpai = "

- SQL 2 (Todos os Artigos com cores e tamanhos)

Mapeamentos

Duplicar o SQL 1 e alterar o mapeamento dos seguintes campos.

Campo	Descrição	Tabela ERP	Camp ERP
ARTCOD	Código	%DEFAULT%	RTRIM(st.u_refpai) + '.' + RTRIM(st.u_refcor) + '.' + RTRIM(st.u_reftam)

Filtros

Alterar o filtro:

and st.u_refpai = "

Para:

and st.u_refpai <> "

- Grelha horizontal (GRH):

- SQL 1

Inserir o código sql de baixo no separador SQL em modo avançado:

```
DECLARE @POSICAO INT
DECLARE @SQL VARCHAR(MAX)
DECLARE @REF VARCHAR(60)
DECLARE @REFT VARCHAR(60)
DECLARE @TAM VARCHAR(25)
DECLARE @MSGRH TABLE (
    REF VARCHAR(60) not null,
    DESCR VARCHAR(100) NOT NULL,
    DESCRV VARCHAR(100) NOT NULL,
    CC1 VARCHAR (30) NOT NULL,
    DC1 VARCHAR (100) NOT NULL,
    CC2 VARCHAR (30) NOT NULL,
    DC2 VARCHAR (100) NOT NULL,
    CC3 VARCHAR (30) NOT NULL,
    DC3 VARCHAR (100) NOT NULL,
    CC4 VARCHAR (30) NOT NULL,
    DC4 VARCHAR (100) NOT NULL,
    CC5 VARCHAR (30) NOT NULL,
    DC5 VARCHAR (100) NOT NULL,
    CC6 VARCHAR (30) NOT NULL,
    DC6 VARCHAR (100) NOT NULL,
    CC7 VARCHAR (30) NOT NULL,
    DC7 VARCHAR (100) NOT NULL,
    CC8 VARCHAR (30) NOT NULL,
    DC8 VARCHAR (100) NOT NULL,
    CC9 VARCHAR (30) NOT NULL,
    DC9 VARCHAR (100) NOT NULL,
    CC10 VARCHAR (30) NOT NULL,
    DC10 VARCHAR (100) NOT NULL,
    CC11 VARCHAR (30) NOT NULL,
    DC11 VARCHAR (100) NOT NULL,
    CC12 VARCHAR (30) NOT NULL,
    DC12 VARCHAR (100) NOT NULL,
    CC13 VARCHAR (30) NOT NULL,
    DC13 VARCHAR (100) NOT NULL,
    CC14 VARCHAR (30) NOT NULL,
    DC14 VARCHAR (100) NOT NULL,
    CC15 VARCHAR (30) NOT NULL,
    DC15 VARCHAR (100) NOT NULL,
    CC16 VARCHAR (30) NOT NULL,
    DC16 VARCHAR (100) NOT NULL,
    CC17 VARCHAR (30) NOT NULL,
    DC17 VARCHAR (100) NOT NULL,
    CC18 VARCHAR (30) NOT NULL,
    DC18 VARCHAR (100) NOT NULL,
    CC19 VARCHAR (30) NOT NULL,
```

```

DC19 VARCHAR (100) NOT NULL,
CC20 VARCHAR (30) NOT NULL,
DC20 VARCHAR (100) NOT NULL,
CC21 VARCHAR (30) NOT NULL,
DC21 VARCHAR (100) NOT NULL,
CC22 VARCHAR (30) NOT NULL,
DC22 VARCHAR (100) NOT NULL,
CC23 VARCHAR (30) NOT NULL,
DC23 VARCHAR (100) NOT NULL,
CC24 VARCHAR (30) NOT NULL,
DC24 VARCHAR (100) NOT NULL,
CC25 VARCHAR (30) NOT NULL,
DC25 VARCHAR (100) NOT NULL,
CC26 VARCHAR (30) NOT NULL,
DC26 VARCHAR (100) NOT NULL,
CC27 VARCHAR (30) NOT NULL,
DC27 VARCHAR (100) NOT NULL,
CC28 VARCHAR (30) NOT NULL,
DC28 VARCHAR (100) NOT NULL,
CC29 VARCHAR (30) NOT NULL,
DC29 VARCHAR (100) NOT NULL,
CC30 VARCHAR (30) NOT NULL,
DC30 VARCHAR (100) NOT NULL,
DAC VARCHAR (20) NOT NULL)

```

```

DECLARE ref_cursor CURSOR FOR SELECT RTRIM(st.u_refpai), RTRIM(st.u_reftam)
FROM st (NOLOCK) WHERE st.inactivo = 0 and st.u_nexp=0 and st.stns=0 and
st.u_refpai <> '' GROUP BY st.u_refpai, st.u_reftam order by st.u_refpai,
st.u_reftam
OPEN ref_cursor
FETCH NEXT FROM ref_cursor INTO @REF, @TAM
SET @SQL = ''
SET @REFT = ''
SET @POSICAO = 1
WHILE @@FETCH_STATUS = 0
BEGIN
    IF (@REFT <> @REF)
    BEGIN
        SET @REFT = @REF

        IF @SQL = ''
            SET @SQL = 'SELECT ''' + @REFT + ''', ''Tamanhos'',
''Cores'''
        ELSE
        BEGIN
            WHILE @POSICAO <= 30
            BEGIN
                SET @SQL = @SQL + ', ''', ''''
                SET @POSICAO = @POSICAO + 1
            END
        END
    END

```

```

SET @POSICAO = 1
SET @SQL = @SQL + ', ''''

INSERT INTO @MSGRH EXEC (@SQL)
SET @SQL = 'SELECT ''' + @REFT + ''', ''Tamanhos'', ''Cores'''
END
END

IF @TAM <> ''
BEGIN
    SET @SQL = @SQL + ', '''' + @TAM + ''', ''TAM ' + @TAM + ''''
    SET @POSICAO = @POSICAO + 1
END
FETCH NEXT FROM ref_cursor INTO @REF, @TAM
END
CLOSE ref_cursor;
DEALLOCATE ref_cursor;

IF @REFT <> ''
BEGIN
    IF @POSICAO <= 30
    BEGIN
        WHILE @POSICAO <= 30
        BEGIN
            SET @SQL = @SQL + ', ''''', ''''
            SET @POSICAO = @POSICAO + 1
        END
    END

    SET @POSICAO = 1
    SET @SQL = @SQL + ', ''''

INSERT INTO @MSGRH EXEC (@SQL)
END
END

SELECT * FROM @MSGRH

DELETE @MSGRH

```

- Grelha horizontal (GRV):

- SQL 1

Inserir o código sql de baixo no separador SQL em modo avançado:

```

DECLARE @POSICAO INT
DECLARE @INC INT
DECLARE @SQL VARCHAR(MAX)
DECLARE @REF VARCHAR(60)
DECLARE @REFT VARCHAR(60)
DECLARE @TAM VARCHAR(25)
DECLARE @TAMT VARCHAR(25)
DECLARE @COR VARCHAR(25)
DECLARE @DESCCOR VARCHAR(100)
DECLARE @CORT VARCHAR(25)
DECLARE @MSGRV TABLE (
    REF VARCHAR(60) not null,
    DESCR VARCHAR(100) NOT NULL,
    DESCRV VARCHAR(100) NOT NULL,
    SEQ INT NOT NULL,
    CC1 VARCHAR (1) NOT NULL,
    CC2 VARCHAR (1) NOT NULL,
    CC3 VARCHAR (1) NOT NULL,
    CC4 VARCHAR (1) NOT NULL,
    CC5 VARCHAR (1) NOT NULL,
    CC6 VARCHAR (1) NOT NULL,
    CC7 VARCHAR (1) NOT NULL,
    CC8 VARCHAR (1) NOT NULL,
    CC9 VARCHAR (1) NOT NULL,
    CC10 VARCHAR (1) NOT NULL,
    CC11 VARCHAR (1) NOT NULL,
    CC12 VARCHAR (1) NOT NULL,
    CC13 VARCHAR (1) NOT NULL,
    CC14 VARCHAR (1) NOT NULL,
    CC15 VARCHAR (1) NOT NULL,
    CC16 VARCHAR (1) NOT NULL,
    CC17 VARCHAR (1) NOT NULL,
    CC18 VARCHAR (1) NOT NULL,
    CC19 VARCHAR (1) NOT NULL,
    CC20 VARCHAR (1) NOT NULL,
    CC21 VARCHAR (1) NOT NULL,
    CC22 VARCHAR (1) NOT NULL,
    CC23 VARCHAR (1) NOT NULL,
    CC24 VARCHAR (1) NOT NULL,
    CC25 VARCHAR (1) NOT NULL,
    CC26 VARCHAR (1) NOT NULL,
    CC27 VARCHAR (1) NOT NULL,
    CC28 VARCHAR (1) NOT NULL,
    CC29 VARCHAR (1) NOT NULL,
    CC30 VARCHAR (1) NOT NULL,
    TES VARCHAR (1) NOT NULL)
DECLARE ref_cursor CURSOR FOR SELECT RTRIM(ST.u_refpai), RTRIM(st.u_refcor),
RTRIM(st.u_desccor), RTRIM(st.u_reftam) FROM ST (NOLOCK) WHERE st.inactivo =
0 and st.u_nexp=0 and st.stns=0 and st.u_refpai <> '' GROUP BY st.u_refpai,
st.u_refcor, st.u_desccor, st.u_reftam ORDER BY st.u_refpai, st.u_refcor,
st.u_reftam
OPEN ref_cursor
FETCH NEXT FROM ref_cursor INTO @REF, @COR, @DESCCOR, @TAM
    
```

```

SET @SQL = ''
SET @REFT = ''
SET @CORT = ''
SET @POSICAO = 1
WHILE @@FETCH_STATUS = 0
BEGIN
    IF (@REFT <> @REF)
    BEGIN
        SET @REFT = @REF
        SET @CORT = @COR
        SET @INC = 1
        IF @SQL = ''
        BEGIN
            SET @SQL = 'SELECT ''' + @REFT + ''', ''' + @CORT + ''',
            ''' + @DESCCOR + ''', ' + RTRIM(CAST(@INC AS CHAR(3)))
            DECLARE ref_cursorTamanhos SCROLL CURSOR FOR SELECT
            RTRIM(ST.u_reftam) FROM ST (NOLOCK) WHERE st.u_refpai = @REFT and st.inativo
            = 0 and st.u_nexp=0 and st.stns=0 GROUP BY st.u_reftam ORDER BY st.u_reftam
            OPEN ref_cursorTamanhos

            END
            ELSE
            BEGIN
                CLOSE ref_cursorTamanhos;
                DEALLOCATE ref_cursorTamanhos;
                WHILE @POSICAO <= 30
                BEGIN
                    SET @SQL = @SQL + ', 'N''
                    SET @POSICAO = @POSICAO + 1

                END

                SET @POSICAO = 1
                SET @SQL = @SQL + ', ''''

                --PRINT @SQL
                INSERT INTO @MSGRV exec (@SQL)
                SET @SQL = 'SELECT ''' + @REFT + ''', ''' + @CORT + ''',
                ''' + @DESCCOR + ''', ' + RTRIM(CAST(@INC AS CHAR(3)))
                DECLARE ref_cursorTamanhos SCROLL CURSOR FOR SELECT
                RTRIM(ST.u_reftam) FROM ST (NOLOCK) WHERE st.u_refpai = @REFT and st.inativo
                = 0 and st.u_nexp=0 and st.stns=0 GROUP BY st.u_reftam ORDER BY st.u_reftam
                OPEN ref_cursorTamanhos

                END
            END

            IF (@CORT <> @COR)
            BEGIN
                SET @CORT = @COR

                WHILE @POSICAO <= 30
                BEGIN
                    SET @SQL = @SQL + ', 'N''
                    SET @POSICAO = @POSICAO + 1

                END

                SET @POSICAO = 1
                SET @INC = @INC + 1
                SET @SQL = @SQL + ', ''''

                INSERT INTO @MSGRV exec (@SQL)

```

```

SET @SQL = 'SELECT ''' + @REFT + ''', ''' + @CORT + ''', ''' +
@DESCCOR + ''', ' + RTRIM(CAST(@INC AS CHAR(3)))

END

IF @TAM <> ''
BEGIN
    DECLARE @IPOST INT = 1
    --DECLARE ref_cursorTamanhos SCROLL CURSOR FOR SELECT
    RTRIM(ST.u_reftam) FROM ST (NOLOCK) WHERE st.u_refpai = @REFT and st.inativo
    = 0 and st.u_nexp=0 and st.stns=0 GROUP BY st.u_reftam ORDER BY st.u_reftam
    --OPEN ref_cursorTamanhos
    --FETCH NEXT FROM ref_cursorTamanhos INTO @TAMT
    FETCH FIRST FROM ref_cursorTamanhos INTO @TAMT
    WHILE @@FETCH_STATUS = 0
    BEGIN
        IF (@POSICAO <= @IPOST)
        BEGIN
            IF (@TAM = @TAMT)
            BEGIN
                SET @SQL = @SQL + ', 'S''
                FETCH LAST FROM ref_cursorTamanhos INTO @TAMT

            END
            ELSE
            BEGIN
                SET @SQL = @SQL + ', 'N''
            END
            SET @POSICAO = @POSICAO + 1
        END
        FETCH NEXT FROM ref_cursorTamanhos INTO @TAMT
        SET @IPOST = @IPOST + 1
    END

    --CLOSE ref_cursorTamanhos;
    --DEALLOCATE ref_cursorTamanhos;
END
    FETCH NEXT FROM ref_cursor INTO @REF, @COR, @DESCCOR, @TAM
END
CLOSE ref_cursor;
DEALLOCATE ref_cursor;
CLOSE ref_cursorTamanhos;
DEALLOCATE ref_cursorTamanhos;

IF @REFT <> ''
BEGIN
    WHILE @POSICAO <= 30
    BEGIN
        SET @SQL = @SQL + ', 'N''
        SET @POSICAO = @POSICAO + 1
    END
    SET @SQL = @SQL + ', ''''

    INSERT INTO @MSGRV EXEC (@SQL)

END

SELECT * FROM @MSGRV ORDER BY REF
DELETE @MSGRV

```

- Tabela Documentos origem (DOI):

- SQL 1

Mapeamentos

Alterar o mapeamento dos seguintes campos.

Campo	Descrição	Tabela ERP	Camp ERP
DOIART	Artigo	%OPTIONS%	<p><u>Campo a validar:</u> st.u_refpai</p> <p><u>Se igual a...:</u> " Então: fi.ref</p> <p>Senão: (st.u_refpai + '.' + RTRIM(st.u_refcor) + '.' + RTRIM(st.u_reftam))</p>

Para concluir a configuração do driver, na opção “Avançadas” e separador “Extras”, deverá ser acrescentado o sql em baixo antes do sql “{call SPMSS_Integra}”.

{call SPMSS_SubRefCorTamanho}

A configuração em cima deverá ficar conforma a imagem em baixo:

