

# SAPscript

<i>O Que é SAPscript e quando utilizar.....</i>	<i>2</i>
<i>Limitações e Problemas da Ferramenta.....</i>	<i>2</i>
<i>Programa de Povoamento de SAPscript.....</i>	<i>3</i>
<i>Abertura de formulário.....</i>	<i>3</i>
<i>Inicialização de um formulário.....</i>	<i>4</i>
<i>Seleção dos dados.....</i>	<i>5</i>
<i>Impressão dos dados nas janelas.....</i>	<i>5</i>
<i>O Formulário.....</i>	<i>7</i>
1. Header.....	7
2. Parágrafos .....	7
3. Character String .....	7
4. Janelas .....	7
MAIN .....	7
VAR .....	8
CONST.....	8
5. Páginas .....	8
6. Page Windows .....	9
7. Text Elements .....	9
<i>Manutenção de formulários.....</i>	<i>9</i>
1. Criar um formulário novo.....	9
2. Copiar formulários em outro idioma (Tradução) .....	10
3. Copiar um formulário de outro cliente .....	10
<i>Inclusão de logos no formulário.....</i>	<i>10</i>
<i>Alterção de Programas / Layout Standards.....</i>	<i>11</i>

## O Que é SAPscript e quando utilizar

SAPscript é a ferramenta do SAP para criação de relatórios mais elaborados, chamados de formulários. A opção pela utilização da ferramenta geralmente se dá pela necessidade de alguns recursos que não podem ser obtidos através de código ABAP comum, como nos chamados "reports", tais como inserção de logotipos, fontes de tamanhos ou tipos diferentes, et... Muitas vezes, também, dependendo da necessidade do relatório, pode compensar mais utilizar o desenvolvimento de um relatório convencional sobre um formulário pré-impresso, que já contenha o layout e logotipo desejados.

## Limitações e Problemas da Ferramenta

Embora seja a ferramenta de geração de formulários mais poderosa do SAP, o SAPscript ainda apresenta uma série de limitações, principalmente no que diz respeito a verificação de erros e análise de debug. Um comando escrito na grafia errada dentro do formulário, jamais será detectado pelo mesmo. Somente a não funcionalidade esperada do comando poderá levar ao programador descobrir se próprio erro.

O modo debug no ambiente de SAPscript pode ser ativado na transação SE71 (Ferramentas -> Processamento de Texto -> Layout Set), através do menu Utilitários -> Ativar Debug. O método correto consiste em se preencher o nome do formulário e idioma nos campos da tela, antes de percorrer o caminho do menu. Algumas vezes, no entanto, principalmente para um processamento em background, o formulário não permite ser debugado, embora percorrendo o caminho acima citado, nenhuma mensagem de impossibilidade seja exibida. Simplesmente, nenhuma tentativa de debug funciona. Nesses casos, o programador deve utilizar de artifícios como imprimir o conteúdo dos campos a serem checados em pontos estratégicos do formulário, para observar seu comportamento.

O comportamento de um formulário SAPscript pode também ser muitas vezes irritante, principalmente quando se desconhece alguns de seus detalhes. Um dos problemas mais comuns do formulário SAPscript é quanto a utilização de logotipos nos relatórios gerados. Sua filosofia pode parecer bastante simples e é, mas essa operação é bastante sensível, e devem ser salientados alguns pontos:

1. O objeto com o **logotipo deve ser gerado a partir de um programa ABAP**, através da transação SE38, chamado RX... . O modo como o arquivo com o logotipo é tratado, deve ser cuidadosamente verificado, pois implicará no sucesso ou fracasso da exibição do logotipo esperado. Inicialmente, deve ser ressaltado que o arquivo deve estar no **foramto .TIF** . Observado esse ponto, assegurar que o arquivo não apresente mais do que 16 cores e por último o seu tamanho não apresenta uma relação direta com o tamanho que será obtido dentro do formulário SAPscript. Até onde se tem notícias, ninguém nunca descobriu uma relação correta e geral para o tamanho de um logo em pixels e seu tamanho dentro do formulário. Geralmente essa relação só é observada caso a caso, após algumas tentativas de erro e acerto.
2. Nessa hora entra a segunda dificuldade de se trabalhar com logotipos em SAPscript. Ao se executar o formulário, em uma visualização na tela, o usuário nunca irá conseguir enxergar o logotipo. Esse somente irá aparecer na impressão em papel (ainda que impressão do layout do formulário, e não seu conteúdo final – Utilitários -> Imprimir Layout).

Essas duas características podem tomar algumas horas do programador, ainda que bem experiente no trabalho com SAPscript.

É importante se observar também que os formulários são "client dependent", o que significa dizer que devem ser transportados a todos os clientes em que se deseja executa-lo, independentemente de estar na mesma instância que já o tenha. Isso não pareceria ser muito complicado se não fosse um trabalho a mais controlar a versão de todos no momento que uma alteração feita. Pior do que eles, são os objetos gerados como os logotipos, que além de dependentes do cliente, não estão vinculados a nenhuma request, não podendo nem ser transportados de um cliente para outro.

Por falar em transporte, é sempre bom reforçar que no momento do transporte de um formulário SAPscript, é necessário assegurar que todas as estruturas utilizadas por ele, tenham

sido transportadas anteriormente. O objetivo é assegurar que tudo que será utilizado pelo formulário já esteja no cliente, no momento em que este for introduzido, para que não haja erros. A ordem mais aconselhada para esses transportes seria:

1. Estrutura
2. Layout
3. Programa de povoamento

Também vale a pena chamar a atenção para que essas 3 classes de objetos estejam em requests separadas no desenvolvimento de um projeto, o que pode evitar problemas quando algum deles não estiver no desenvolvimento propriamente e necessitar ser reparado.

## Programa de Povoamento de SAPscript

O programa de povoamento é o programa responsável pelo controle da impressão do formulário. É através dele que o formulário é aberto, que os dados são enviados para as janelas corretas, que é feito o controle de quebra de páginas, que é feito o fechamento do formulário, etc..., é feita nele. O formulário é apenas um "dispositivo" para output dos dados.

A construção de um programa de povoamento deve ser a seguinte estrutura:

- Abertura de um formulário
- Inicialização de um formulário
- Seleção dos dados do formulário
- Impressão dos dados nas janelas do formulário
- Fechamento do formulário

### **Abertura de formulário**

O primeiro passo para a impressão de um SAPscript através de um programa é a abertura de um formulário para que se possa fazer o povoamento.

Para que se possa inicializar um formulário é necessário que este esteja aberto. Portanto, é obrigatória a presença de um comando de abertura num programa de povoamento.

A abertura de um formulário gera automaticamente a inicialização de um formulário.

O comando para a abertura de um formulário é a seguinte função:

```
Call function 'OPEN_FORM'
  exporting
*     APPLICATION           = 'TX'
*     ARCHIVE_INDEX        = ' '
*     ARCHIVE_PARAMS       = ' '
*     DEVICE                = 'PRINTER'
*     DIALOG               = 'X'
*     FORM                 = (nome do formulário)
*     LANGUAGE            = (idioma do formulário)
*     OPTIONS              = ITCPO
*   importing
*     language              =
*     new_archive_params    =
*     result                =
*   exceptions
*     canceled              = 1
*     device                = 2
*     form                  = 3
*     options               = 4
*     unclosed              = 5
*     others                = 6.
```

Os parâmetros mais importantes a serem passados para a função são:

**DEVICE** - Dispositivo onde será gerado o formulário (geralmente PRINTER)  
**FORM** - Nome do formulário a ser aberto  
**LANGUAGE** - Idioma a ser utilizado

Dois outros parâmetros também muitas vezes utilizados são o **dialog** e o **options**.

No primeiro, indica-se se a caixa de diálogo para confirmação da impressão deve ou não ser apresentado. Caso este deva ser omitido, os dados para impressão serão tomados de uma estrutura a ser colocada no segundo parâmetro. Nessa estrutura são passados os dados tais como número de cópias e nome da impressora. A opção de não exibição da caixa de diálogo está diretamente ligada a utilização do segundo parâmetro. Se a exibição estiver desabilitada (DIALOG=' '), mas a estrutura ITCPO não estiver preenchida, a caixa de diálogo será exibida.

Alguns dos campos mais importantes e que normalmente são utilizados na estrutura ITCPO são:

**TDCOPIES** - Número de cópias  
**TDDEST** - Dispositivo de saída  
**TDPREVIEW** - Print Preview  
**TDIMMED** - Saída imediata

Exemplo de abertura do formulário ZSCR\_CURSO.

```
Call function 'OPEN_FORM'
  exporting
*     APPLICATION           = 'TX'
*     ARCHIVE_INDEX        = ' '
*     ARCHIVE_PARAMS       = ' '
*     DEVICE              = 'PRINTER'
*     DIALOG             = ' '
*     FORM               = 'ZSCR_CURSO'
*     LANGUAGE          = 'P'
*     OPTIONS           = ITCPO
*   importing
*     language              =
*     new_archive_params   =
*     result                =
  exceptions
    canceled                = 1
    device                  = 2
    form                    = 3
    options                 = 4
    unclosed                = 5
    others                  = 6.
```

Neste exemplo, o formulário ZSCR\_CURSO no idioma português está aberto, sem a exibição da caixa de diálogo de impressão, utilizando a impressora IMP1 e executando a impressão imediata de 2 cópias.

### ***Inicialização de um formulário***

A abertura de um formulário automaticamente gera a inicialização do mesmo.

Mas, imagine por exemplo que estejamos gerando relatórios para uma empresa de computadores onde, para cada novo computador vendido, deve ser gerado um novo relatório, e esse relatório deve conter todos os componentes presentes no computador em questão.

A abertura de um formulário inicia o primeiro, mas para cada novo computador vendido um novo formulário deve ser gerado.

Para isso utilizados o comando de inicialização. Este comando permite que, numa mesma impressão, sejam gerados vários formulários diferentes, como se estivéssemos abrindo um novo para cada computador vendido.

Este comando não é obrigatório e pode não ser utilizado caso não haja necessidade da quebra e criação de vários formulários para uma só seleção de dados.

O comando para inicialização de um formulário é a seguinte função:

```
Call function 'START_FORM'
  exporting
*     ARCHIVE_INDEX      = \ \
      FORM              = \ /
      LANGUAGE          = \ /
*     STARTPAGE         = \ \
*     PROGRAM           = \ \
*   importing
*     language          =
  exceptions
      form              = 1
      format            = 2
      unended           = 3
      unopened         = 4
      unused            = 5
      others            = 6.
```

A função é muito similar á função OPEN\_FORM, e novamente mais importante são o nome (FORM) e o idioma (LANGUAGE) do formulário.

Para que se possa utilizar um comando START\_FORM é obrigatório que um comando de OPEN\_FORM tenha sido executado.

### **Seleção dos dados**

A seleção dos dados é feita normalmente com comando SELECT e demais comandos ABAP, como já foi visto em tópicos anteriores.

Um ponto importante que deve ser observado é que não é possível utilizar tabelas internas e variáveis do programa para a impressão dos dados SAPscript. Portanto, se algum dado que deva ser impresso estiver numa tabela interna ou variável, este deve ser copiado para uma estrutura para que possa ser enviado ao SAPscript.

### **Impressão dos dados nas janelas**

A impressão dos dados nas janelas na maioria das vezes é feita simultaneamente com a seleção dos dados, ou seja, a medida que os dados são selecionados, são enviados imediatamente para o formulário.

O comando para impressão dos dados é a seguinte função:

```
Call function 'WRITE_FORM'
  exporting
      ELEMENT           = \ \
*     FUNCTION          = 'SET \
*     TYPE              = 'BODY \
      WINDOW           = 'MAIN \
*   importing
*     pending_lines     =
  exceptions
      element           = 1
      function          = 2
```

type	= 3
unopened	= 4
unstarted	= 5
window	= 6
others	= 7.

Dois parâmetros são os mais importantes:

- ELEMENT** - Determina qual Data Element será utilizado dentro do SAPscript  
**WINDOW** - janela na qual os dados devem ser impressos

Neste ponto fica evidente a diferenciação entre os tipos de janela MAIN e demais janelas.

Para as janelas do tipo MAIN, cada comando de escrita (write\_form) significa uma nova linha no formulário e o valor a ser impresso é o valor que o campo armazena no momento do comando de impressão.

Por exemplo, digamos que o campo MARA-MATNR tenha o valor '1234' e que um comando de impressão seja dado para a janela MAIN que irá imprimir este campo. Logo em seguida uma nova seleção da tabela MARA é feita e o campo MATNR agora vale '5678'. Se uma nova impressão na janela MAIN for executada o resultado será o seguinte:

**1234**  
**5678**

Já as janelas que não forem do tipo MAIN imprimem os dados uma única vez, no final da impressão do formulário ou na quebra de página, com os valores armazenados nos campos no momento do encerramento ou no momento da quebra, e não no momento da escrita (write\_form), ou seja, se no exemplo anterior fosse utilizada uma janela não-MAIN, o resultado final seria somente **5678**.

Na realidade, a utilização de um comando WRITE\_FORM numa janela não-MAIN é utilizada para a escolha de qual elemento de texto será utilizado para a impressão dos dados.

Exemplo:

Uma janela HEADER não-MAIN contém dois elementos de texto chamados FRASE1 e FRASE2 da seguinte forma:

```
/E   FRASE1
      &MARA-MATNR&           Teste de Frase 1
/E   FRASE2
      &MARA-MATNR&           Teste de Frase 2
```

Se um comando WRITE\_FORM for executado para a janela HEADER utilizando o elemento de texto FRASE1,

```
Call function 'WRITE_FORM'
  exporting
    ELEMENT      = 'FRASE1'
    * FUNCTION   = 'SET'
    * TYPE       = 'BODY'
    WINDOW       = 'HEADER'
  ...
```

estará sendo indicado ao programa que, ao se escrever o formulário, deve ser impresso o elemento de texto FRASE1 para a janela HEADER (somente um elemento de texto é utilizado para cada janela não-MAIN).

No caso do exemplo, se o campo MATNR for igual a '1234' no encerramento do formulário, seria impressa a seguinte frase:

**1234            Teste de Frase 1**

O elemento de texto a ser impresso será sempre o último a ser selecionado antes do final ou da quebra de página.

Se nenhum for selecionado, ao encerrar o formulário ou mudar de página serão impressos os dados que não pertençam a nenhum elemento de texto. Portanto, no caso de um campo que deva ser impresso sempre em todas as páginas, como número de página por exemplo, basta coloca-lo fora de qualquer elemento de texto e não selecionar nenhum elemento de texto para a janela que, automaticamente, este dado será impresso em todas as páginas.

## O Formulário

Trabalhar com um formulário SAPscript não é complicado, desde que se entenda seus componentes da maneira correta. Um layout é composto por vários componentes, dentre os quais podemos citar:

### 1. *Header*

Informações gerais do formulário, como formato e orientação da página em que será utilizado, parágrafo e fontes default do mesmo (quando não for especificado nenhuma dentro do form). Nesse tópico aparece também o título do formulário, a classe de desenvolvimento do mesmo e os idiomas do formulário original e o corrente.

### 2. *Parágrafos*

São elementos que podem ser criados pelo usuário e apresentam informações dos textos de uma determinada janela. Nesses parágrafos são determinados tipos de fontes, tabulações, negrito ou sublinhado, etc...

Para criar um parágrafo novo, utilizar o caminho do menu Processar -> Create Element, lembrando que em um formulário standard, somente se pode alterar o layout no idioma original.

### 3. *Character String*

São informações semelhantes às do parágrafo, no que diz respeito a formatação de textos, no entanto podem ser aplicados a trechos e não necessariamente a linhas do formulário.

Sua sintaxe de utilização se assemelha a programação HTNL, conforme exemplo a seguir. Imagine um Character string definido com o nome B, cuja formatação de texto seja o incremento do modelo negrito a uma determinada fonte. Sua utilização seria:

```
<B>Texto</>
```

E o resultado seria:

**Texto**

### 4. *Janelas*

As janelas são a base da construção do layout. As janelas podem ser criadas sem limites e tem por finalidade dividir espacialmente as regiões de um formulário para operações independentes.

Essas janelas podem ser criadas basicamente em 3 tipos distintos:

#### **MAIN**

Janelas do tipo MAIN são as mais importantes do relatório (janela pela qual o formulário se guia). Somente é permitido um tipo de janela MAIN por formulário, e é a única janela obrigatória em uma página. Pode-se criar mais de uma janela MAIN por página, no entanto essas janelas, na verdade, só são divididas fisicamente, uma vez que seus elementos e processamentos obedecem aos mesmos comandos, e não podem ter nomes diferentes. A passagem de uma janela MAIN para outra pode ser automática ou forçada pelo usuário. Quando o espaço físico de uma janela MAIN está tomado e necessita-se de mais dados na mesma, o formulário automaticamente passa à próxima janela MAIN do formulário. É através desse princípio que um relatório ganha número de páginas automaticamente quando da

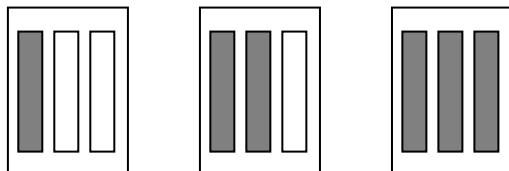
execução de um relatório desse tipo, conforme os dados são preenchidos. Para forçar a passagem de uma janela MAIN para outra, o usuário poderá utilizar o comando:

/: NEXT-WINDOW

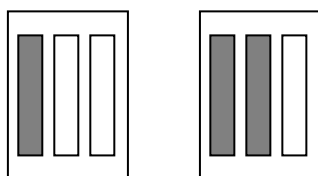
A lógica de seqüência das janelas MAIN de um formulário segue a um critério. Dentro de uma página a navegação de uma MAIN para outra segue de acordo com os índices dessas janelas dentro do formulário, e quando uma nova página é gerada, a contagem segue a mesma disposição, continuamente.

O exemplo abaixo, figura a situação de uma diagramação conforme uma página de jornal. A seqüência de preenchimento dos dados na página através do povoamento segue:

Pág. 1



Pág. 2



## VAR

Essa janela apresenta uma característica própria, que necessita de verificação constante do formulário com seus dados, que podem variar durante a execução do relatório. Esse tipo de janela deve ser utilizada quando se pretende trabalhar com textos que variam, como exemplos de mensagens que podem mudar durante o decorrer da execução do formulário.

## CONST

São janelas que não merecem muita preocupação de processamento por parte do formulário, uma vez que seus dados devem ser mantidos os mesmos desde seu primeiro preenchimento até o fechamento do formulário. Geralmente nesse tipo de janela são incluídos objetos e textos que permanecem os mesmos durante todo o relatório, independente do número de páginas, como logo e rodapés.

Na atual versão do SAP, esta distinção de atenção de processamento do formulário é só teórica, mas deve ser levada em conta, pois deve ser tratada por versões mais novas do sistema.

## 5. Páginas

As páginas são elementos responsáveis pelo agrupamento das janelas durante certos momentos do relatório. Geralmente só são definidas duas páginas, que acabam em processo reflexivo, por formulário.

Geralmente uma página FIRST, que apresenta um layout de apresentação dos documentos e outra NEXT, que contém os layouts das outras páginas subseqüentes do documento. Isso acontece, por exemplo, quando se quer obter um relatório com o logotipo da empresa na primeira página, mas as seguintes, devem ocupar a folha inteira com os dados do relatório até o final. Cada uma das páginas criadas devem conter um nome (FIRST, NXT, ...) e pode conter um parágrafo default e um link para a próxima página. Esse link serve para formar a seqüência em que as páginas do relatório devem ser geradas. Seguindo o exemplo anterior, deveríamos informar:

FIRST -> NEXT  
NEXT -> NEXT



Após a primeira página, seria chamada a segunda, com um layout diferente (sem o logotipo). A partir dessa, todas apresentariam o mesmo tipo de layout (com dados do relatório aproveitando a página inteira).

## 6. *Page Windows*

As page windows podem ser entendidas pelo nosso conceito de layout propriamente dito. São os elementos responsáveis pelo aparecimento ou não das janelas dentro de uma determinada página, bem como seus posicionamentos em relação à folha a ser impressa. É também através das page windows que o programador terá acesso aos Text Elements de cada uma das janelas, responsáveis pelo tratamento dos dados em tempo de execução do formulário.

## 7. *Text Elements*

Corresponde à lógica de processamento e comportamento de uma janela durante a execução do formulário. É responsável pelos valores que serão impressos no mesmo, bem como seu posicionamento dentro da janela e seu tipo de caracter.

De dentro de uma Page Window, é possível selecionarmos qualquer uma das janelas existentes em uma determinada página, simplesmente clicando o mouse sobre o seu nome e apertando o botão selecionar, ou mesmo dando um “doublé-click” diretamente sobre o nome da janela desejada. Uma vez selecionada a janela, se apertarmos o botão Text Element, uma tela semelhante a do editor de ABAP/4 do SAP irá se abrir, com as linhas onde a lógica da janela será introduzida.

Nesses elementos do SAPscript, o programador poderá utilizar alguns comandos semelhantes ao do ABAP/4, com a indicação de que se trata de uma linha de comando para o formulário (/:, à esquerda da linha). Abaixo, alguns dos comandos que podem ser utilizados de dentro dos elementos de texto de uma janela em SAPscript:

**IF...ELSE...ENDIF**  
**CASE...WHEN**  
**DEFINE**

**PERFORM...USING...CHANGING**  
**BOX...FRAME**

**INCLUDE**

O comando INCLUDE dentro de uma janela de um formulário SAPscript serve para inserir um objeto no mesmo. Esse objeto pode ser desde um logo, até um texto com várias linhas. A sintaxe é bastante simples, e somente deve-se ter um cuidado bastante especial com o idioma que se trata esse objeto. Independentemente do idioma do formulário, deve-se sempre utilizar o idioma em que o objeto foi criado.

## **Manutenção de formulários**

A maioria dos formulários é feita pela mesma transação citada anteriormente (SE71). Nessa ela o nome do formulário e o idioma do mesmo são parâmetros obrigatórios. Nessa tela são feitas praticamente todas as operações com os formulários do SAP.

### **1. Criar um formulário novo**

Um formulário pode ser criado inteiramente a partir dessa transação. Preenchendo um nome do formulário, que comece por Z ou Y, não diretamente seguidos por caracteres especiais, como o ‘\_’, o usuário poderá apertar o botão **Criar** ou ir ao menu **Layout -> Criar**. O idioma preenchido será entendido como idioma original do formulário. A partir daí, outros formulários

poderão ser traduzidos a partir desse original, nas mais diferentes línguas, porém todos manterão as características do formulário original. Ao mesmo tempo que uma alteração desse original, implica na imediata alteração de todos os seus “filhos”.

## **2. Copiar formulários em outro idioma (Tradução)**

Uma vez criado um formulário (conforme descrito acima), poderá ser copiado sobre ele, as informações de um outro formulário do mesmo client, através do menu **Layout -> Copy form**, já de dentro da primeira tela do formulário criado (Header).

Atenção que essa cópia é fiel ao formulário copiado, e não poupa nenhuma informação que já existisse no formulário que recebe a cópia.

## **3. Copiar um formulário de outro cliente**

Um formulário pode também ser copiado de um outro client na mesma instância, através do menu **Utilitários -> Copy from client**, da primeira janela da transação **SE71**, fora de qualquer formulário específico.

Essa opção irá abrir uma outra janela que recebe basicamente 3 informações.

**Nome do formulário a ser copiado**  
**Cliente em que se encontra o original**  
**Nome da cópia no client corrente**

Os outros dois campos de check Box geralmente não são alterados, pois tratam de informações como cópia do formulário somente no idioma original e exibição ou não de um log de erros ao final da execução da cópia.

Uma vez entrados os parâmetros, deve-se apertar o botão **Executar** e obter uma cópia com sucesso em todos os idiomas existentes do formulário.

O cliente de origem do formulário, quando for cópia de forms standards, deve ser mantido 000, uma vez que o repositório dos formulários originais do SAP encontram-se todos no 000, e podem ser lidos de todos os outros.

## **Inclusão de logos no formulário**

Utilizar logotipos em formulários SAPscript, não é das tarefas mais simples que se possa ter. Teoricamente falando, o processo é bastante simples e consiste em gerar no sistema um objeto no formato hexadecimal, que possa ser interpretado pelo SAPscript, originando a inclusão de uma imagem. No entanto, devido as limitações expostas no começo deste documento, isso pode levar um certo tempo até reproduzir o resultado desejado.

Para gerarmos o objeto no client desejado (lembre-se que esse objeto não pode ser transportado...), devemos executar o programa standard **RSTXLDMC**, com um arquivo no formato **.TIF**.

### **(A figura não pode ter mais de 16 cores)**

Preencher o primeiro campo (file name), com o path complexo do arquivo da imagem do logotipo. Esse caminho pode ser no servidor ou local.

O segundo campo (type) determina se o logotipo deve ser gerado monocromático (default) ou colorido.

O último parâmetro que exige ser preenchido é o Text Name, onde é feita a atribuição do nome do objeto que será referenciado de dentro do formulário (padrão: ZHEX-MACRO-...).

A geração desse logo está intimamente ligado ao formato do arquivo TIF. Isso quer dizer que o seu tamanho obtido no formulário está relacionado ao tamanho da imagem geradora do objeto.

Tomar cuidado com o número de cores da imagem (16 cores) e lembrar na hora de fazer os testes, que o logotipo não aparece na tela, somente em impressões no papel.

## Alterção de Programas / Layout Standards

Na maioria das vezes o trabalho a ser realizado com SAPscript é alterar os originais do SAP, pois estes raramente atendem às exigências do cliente.

O procedimento para alteração depende muito das características do programa de povoamento standard e das características das funções do SAP que irão utilizar tal SAPscript. No entanto, alguns procedimentos são padrão, e outros são utilizados na grande maioria das alterações.

1. Fazer uma cópia do programa standard e dos includes (se existirem) que necessitam de alteração
2. Fazer uma cópia do layout standard
3. **Não DELETAR** janelas, elementos de textos, etc. na cópia do, layoutset. Mesmo que alguns (ou todos) os dados e janelas do standard não sejam utilizados, o programa de povoamento continuará tentando executar escritas em janelas e elementos de dados específicos, já que é uma cópia do standard. Os novos dados e janelas devem ser apenas adicionados.
4. Encontrar as posições idéias para alterações no programa de povoamento. Quando os novos dados devem ser impressos na MAIN, procurar por comandos WRITE\_FORM dentro do programa e colocar as alterações o mais próximo possível do WRITE\_FORM correto. Se devem ser impressos em outras janelas, procurar pelo comando CLOSE\_FORM / END\_FORM e colocar as alterações imediatamente antes destes comandos (sempre que possível)
5. Colocar todas as alterações juntas sempre que possível, identificando visivelmente, através de comentários, o bloco de alterações. Isso facilita a manutenção e visualização do código, além de facilitar a busca por possíveis erros, visto que, na sua grande maioria, os programas standard da SAP são bastante complicados.