

<b>MANUAL DO INTERPRETADOR DE FÓRMULAS</b> .....	<b>1</b>
<i>INTRODUÇÃO</i> .....	1
<i>NOVO INTERPRETADOR DE FÓRMULAS</i> .....	2
1. <i>DIFERENÇAS ENTRE A VERSÃO 3.17 E A NOVA VERSÃO</i> .....	2
2. <i>PROCEDIMENTOS PARA A CONVERSÃO</i> .....	6
<i>CONVERSÃO MANUAL</i> .....	9
1. <i>CONVERSÃO DE TIPOS</i> .....	9
2. <i>CONVERSÃO DE FUNÇÕES</i> .....	9
3. <i>CONVERSÃO DE SCRIPTS DO MEGA INTELLIGENCE</i> .....	9
4. <i>ERROS QUE PODEM ACONTECER</i> .....	11



# Manual do Interpretador de Fórmulas

## Introdução

- ✓ O novo interpretador de fórmulas é um interpretador que está incorporado ao Mega Empresarial e já não depende mais da DLL VBScript.dll, por esse motivo a sintaxe das fórmulas mudou. No entanto, para se tentar manter alguma compatibilidade com a versão 3.17, alguns dos comandos do VB Script foram incorporados ao novo interpretador.
- ✓ Mesmo com essa incorporação, as fórmulas utilizadas na versão 3.17 não são compatíveis com a nova versão, sendo assim, é obrigatória a conversão das fórmulas para que processos do Mega que utilizam fórmulas funcionem corretamente. Os comandos portados e as diferenças de sintaxe podem ser vistas no tópico “Diferenças entre a versão 3.17 e a nova versão”.
- ✓ No novo interpretador as fórmulas devem respeitar a uma estrutura básica, conforme exemplo abaixo:

### **Exemplo 1:**

```
function AloMundo
```

```
    Dim vAloMundo /* declara todas as variáveis que serão utilizadas na fórmula */
```

```
        vAloMundo = “Alo Mundo!”
```

```
    return vAloMundo /* sempre devemos possuir o retorno da fórmula */
```

```
end function
```



MEGA

Empresarial

## Novo Interpretador de Fórmulas

### 1. Diferenças entre a versão 3.17 e a Nova Versão

- ✓ Uso obrigatório da palavra reservada "Return" para devolver o resultado da função.

#### Interpretador até versão 3.17

```
Ex: Function FUNC_A  
    FUNC_A = ""  
End Function
```

#### Interpretador versão 4.0

```
Function FUNC_A  
    return ""  
End Function
```

- ✓ Declaração obrigatória das variáveis usadas na função através da palavra reservada "DIM".

#### Interpretador até versão 3.17

```
Ex: Function FUNC_A  
    FUNC_A = vTexto  
End Function
```

#### Interpretador versão 4.0

```
Function FUNC_A  
    Dim vTexto  
    return vTexto  
End Function
```

- ✓ Não é permitido utilizar uma variável com o mesmo nome da função. Pois, caso alguma fórmula se apresente desta forma, no momento do cálculo, será apresentado erro de referência circular.

#### Interpretador até versão 3.17

```
Ex: Function FUNC_A  
    FUNC_A = 1  
    FUNC_A = FUNC_A + 1  
End Function
```

#### Interpretador versão 4.0

```
Function FUNC_A  
    Dim vFunc  
    vFunc = 1  
    return vFunc + 1  
End Function
```

- ✓ Se houverem mais de uma condição no comando "IF" deve-se usar parênteses entre as condições.

#### Interpretador até versão 3.17

```
Ex: if vTexto = "" or vTexto=""
```

#### Interpretador versão 4.0

```
if (vTexto="") or (vTexto="" ) then
```

- ✓ Ao chamar outra fórmula dentro de outra fórmula, não colocar "FUNC."+<nome da fórmula> antes do nome da fórmula.

#### Interpretador até versão 3.17

```
Ex: Function FORMULA_A  
    vTexto = FUNC. FORMULA_B  
    FORMULA_A = vTexto
```

#### Interpretador versão 4.0

```
Function FORMULA_A  
    vTexto = FORMULA_B  
    return vTexto
```

# Manual do Interpretador de Fórmulas

- ✓ Comentário parcial de linha que antes era com aspas simples(') agora mudou para /\* \*/

**Interpretador até versão 3.17**

**Interpretador versão 4.0**

Ex: **if** True **then** ' teste comentario

**if** True **then** /\* teste comentario\*/

- ✓ No comando "with" as propriedades do objeto não devem iniciar com ponto (.).

**Interpretador até versão 3.17**

**Interpretador versão 4.0**

Ex: **with** RH

**with** RH

ANO = .F\_FormataData("YYYY", DT)      ANO = F\_FormataData("YYYY", DT)

MES = .F\_FormataData("MM", DT)      MES = F\_FormataData("MM", DT)

**End with**

**End with**

- ✓ A forma de como se criar e destruir um objeto "TTabela" foi alterada, na criação não se usa mais a palavra reservada "Set" e para destruir se usa o método "Free"

**Interpretador até versão 3.17**

**Interpretador versão 4.0**

Ex: **SET** T=TTABELA.CREATE()

**Dim** T  
T = TTabela  
T.Create(RH)

T.Destroy      T.Free

- ✓ O nome do DataModule ao qual pertence o DataSet não precisa ser especificado no novo interpretador

**Interpretador até versão 3.17**

**Interpretador versão 4.0**

Ex: **RH.T\_Colaborador**.Close

T\_Colaborador.Close

**RH.T\_Colaborador**.Open

T\_Colaborador.Open

- ✓ O objeto Mega2000 passou a se chamar Mega.

**Interpretador até versão 3.17**

**Interpretador versão 4.0**

Ex: Mega2000.Bissexto(2008)

Mega.Bissexto(2008)

- ✓ Comparações de uma variável não inicializada com vazio("") ou 0, no novo interpretador as variáveis não inicializadas contém o valor Null.

**Interpretador até versão 3.17**

**Interpretador versão 4.0**

Ex: **if** vTeste = "" **then**

**if** vTeste = **Null** **then**

vResult = 0

vResult = 0

**else**

**else**

vResult = 1

vResult = 1

**end if**

**end if**

- ✓ Valor literal de data entre sustenidos (#) não é suportado na versão do novo interpretador.

**Interpretador até versão 3.17**

**Interpretador versão 4.0**

Ex: vData = #2/12/08#

vData = StrToDate("2/12/08")

# Manual do Interpretador de Fórmulas

---

- ✓ Soma entre tipos texto e numérico não é suportado na nova versão

**Interpretador até versão 3.17**

**Interpretador versão 4.0**

Ex: vSoma = 1 + "2"

vSoma = 1 + StrToInt("2") ou

vSoma = 1 + 2

- ✓ No comando "Else IF" o IF não pode ficar na mesma linha que o ELSE.

**Interpretador até versão 3.17**

**Interpretador versão 4.0**

Ex: **if** vTeste = "" **then**

**if** vTeste = "" **then**

vResult = 0

vResult = 0

**else if** vTeste=" " **then**

**else**

vResult = 1

**if** vTeste=" " **then**

**end if**

vResult = 1

**end if**

**end if**

**end if**

As funções nativas do VB que foram adicionadas no novo interpretador são:

<b>DateDiff</b>	<b>Day</b>	<b>Month</b>	<b>Year</b>	<b>WeekDay</b>
<b>Round</b>	<b>Fix</b>	<b>Len</b>	<b>DateAdd</b>	

Obs: Existem duas funções DateDiff disponíveis no interpretador, uma do VBScript e uma do objeto Mega2000. Segue abaixo as diferenças entre elas:

VBScript:

```
Function DateDiff(interval : String; Date1, Date2 : TDateTime) : Word
```

```
Interval = d -> dia
```

```
          m -> mês
```

```
          y -> ano
```

Mega2000:

```
Function DateDiff(Date1, Date2 : TDateTime; iTipo : Byte) : Word
```

```
iTipo = 1 -> dia
```

```
          2 -> mês
```

```
          3 -> ano
```

# Manual do Interpretador de Fórmulas

---

Abaixo segue uma listagem de funções antes usadas no VBScript com sua equivalente no novo Interpretador

## Interpretador até versão 3.17

CINT

Call

=>

=<

SysDate

CBool

DateSerial

MsgBox

CCur

CStr

CDate

## Interpretador versão 4.0

StrToInt

Não existe

>=

<=

Date

Não existe

RH.F\_CriaData

ShowMessage

StrToFloat

FloatToStr

StrToDate

Exemplo das funções *CINT*, *MsgBox*, *CCur*, *CStr* e *CDate*:

## Interpretador até versão 3.17

**Function** Somatoria

vData = *CDate*("10/03/2008")

vNumTexto = "1"

vDecTexto = "2,3"

vSoma=*CCur*(vDecTexto)+*CINT*(vNumTexto)

*MsgBox*(*CStr*(vSoma))

Somatoria = vSoma

**End Funciton**

## Interpretador versão 4.0

**Function** Somatoria

**Dim** vSomatoria, vData, vNumTexto, vSoma, vDecTexto

vData = *StrToDate*("10/03/2008")

vNumTexto = "1"

vDecTexto = "2,3"

vSoma = *StrToFloat*(vDecTexto)+*StrToInt*(vNumTexto)

*ShowMessage*(*FloatToStr*(vSoma))

vSomatoria = vSoma

**Return** vSomatoria

**End Funciton**

Para pesquisar dentro do interpretador devemos pressionar a tecla F3.

Para checar a sintaxe da fórmula, devemos pressionar a tecla F5.

Para executar a fórmula, devemos pressionar a tecla F9.

## 2. Procedimentos para a conversão

A conversão de fórmulas é um procedimento que deve ser executado, quando uma base de dados Mega é atualizada de uma versão 3.17 para uma versão 4.0. Sem essa conversão os processos que dependem das fórmulas não funcionarão nesta nova versão. Para executar a conversão devemos:

- a) Entrar no Cadastro de Fórmulas na versão 4.0 e clicar no botão “Converter” conforme a Figura 1. O botão “Converter” faz a conversão de todas as fórmulas que estão cadastradas, independente de ter realizado algum filtro na tela. Caso sejam encontrados erros durante o processo de conversão, será exibido um log das fórmulas com erros. Ao pressionar o botão “Converter” será apresentado a tela abaixo (Figura 2), informando sim para começar será apresentada uma barra de progresso de conversão.

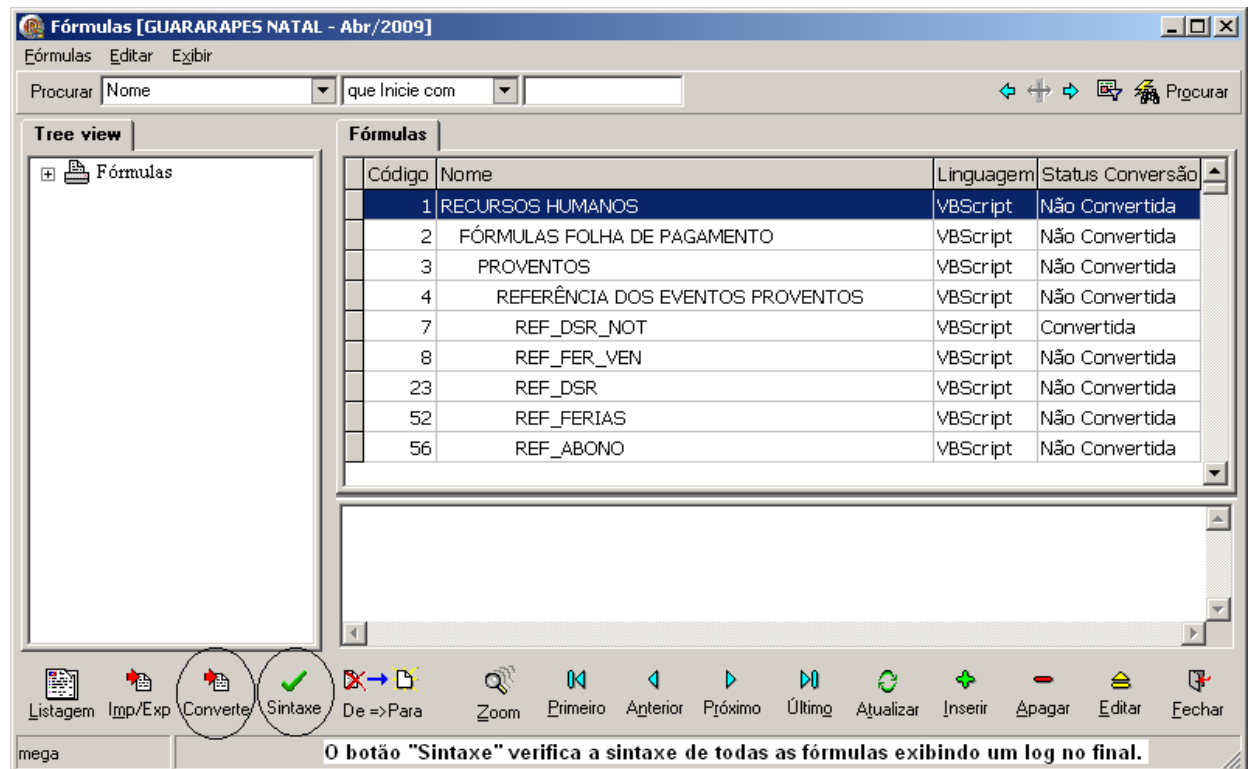


Figura 1 – Cadastro de Fórmulas

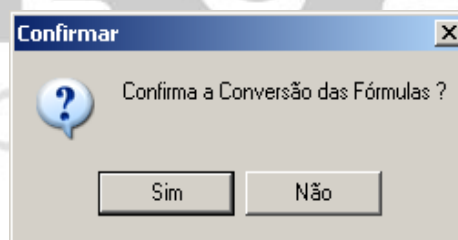


Figura 2 – Tela de confirmação da conversão

## Manual do Interpretador de Fórmulas

- b) Após a conversão será apresentado um log para cada fórmula que apresentou divergência e não foi possível convertê-la automaticamente. Nesta tela (Figura 3) devemos pressionar o botão “Gravar” para que o sistema aplique e grave a conversão executada. Caso escolha a opção “Cancelar” será cancelada a conversão realizada.

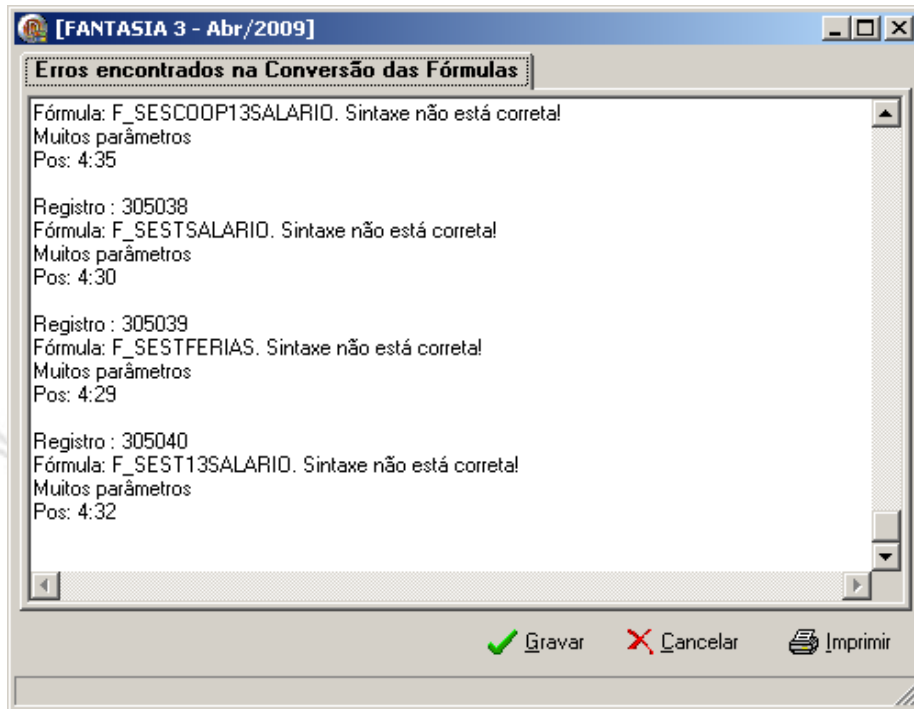


Figura 3 – Log de conversão das Fórmulas

- c) Em seguida, devemos analisar de forma individual, todas as fórmulas apresentadas no log, a fim de fazer os ajustes manuais necessários. Para isso devemos selecionar a fórmula desejada e em seguida pressionar o botão “Editar” conforme Figura 4.

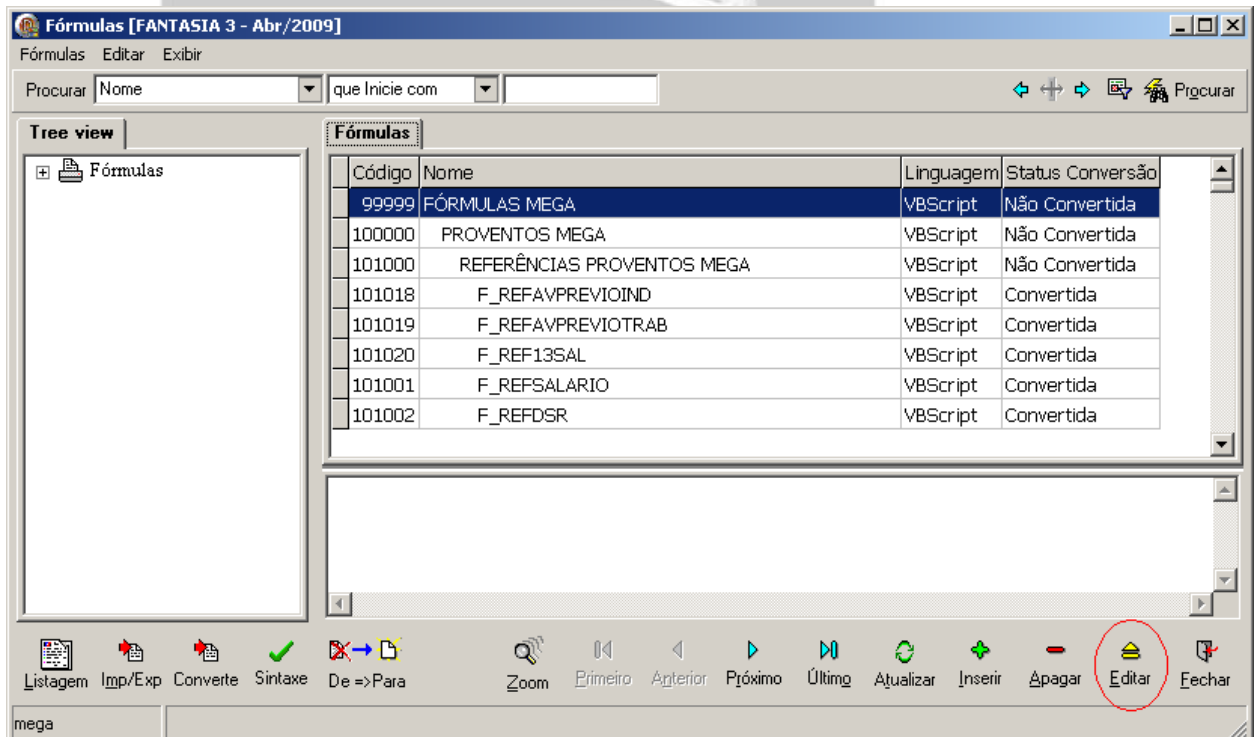


Figura 4 – Cadastro de Fórmulas



# Manual do Interpretador de Fórmulas

- d) Pressionar o botão “Editar Fórmula”, conforme Figura 5.

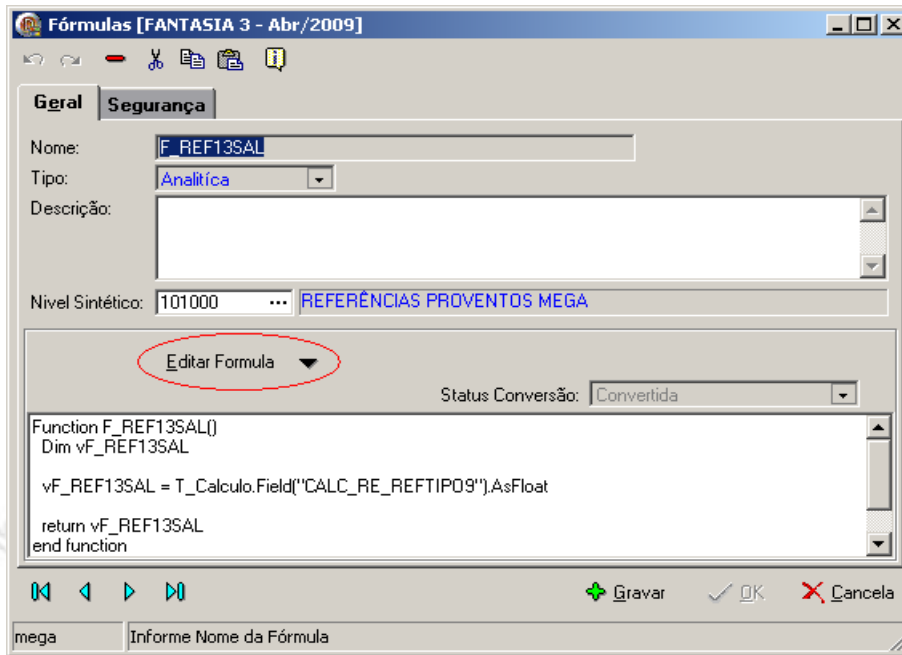


Figura 5 – Edição do cadastro da fórmula

- e) Pressionar a tecla F5 para analisar o erro apresentado. Será exibida a linha e a coluna do erro. Realizar a intervenção necessária e em seguida pressionar F5 novamente para checar se a sintaxe está correta. Para confirmar a alteração pressionar o botão “OK”. Nesta tela também podemos converter automaticamente a fórmula, para isso devemos pressionar o botão “Converter” conforme Figura 6.

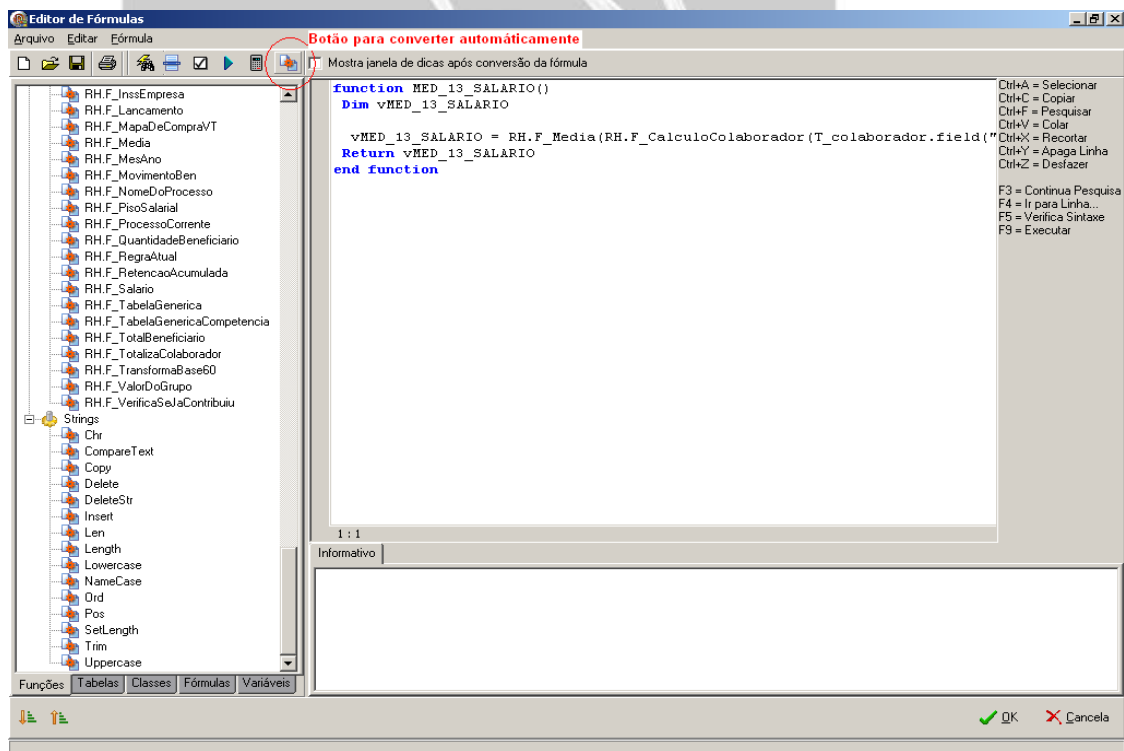


Figura 6 – Edição do conteúdo da fórmula

- f) Devemos repetir os passos de “c” até “e” para todas as fórmulas que não foram convertidas automaticamente pelo sistema.

## Conversão Manual

Ajustes que não são possíveis tratar de forma automática, sendo necessária a intervenção humana.

### 1. Conversão de Tipos

#### Interpretador até versão 3.17

- ✓ Var1 = "1" + 2
- ✓ Var1 = "1" + 2
- ✓ IF FieldByName("DATA").AsDate = ""

#### Interpretador versão 4.0

- Var1 = StrToInt("1") + 2
- Var1 = "1" + IntToStr(2)
- IF FieldByName("DATA").IsNull

### 2. Conversão de Funções

#### Interpretador até versão 3.17

- ✓ Do Loop
- ✓ Repeat Until

#### Interpretador versão 4.0

- While Wend
- While Wend

### 3. Conversão de scripts do Mega Intelligence

Os ajustes abaixo devem ser realizados após a conversão automática (Fig. 7) do script da Tarefa:

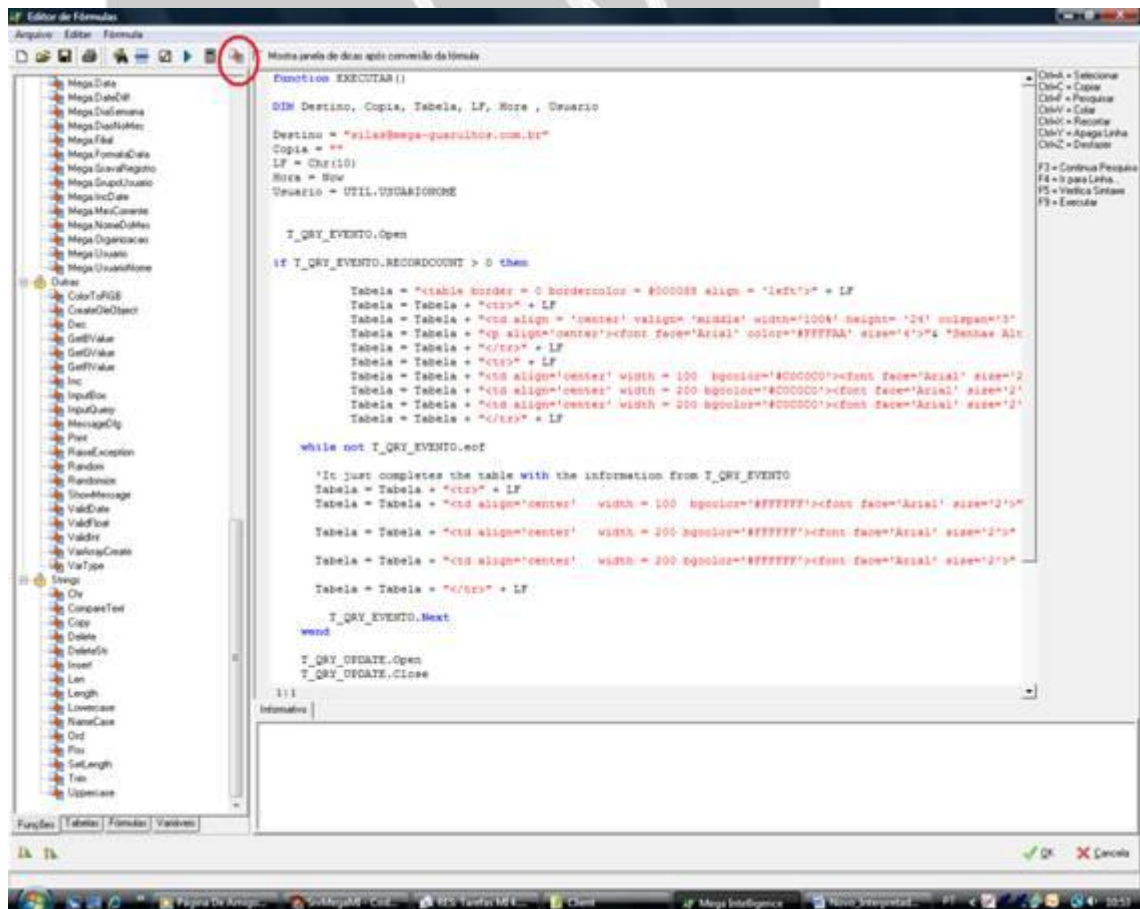


Figura 7 – Botão para conversão automática do script da Tarefa

## Manual do Interpretador de Fórmulas

- a) Não utilizar mais a função **BI.TAG()**, utilizar diretamente a função **BI.TROCATAG()**, conforme o exemplo abaixo:

### Tarefa versão 3.17

```
BI.DOC("HTML")
```

```
Do
```

```
S = BI.TAG("HTML")
```

```
Select Case S
```

```
Case "#PDC_IN_CODIGO"
```

```
    TAG = BI.TROCATAG("HTML", S, T_SELPEDIDOENV("PDC_IN_CODIGO"))
```

```
Case "#FIL_ST_NOME"
```

```
    TAG = BI.TROCATAG("HTML", S, T_SELFILIAL("ORG_ST_NOME"))
```

```
end Select
```

```
Loop Until S = ""
```

### Tarefa versão 4.0

```
BI.DOC("HTML")
```

```
{ BI.TROCATAG("HTML", "#PDC_IN_CODIGO", T_SELPEDIDOENV("PDC_IN_CODIGO"))
```

```
{ BI.TROCATAG("HTML", "#FIL_ST_NOME", T_SELFILIAL("ORG_ST_NOME"))
```

- b) O acesso aos campos e aos parâmetros do componente “**Tabela**” foi alterado, conforme exemplo abaixo:

### Tarefa versão 3.17

```
T_SELFILIAL("ORG_ST_NOME")
```

```
T_SELPEDIDOENV.Param("ORG_IN_CODIGO")
```

### Tarefa versão 4.0

```
T_SELFILIAL.Field("ORG_ST_NOME").AsString
```

```
T_SELFILIAL.ParamByName("ORG_IN_CODIGO").AsInteger
```

- c) As funções que estavam no objeto **Util** foram para o objeto **Mega**, conforme exemplo abaixo:

### Tarefa versão 3.17

```
vNomeUsu = Util.UsuarioNome
```

```
vMes = Util.MesCorrente
```

### Tarefa versão 4.0

```
vNomeUsu = Mega.UsuarioNome
```

```
vMes = Mega.MesCorrente
```

- d) A palavra reservada **Const** foi substituída por **Dim**.

## Manual do Interpretador de Fórmulas

---

- e) O **caractere barra** “\” dentro de uma string literal, normalmente usado em caminho de arquivo, foi substituído por dois **caracteres barra** “\\”.

**Tarefa versão 3.17**

vPath = “\Servidor\Formatos”

**Tarefa versão 4.0**

vPath = “\\Servidor\Formatos”

- f) Houve algumas mudanças com relação a **Objetos**. Não é mais necessário o uso da palavra reservada **Set**. A função **CreateObject** foi substituída por **CreateOleObject**. O objeto **Nothing** foi substituído por **Null**.

**Tarefa versão 3.17**

Set vObject = *CreateObject*(“Objeto”)

Set vObject = *Nothing*

**Tarefa versão 4.0**

vObject = *CreateOleObject*(“Objeto”)

vObject = *Null*

- g) O **For Each** não existe mais, deve ser substituído por **For**.

**Tarefa versão 3.17**

**For Each** vItem in Colecao

vItem.Metodo()

**Next**

**Tarefa versão 4.0**

**For** vIndice = 1 **to** Colecao.Count – 1

Colecao.Item(vIndice).Metodo()

**Next**

- h) As seguintes funções foram substituídas:

**Tarefa versão 3.17**

InStr

Mid

**Tarefa versão 4.0**

Pos

Copy

## 4. Erros que podem acontecer

- ✓ Mensagem: Invalid File Format: isso significa que há algum erro na estrutura da sua fórmula e então não é possível compilar a mesma.
- ✓ Parênteses em excesso: isso pode acontecer devido à implementação de colocar parênteses nos ifs das fórmulas.

Alguns dos possíveis erros de sintaxe das fórmulas após a conversão são:

- ✓ **Parâmetros insuficientes** -> Este erro ocorre geralmente quando alguma função está com um número de parâmetros diferente do que a função espera.
- ✓ **Identificador não encontrado** -> Este erro ocorre quando tem espaços no nome da fórmula ou a fórmula tem alguma variável array declarada usando parênteses para definir o tamanho. No novo interpretador deve-se usar colchetes em arrays.
- ✓ **O lado esquerdo não pode ser atribuído** -> Este erro ocorre quando o nome da fórmula no registro da fórmula está diferente do nome na própria fórmula.
- ✓ **Tipos incompatíveis: 'String', 'Extended'** -> Este erro pode ocorrer devido a uma conversão de data que era suportada no VBScript, mas que não é no novo interpretador. O comando "CDate" do VBScript aceita como parâmetro um texto ou uma data. Na conversão o "CDate" é alterado para "StrToDate" e o "StrToDate" só suporta como parâmetro um texto.
- ✓ **Muitos parâmetros** -> Ocorre quando uma variável está na fórmula como se fosse função, ou seja, tem na frente parênteses e parâmetros.

