



**Funcionamento e configuração
de operações customizadas no
cadastro de serviços x tarefas
(WMSA070)**

**TOTVS Logística WMS –
Linha Protheus**

20/05/2021



FUNCIONAMENTO E CONFIGURAÇÃO DE OPERAÇÕES CUSTOMIZADAS NO CADASTRO DE SERVIÇOS X TAREFAS (WMSA070)



1. Visão geral

O cadastro de **Serviços x Tarefas (WMSA070)** permite configurar os serviços que serão informados obrigatoriamente em todos os documentos que geram integração com o WMS Protheus.

Por padrão, estes serviços são a base para a geração dos registros que possibilitam a movimentação de estoque dos produtos e execução de processos complementares - como conferência, montagem de volumes, distribuição de produtos, entre outros - dentro do armazém.

Todo serviço possui ao menos uma tarefa que, por sua vez, deve estar vinculada diretamente a uma operação.

As operações que estão disponíveis atualmente são:

- 0 - Customizada
- 1 - Endereçamento
- 2 - Endereçamento *Crossdocking*
- 3 - Separação
- 4 - Separação *Crossdocking*
- 5 - Reabastecimento
- 6 - Conferência Entrada
- 7 - Conferência Saída
- 8 - Transferência

Os processos de **1 a 8** são funcionalidades padrão do sistema, que englobam diversas regras de negócio e boas práticas de mercado para a geração automática dos movimentos de distribuição, no momento da execução de um serviço WMS Protheus.

Estes movimentos podem ou não ser executados posteriormente por meio de coletores de dados de radiofrequência.

Devido à necessidade que alguns usuários possuem de executar regras ou rotinas específicas, que não existem ou não são contempladas pelo produto padrão, o WMS Protheus também disponibiliza a operação **0 (zero)**, que é executada como qualquer outra, ou seja, no mesmo momento e da mesma forma que as demais operações.

A diferença é que todas as regras de negócio e geração dos movimentos permanece sob responsabilidade do desenvolvedor da rotina específica, pois o sistema apenas realiza a chamada da função customizada.

Importante

Essa documentação é válida apenas para o **WMS com controle de estoque exclusivo (D14)**.

2. Cadastro de tarefa customizada

O cadastro de uma tarefa customizada, por ser realizado independentemente do tipo de serviço, pode ser uma saída, uma entrada ou um movimento interno.

Além disso, um serviço pode ser composto apenas por tarefas vinculadas a operações padrão, somente por tarefas de operação customizada ou misto.



FUNCIONAMENTO E CONFIGURAÇÃO DE OPERAÇÕES CUSTOMIZADAS NO CADASTRO DE SERVIÇOS X TAREFAS (WMSA070)



Funcão Exéc	Req. Saída	Operação	Desc. Operação	Req. Serviço	Mult. Ativ	Teste
Não		3	SEPARAÇÃO	Não	Automático/Normal	
Não		0	CUSTOMIZADO		Não Utiliza	

Sempre que a operação **0 - Customizada** for atribuída a uma tarefa é necessário informar também a **Função a Executar (DC5_FUNEXE)**.

Em versões anteriores do sistema este campo era utilizado como base para execução de todos os serviços WMS.

Cod. Tarefa	Desc. Tarefa	Funcão Exéc	Req. Saída	Operação	Desc. Operação
302	APANHADA DE PRODUTOS		Não	3	SEPARAÇÃO
303	CONFERENCIA		Não	0	CUSTOMIZADO



FUNCIONAMENTO E CONFIGURAÇÃO DE OPERAÇÕES CUSTOMIZADAS NO CADASTRO DE SERVIÇOS X TAREFAS (WMSA070)



The screenshot shows the 'Serviços X Tarefas - ALTERAR' form in the TOTVS WMS system. The main form contains fields for 'Cod. Serviço' (001), 'Desc. Serviço' (EXPEDICAO), 'Tipo Serviço' (2 - Saída), and 'Tipo Exec.' (1 - Manual). A modal window titled 'Consulta Padrão - Tabelas do Sistema' is open, displaying a table of system tables:

Chave	Descrição
000001	DLEndereca()
000002	DLApanhe()
000003	DLGvAbast()
000004	DLMontKit()
000005	DLConferen()
000006	DLCrossDoc()
000007	DLTransfer()

Importante

Para que a função a executar seja exibida na consulta padrão do campo é necessário cadastrá-la por meio do módulo **Configurador**, no arquivo **SX5 - Tabelas Genéricas (tabela L6)**.

3. Execução

Toda função customizada receberá por parâmetro o objeto **oMovimento** e a variável **cAcao**, nesta ordem, onde:

- **Movimento:** instância da classe WMSBCCCustomizacao que é derivada (herda as informações) da classe WMSDTCMovimentosServicoArmazem.
- **Acao:** ação que está sendo executada no momento da chamada da função customizada. Quando o valor recebido for igual a 1 (um), quer dizer que a função está sendo chamada no momento da execução de uma ordem de serviço; quando o valor recebido for igual a 2 (dois), quer dizer que a função está sendo chamada pela rotina de convocação ativa do coletor de dados.

Não é obrigatório que as regras de negócio implementadas para a ação 1 (um) gerem movimentos de distribuição (D12), e nem que a ação 2 (dois) execute algum processo, porém, a customização sempre deve tratar as duas ações para que o sistema não apresente inconsistências em tela.

Exemplo

A ação 1 (um) poderia gerar dados em uma tabela específica, os quais seriam considerados por uma rotina acessada através do menu do ACD; ou até mesmo executar regras de negócio que não teriam relação com um processo de coletor e encerrar a customização neste ponto. Porém, ambas as situações deverão prever a chamada da ação 2 (dois), que neste caso poderia apenas retornar verdadeiro (.T.).



FUNCIONAMENTO E CONFIGURAÇÃO DE OPERAÇÕES CUSTOMIZADAS NO CADASTRO DE SERVIÇOS X TAREFAS (WMSA070)



```
-----  
#INCLUDE "RWMAKE.CH"
```

```
/*
```

```
+-----+  
|User Function Customizada -> Recebe dois parâmetros, onde: |
```

```
+-----+  
|oMovimento | Objeto que contém as informações da movimentação |  
| | Este objeto pode ser uma das instâncias das classes padrão: |  
| | Onde: |  
| | cAcao == '1' -> WMSBCCCustomizacao() herdada de da classe |  
| | WMSDTCMovimentosServicoArmazem() |  
| | cAcao == '2' -> WMSDTCMovimentosServicoArmazem() |
```

```
+-----+  
|cAcao | Ponto onde foi chamada a função customizada: |  
| | '1' - Na execução da OS |  
| | '2' - Na execução do Coletor |
```

```
+-----+  
*/
```

```
/*
```

```
*****  
DE -> PARA = aParam150 -> oMovimento  
*****
```

```
aParam150 := Array(34)
```

```
aParam150[01] -> oMovimento:oMovPrdLot:GetProduto() //-- Produto  
aParam150[02] -> oMovimento:oMovEndOri:GetArmazem() //-- Armazém Origem  
aParam150[03] -> oMovimento:oOrdServ:GetDocto() //-- Documento  
aParam150[04] -> oMovimento:oOrdServ:GetSerie() //-- Serie  
aParam150[05] -> oMovimento:oOrdServ:GetNumSeq() //-- Sequencial  
If cAcao == '1'  
aParam150[06] -> oMovimento:GetQuant() //-- Quantidade Ordem Serviço  
Else  
aParam150[06] -> oMovimento:GetQtdMov() //-- Quantidade Movimentação  
EndIf  
aParam150[07] -> Date() //-- Data da Movimentacao  
aParam150[08] -> Time() //-- Hora da Movimentacao  
aParam150[09] -> oMovimento:oMovServic:GetServico() //-- Servico  
aParam150[10] -> oMovimento:oMovTarefa:GetTarefa() //-- Tarefa  
aParam150[11] -> oMovimento:oMovTarefa:GetAtivid() //-- Atividade  
aParam150[12] -> oMovimento:oOrdServ:GetCliFor() //-- Cliente/Fornecedor  
aParam150[13] -> oMovimento:oOrdServ:GetLoja() //-- Loja  
aParam150[14] -> Nil //-- Tipo da Nota Fiscal - Obsoleto
```



FUNCIONAMENTO E CONFIGURAÇÃO DE OPERAÇÕES CUSTOMIZADAS NO CADASTRO DE SERVIÇOS X TAREFAS (WMSA070)



```
aParam150[15] -> Nil //-- Item da Nota Fiscal - Obsoleto
aParam150[16] -> "0" //-- Tipo de Movimentacao - Obsoleto
aParam150[17] -> oMovimento:oOrdServ:GetOrigem() //-- Origem de Movimentacao
aParam150[18] -> oMovimento:oMovPrdLot:GetLoteCtl() //-- Lote
aParam150[19] -> oMovimento:oMovPrdLot:GetNumLote() //-- Sub-Lote
aParam150[20] -> oMovimento:oMovEndOri:GetEnder() //-- Endereco Origem
aParam150[21] -> oMovimento:oMovEndOri:GetEstFis() //-- Estrutura Fisica Origem
aParam150[22] -> oMovimento:oOrdServ:GetRegra() //-- Regra de Apanhe (1=LOTE/2=NUMERO DE
SERIE/3=DATA)
aParam150[23] -> oMovimento:oOrdServ:GetCarga() //-- Carga
aParam150[24] -> oMovimento:GetIdUnit() //-- Nr. do Pallet
aParam150[25] -> oMovimento:oMovEndDes:GetArmazem() //-- Armazém Destino
aParam150[26] -> oMovimento:oMovEndDes:GetEnder() //-- Endereco Destino
aParam150[27] -> oMovimento:oMovEndDes:GetEstFis() //-- Estrutura Fisica Destino
aParam150[28] -> oMovimento:oMovServic:GetOrdem() //-- Ordem da Tarefa
aParam150[29] -> oMovimento:oMovTarefa:GetOrdem() //-- Ordem da Atividade
aParam150[30] -> oMovimento:oMovTarefa:GetFuncao() //-- Funcao do Recurso Humano
aParam150[31] -> oMovimento:oMovTarefa:GetTpRec() //-- Recurso Fisico
aParam150[32] -> oMovimento:oOrdServ:GetIdDCF() //-- Identificador do DCF
aParam150[34] -> oMovimento:GetIdMovto() //-- Identificador exclusivo do Movimento no D12
*/
```

```
User Function DLApona(oMovimento,cAcao)
Local lRet := .T.
```

```
// Executa as ações complementares
// Para efeito de teste esta função vai replicar o registro da ordem de serviço (DCF) para a tabela D12
// Ou seja, irá gerar um registro de movimentação (convocação) exatamente igual à DCF
```

```
If cAcao == '1' // Na execução da OS
// Se for um serviço de tipo 1 - Entrada, atribui o endereço destino igual ao origem,
// pensando que a tarefa customizada será executada ainda na doca de entrada
If oMovimento:oMovServic:GetTipo() == "1" .And. Empty(oMovimento:oMovEndDes:GetEnder())
oMovimento:oMovEndDes:SetEnder(oMovimento:oMovEndOri:GetEnder())
// Se for um serviço de tipo 2 - Saída, atribui o endereço origem igual ao destino,
// pensando que a tarefa customizada será executada apenas na doca de saída
ElseIf oMovimento:oMovServic:GetTipo() == "2" .And. Empty(oMovimento:oMovEndOri:GetEnder())
oMovimento:oMovEndOri:SetEnder(oMovimento:oMovEndDes:GetEnder())
EndIf
//-- Status movimento
oMovimento:cStatus := '4'
//Copiando a quantidade para o movimento
oMovimento:SetQtdMov(oMovimento:GetQuant())
If !oMovimento:AssignD12()
lRet := .F.
EndIf
```



FUNCIONAMENTO E CONFIGURAÇÃO DE OPERAÇÕES CUSTOMIZADAS NO CADASTRO DE SERVIÇOS X TAREFAS (WMSA070)



```
ElseIf cAcao == '2' // Na execução do Coletor
  IRet := U_AptColUsr(oMovimento) // Chama rotina customizada de coletor
EndIf
```

```
Return IRet
```

```
// Função para simples apontamento de tempo. Abrirá no coletor uma tela para que o
// usuário informe a hora de início da movimentação.
```

```
User Function AptColUsr(oMovimento)
Local IRet := .T.
```

```
// Se o movimento não tiver sido iniciado
If oMovimento:GetStatus() $ '3|4' .And. Empty(oMovimento:GetDataIni())
  If (IRet := WmsQuestion("Confirma o inicio da atividade?"))
    oMovimento:SetRecHum(__cUserID)
    oMovimento:SetStatus("3")
    oMovimento:SetDataIni(Date())
    oMovimento:SetHoraIni(Time())
    oMovimento:UpdateD12()
  EndIf
ElseIf oMovimento:GetStatus() == '3' .And. !Empty(oMovimento:GetDataIni())
  If (IRet := WmsQuestion("Confirma a finalização da atividade?"))
    oMovimento:SetStatus("1")
    oMovimento:SetRecHum(__cUserID)
    oMovimento:SetDataFim(Date())
    oMovimento:SetHoraFim(Time())
    oMovimento:UpdateD12()
  EndIf
Else
  WMSVTAViso("Atividade inconsistente.")
  IRet := .F.
EndIf
```

```
WMSAltSts(.F.) // Informa para o produto padrão que não deve alterar a situação do movimento
```

```
Return IRet
```

4. Assuntos relacionados

Documentos de Referência

- [Serviços Vs Tarefas \(WMSA070 - SIGAWMS\)](#)