

User-Exit – Não deixe as brechas descontroladas – Parte 2 – SMOD/CMOD

20/06/2011 11:14

Dando continuidade à sequência de posts que começou [aqui](#), hoje vamos desvendar mais uma maneira de colocar sua lógica ~~trava-sistemas~~ salva-vidas no meio do processamento Standard. Com vocês, SMOD/CMOD!

Você que já leu outros artigos do ABAPZombie deve estar acostumado: nós tentamos informar de uma maneira divertida, provendo meios para você se aprofundar no assunto, caso tenha interesse. Portanto, vou ir direto ao ponto: esta sequência de artigos não foi feita para explicar em detalhes os motivos que levaram a SAP a criar as mil-e-uma maneiras de implementar user-exits no SAP, a idéia é explicar como utilizar essas diversas técnicas.

Mas, se você for um freak por conhecimento, [leia isso aqui](#), e faça sucesso com as garotinhas(os) sap-geeks do mundo (boa sorte tentando encontrar essa raça).

Enfim, o que diabos são SMOD e CMOD? Transações, é claro! Elas trabalham em conjunto para que você possa criar **Projetos Z para agrupar Ampliações e os seus Componentes**. Aí você pensa, mas heim? Vamos com calma 😊

Transação SMOD: Aqui você tem acesso à todas as **ampliações** que a SAP provê para que você possa colocar seu código Z no meio do processamento Standard. Nessa transação você também pode acessar todos os **componentes de uma ampliação**, que são as funções, áreas de telas, includes e etc, onde você vai codificar de verdade.

Transação CMOD: Onde você irá criar um novo projeto, que pode agrupar várias ampliações.

Ainda está confuso? Tá, eu desenho:

Sacou? Projeto -> Ampliações -> Componentes
(Código Z)

Vamos agora fazer um exemplo prático. Vamos supor que eu queria colocar código em uma exit de Ordens de Vendas. Tenho então, que seguir alguns passos:

1) Encontrar a User-Exit: Esse passo é o mais cruel de todos, muitos ABAPers sofrem para encontrar User-Exits. No caso de exists atreladas à ampliações da SMOD, você pode fazer o seguinte:

– Dar um /H no standard, e colocar um break-point no comando **CALL CUSTOMER-FUNCTION** . Esse comando é o que chama as funções registradas como componentes em apliações da SMOD.

– Usar o **F4 da SMOD**. Parece besta, mas eu já encontrei coisas aqui em minutos, utilizando ali o texto breve. Um *vendas* vai me trazer algumas ampliações de SD, e é uma delas que eu vou usar no exemplo deste artigo. No dia-a-dia, se você colocar uma palavra chave e ter sorte, você pode encontrar a exit que procura! 😊

Essa é a Tela do F4 da SMOD. Clicando no " + " ao lado do fechar...

...você pode ver mais campos para busca.
Eu achei a Ampliação do nosso evento colocando a máscara *vendas* e dando um Ok (enter)

Tem ainda outros meios, e eu irei fazer um post sobre isso no futuro.

Vou usar a ampliaçãoV45P0001 de SD para nosso exemplo.

2) Criar o Projeto na CMOD: Você deve criar um projeto Z para agrupar ampliações. Faça o seguinte:

Acesse a transação SMOD, dê um nome Z para o projeto.

Clique em Criar, coloque um nome, dê um pacote e etcs. Salve atribua uma request e etcs. E etcs, etcs, etcs... Projeto criado? Então temos que...

3) Adicionar as ampliações da SMOD: Dentro da tela do Projeto, clique em modificar, e depois em “Ampliação atribuição”

Exibição do Projeto após a criação. Clicar no botão destacado, com a descrição Mestre YODA-Style

Aqui, vamos colocar a ampliação que encontramos na SMOD e Salvar.

4) Implementar os componentes da ampliação: Ainda dentro da tela de ampliações do seu projeto, coloque a linha sobre a ampliação que acabamos de inserir e clique em “Componentes”. Você vai ver a tela abaixo:

Vamos colocar um código nesse componente. Um duplo-clique no nome da “Exit da Função”, vai te levar pra essa tela aqui:

Viu ali o nome de uma include Z? Apesar de ela estar codificada, ela ainda não existe. Vamos criá-la:

Opaaa mas peraí.. você deu o duplo-clique e ele não criou? Apareceu essa mensagem em amarelo? Eu vou te responder: E DAÍ QUE APARECEU EM AMARELO? É WARNING, APERTE ENTER E CONTINUE COM O PROCESSO. Desculpem o rage caps lock, mas por várias vezes eu tive que explicar pra pessoas que trabalham com SAP há mais de 2 anos que mensagens de warning são AVISOS, você lê, entende, compreende, guarda na memória e depois aperta ENTER e continua com a sua vida, feliz e contente. Voltando à nossa include Z...

Depois de criar a INCLUDE, você pode colocar o código que você quiser, de preferência dentro de um IF para seu usuário, de modo que o seu código não atrapalhe a vida de ninguém que não seja você mesmo:

Voltando à tela de Componentes dentro do seu projeto, você vai ver uma tela assim:

Note que o "Impl" agora tem um check verde, indicando que aquela exit tem um código. Notou também que tem botões de Ativar ali? Pois é aqui que você vai ver a magia dessa bagaça toda: você pode ativar/desativar componentes individualmente em suas ampliações. Cool, falaí? Agora se o seu código for destruir a produção, é só desativar e todo mundo fica feliz denovo 😊

5) Ativando sua implementação Z: Clique na linha da Exit da Função que criamos ali em cima, e clique no botão de Ativar. Você verá a tela abaixo:

Que indica que você está pronto para testar o código Z que você criou dentro da transação standard VA01.

E pronto, agora é pura diversão no SAP 😊

Bom galera, esse foi um overview de como a CMOD/SMOD são estruturadas, e de como implementar uma exit organizada nessas transações. Se você prestou atenção, viu que na parte de componentes, não temos só funções,

temos também includes e áreas de tela. O processo para implementar é praticamente o mesmo, mas, qualquer dúvida, poste nos comentários que vamos tentar ajudar, ok?

Abraços!

Comentários

Jailson — 14/09/2021 07:59

Muito bom!! Obrigado pela desmistificação do tema!

Marcelo — 12/08/2020 16:23

Muito bom! Valeu pela ajuda!

Celson Ramos Neves — 23/05/2016 11:07

Fiz todo o procedimento, porém, quando transporto para o ambiente de qualidade, as ampliações criadas na SMOD não estão na CMOD, apesar da include existir ativa.

Para inserir as ampliações da SMOD na CMOD em qualidade é somente por ambiente?

aline canteiro — 18/02/2016 15:28

Muito obrigada ^^
sempre salvando !

Priscila Ramos — 31/01/2014 16:28

Dica para achar o nome da aplicação pela SMOD (com o nome da EXIT); Acesse a aba de características da exit(SE37), em dados gerais copie o nome do pacote e informe no help(F4) da SMOD. Exemplo: EXIT (EXIT_SAPLRHIN_001) Pacote (SP00) Ampliação (RHIV0001).

Pri 😊

Carlos Melo — 15/07/2013 15:03

Parabens, pessoal....

Bruno Vianna — 15/01/2013 14:53

Muito bom post Maurício me ajudou muito, obrigado.

Vinicius Ostan — 13/10/2011 12:56

Quero agradecer novamente,
Este post virou referencia quando esqueço de algum pequeno passo ao implementar exit.

Vinicius ostan — 04/07/2011 07:46

Muito bom Mauricio, ótimo post,
Ajudou de +++ !!!

Renato Bonini — 22/06/2011 06:52

Maurício.... salvando algumas pessoas...

Só prá adicionar outro modo de encontrar user EXIT, que eu acho muito simples e rápido, é pegar o programa Base de uma Transação e fazer uma busca na SE37.

Por exemplo: transação VA01 – programa base SAPMV45A (Você pode pegar este programa pela SE93).

Entra na SE37 e faz uma busca usando pelo nome da função com `"*SAPMV45A*"`.

Todas as funções que retornarem `"EXIT_XXXXXX"` são EXITS que você pode utilizar.

Exemplo do retorno desta busca:

XOIK IS-Oil TAS/TPI user exits

EXIT_SAPMV45A_910 IS-OIL/TAS: Exit Change-Flag Output determination

EXIT_SAPMV45A_911 TAS – Customer exit after creation of calloff

EXIT_SAPMV45A_920 User Exit: TAS data Incompletion Log on Doc.-header-level

EXIT_SAPMV45A_930 Userexit to allow overdelivery in Quantity schedule

XVVA User Exit Order

EXIT_SAPMV45A_001 User Exit for Determining the Billing Plan Type

EXIT_SAPMV45A_002 Preassignment of Sold-to Party in Sales Documents

EXIT_SAPMV45A_003 Rev.Rec.: Copy Requirements An Header Level

EXIT_SAPMV45A_004 Rev.Rec.: Field Modification Sales

EXIT_SAPMV45A_005 Copyy Packing Proposal Into Outobund Delivery Orders

Mauro Laranjeira — 22/06/2011 09:15

Opa..

Valeu Bonini, toda contribuição é sempre bem vinda XD

Abraços