

Generate REXX with an XML to update the appl in use flag

- [<Use cases>](#) |
- [<XML services in CMN/ZMF>](#) |
- [<CMN/ZMF Application Administration>](#) |
- [<CMN/ZMF Helpdesk>](#)

Summary: Develop a REXX procedure using appropriate XML service(s) for updating the "application in use" indicator (e.g. in a batch job during ChangeMan ZMF housekeeping).

Related Topics: [Find the XML service to update the appl in use flag](#)

Scenario:

Step 01: Initiate SERXMLRC REXX code generation for XML service PARMS / APPL / UPDATE

```
ASCI - AbitMORE® SCM Commander 1.4.1 ----- Row 1 from 2
Home   > byRequest (CMN611)
Command =>                                     scroll => CSR

Execute : EX execute xml service
View     : VI view request tags  VO view result tags  VC view cobol copybook
Generate : GA AbM XML Framework  GS Serxmlrc code    GX serxmlbc XMLIN
Compare  : CX Compare XML services

Flt= * _____ * _____ * _____ * _____ *

Cmd  Service  Scope  Message  Color  Copybook  Category
---  -
gs   PARMS    APL    CREATE   YELLOW  XMLYAPRM  APL
gs   PARMS    APL    UPDATE   YELLOW  XMLYAPRM  APL
***** Bottom of data *****
```

Step 02: SERXMLRC REXX code generation - Specify REXX vars for request tags (1/2)

```
ASC1 - AbitMORE® SCM Commander 1.4.1 ----- Row 1 to 21 of 65
Home   > byRequest (CMN611) > SER REXX Generator 1/3 - Request tags
Command =>                               Scroll => CSR

Press <ENTER> to process, <END> to return

Service: PARS      Scope: APL      Message: UPDATE

List of valid input tags      Lvl REXX variable name
-----
applName                     1   @AppId_
applDesc                     1   _____
skelsReleaseId               1   _____
businessFromTime             1   _____
businessToTime               1   _____
auditLevel                   1   _____
chkOutRule                   1   _____
chkOutEntity                 1   _____
stageLimitLevel              1   _____
stageEntity                  1   _____
promotionRule                1   _____
packageAgingPeriod           1   _____
stageLibAgingPeriod          1   _____
componentAgingPeriod         1   _____
defaultUnitName              1   _____
defaultVolume                1   _____
allowStageOverlay            1   _____
createCmpWorkRecs            1   _____
showUserPanels               1   _____
allowOnlyOneApproval         1   _____
```

Step 03: SERXMLRC REXX code generation - Specify REXX vars for request tags (2/2)

```
ASC1 - AbitMORE® SCM Commander 1.4.1 ----- Row 47 to 65 of 65
Home   > byRequest (CMN611) > SER REXX Generator 1/3 - Request tags
Command =>                               Scroll => CSR
```

Press <ENTER> to process, <END> to return

Service: PARMS Scope: APL Message: UPDATE

List of valid input tags Lvl REXX variable name

List of valid input tags	Lvl	REXX variable name
allowLinkPackages	1	
processPartPackageByInsDate	1	
forcePackageAudit	1	
useZPrefixForBatchJobs	1	
suppressMsgsForINSJobs	1	
onlyMemoDelEmptyPackages	1	
jobCard01	1	
jobCard02	1	
jobCard03	1	
jobCard04	1	
nextPackageNumber	1	
infoRule	1	
isApplLocked	1	
isApplBusy	1	@BusyYN
packageBindLibType	1	
inUseOwner	1	
defaultScheduler	1	
cmnaudrcEntity	1	
auditLockPackage	1	

***** Bottom of data *****

Step 04: SERXMLRC REXX code generation - Specify execution parms

```
ASC1 - AbitMORE® SCM Commander 1.4.1 ----- No output tags
Home   > byRequest (CMN611) > SER REXX Generator 3/3 - ExecuteParms
Command =>                               Scroll => CSR
```

Press <ENTER> to process, <END> to return

Service: PARMS Scope: APL Message: UPDATE

CMN/ZMF subsys.: T CMN/ZMF subsystem ID

Coding style....: S (A=AbitMORE style, S=Serena full code)

Additional code information

CMN/ZMF subsys.: ascCmSys variable indicating the CMN/ZMF subsystem ID
XML Test.....: N (Y=Yes, N=No)
Stem variable...: SER1 name of stem variable that will be used

Generated code data set

Data set name...: U87462.ASC.DATA.CODESER.EXEC
Member.....: APPLLOCK membername in which code will be saved

Step 05: SERXMLRC REXX code generation - Review generated REXX code (1/2)


```
***** ***** Top of Data *****
000001 /* REXX */
000002 /*=====
000003 * AbitMORE® SCM Commander 1.4.1 AbitMORE REXX generator *
000004 *=====*/
000005
000006 @AppId="?????"
000007 @BusyYN="?????"
000008 ascCmSys="T"
000009 tsoUser=userid()
000010
000011 /* initialize stem variable */
000012 stem="SER1."
000013 drop SER1. result Subsys Userid Product Service Scope Message
000014 SER1.= ""
000015
000016 /* initialize header variables */
000017 SER1.Subsys=ascCmSys
000018 SER1.Userid=userid()
000019 SER1.Product="CMN "
000020 SER1.Service="PARMS"
000021 SER1.Scope="APL"
000022 SER1.Message="UPDATE"
000023
000024 /* initialize request variables */
000025 SER1.appName=@AppId
000026 SER1.isAppBusy=@BusyYN
```

Step 06: SERXMLRC REXX code generation - Review generated REXX code (2/2)

```
000027
000028 /* call serxmlrc */
000029 address LINKMVS "SERXMLRC stem"
000030 if rc<>0 then do
000031     say "SERXMLRC failed - rc:"rc
000032     say "-Subsys      : "SER1.Subsys
- - - - - 4 Line(s) not Displayed
000037     say "-statusMessage: "SER1.statusMessage
000038     call Disconnect
000039     end
000040 /* Disconnect from CMN/ZMF */
000041 call Disconnect
000042 exit
000043
000044 /*-----*/
000045 Disconnect: /* Disconnect from CMN/ZMF
000046 -----*/
000047 /* initialize stem variable */
000048 stem="SER0."
000049 drop SER0.
000050 SER0=""
000051 /* initialize header variables */
000052 SER0.Subsys=ascCmSys
000053 SER0.Userid=userid()
- - - - - 10 Line(s) not Displayed
000064 return
***** ***** Bottom of Data *****
```

Source URL (retrieved on 19/04/2024 - 22:44): <http://www.abitmore-scm.com/products/commander/usecases/generate-rexx-to-reset-isApplBusy>