# Use ASZ$$VAR in any CMN/ZMF submitted job

- [Z-Recipes] |
- [ZMF Administrator] |
- [Helpdesk operator] |
- [Public] |
- [1.0-Orval]

Z-Recipe**License:** Creative Commons Attribution-ShareAlike license agreement
Release info**Release date:** Mon, 05/09/2011
**ChangeMan ZMF Compatibility:** Any ChangeMan ZMF release

ChangeMan ZMF uses the CMN$$VAR skeleton to set various skeleton variables used within staging jobs (only). This solution introduces a similar skeleton, ASZ$$VAR, which can be used (embedded) in any phase in the ChangeMan ZMF lifecyle for which a job is submitted.

Skeleton ASZ$$VAR refers to a skeleton variable **&ASZSKLNM** (= 'skeleton name') with possible values like:

- CMN$$CKO (checkout)
- CMNSTAGE (stage)
- CMN$$RPM (promote or demote)
- CMN10 (distribute)
- CMN20 (install)
- CMN20T (temp install)
- CMN30 (baseline ripple)
- CMN31T (TCC)
- CMN50 (backout)
- CMN55 (reverse baseline ripple)

That way, the value of &ASZSKLNM reflects the phase in the ChangeMan ZMF lifecyle for which a job is submitted.

You can then use (embed) ASZ$$VAR in skeletons like CMN$$CKO, any staging skeleton, CMN$$PRM, CMN10, CMN20(I), CMN30(I), CMN50(I) and CMN55(I). Based on the value of &ASZSKLNM, you can then have ASZ$$VAR set any desired skeleton variable and assign it the appropriate value.

If needed, you can further extend this technique to DB2 option skeletons also (CMN21, CMN49, CMN32, CMN56).

A variation of this technique can be found in Add restart instructions in Xnode jobs.
Example
In the beginning of CMN20 skeleton, add these lines:

```
)CM
)SET ASZSKLNM = ASZ20
)CM
)IM ASZ$$VAR
)CM
```



Copyright © 2000-2025 AbitMORE ® BV. All rights reserved

Part of the ASZ$$VAR skeleton might look like this:

```
...
)SEL &ASZSKLNM EQ ASZ20
)CM
)SET ASZNODAG = ABMTST
)CM
)CM*******************************************************
)CM  ASZLDSAJ : DSN CONTAINING APSSCRN (GEN INPUT)
)CM  ASZLDSAM : DSN CONTAINING SCRSYMB (GEN OUTPUT)
)CM  ASZLDSAC : DSN CONTAINING MFS SRCS (GEN OUTPUT)
)CM*******************************************************
)CM
)SET ASZLDSAJ = &ASZNODAG..APS.APSSCRN
)SET ASZLDSAM = &ASZNODAG..APS.SCRSYMB
)SET ASZLDSAC = &ASZNODAG..MFS.IMS
)CM
)ENDSEL &ASZSKLNM EQ ASZ20
...
```

To better understand the sample above (and value of the proposed solution), a few more comments on this:

- the skeleton vars &ASZLDSAJ, &ASZLDSAM and &ASZLDSAC represent some system DSNs (in this case, related to APS, but it could be anything like DB2 DSNs, IDMS DSNs, CICS DSNs, etc).
- imagine you have multiple remote sites to install a package, but for the remote sites (on the P-site) these system DSNs are different from the system DSNs on the DP-site.
- using the test "&ASZSKLNM EQ ASZ20" (which is like "am I doing an FTINCL of CMN20(I)"), you can then produce the exact DSNs to be used at these remote sites.

Related topics

1. Common skeletons, includes, EXECs, etc (which will be enhanced to also include such an ASZ$$VAR skeleton).
2. Use custom ISPF tables in CMN/ZMF's FTINCL includes another technique which is a bit tougher to implement, but definitely much easier to use/maintain.
3. Analyse the content of a package contains a sample of how this technique is used.

**Source URL (retrieved on *2026-11-11 01:25*):** http://dr.chgman.com/z-wiki/z-recipes/s021



Copyright © 2000-2025 AbitMORE ® BV. All rights reserved