Secure CMN/ZMF panel commands via entity checks

- [SCMBeans] |
- [ZMF Administrator] |
- [IT Auditor] |
- [1.1-Chimay]

Release info**Release date:** Mon, 24/09/2012
Applicable ChangeMan ZMF: Any ChangeMan ZMF release

There are some ChangeMan ZMF functions in ChangeMan ZMF's ISPF dialog that are typically NOT allowed to be used by just any developer, e.g. because they may impact the integrity of ChangeMan ZMF managed projects. Some examples are:

- Revert a package back to development when the approval process has not been started (= not even 1 approval has been given), because in that case revert is allowed by anybody with application update access (and the CMNxREVR / CMNREVRT entity is not checked!).
- Setting certain user options to some special value (e.g. some Y/N flag to skip some validation in the staging job).
- A scratch request is commonly considered as a **dangerous** ChangeMan ZMF feature, because of the known ChangeMan ZMF issues related to using scratch request, e.g.: ChangeMan ZMF audit does not include any validations related to scratch requests.
- A selective unfreeze (followed by edit-in-stage and re-freeze) of components in a successfully audited and frozen package, which may introduce inconsistencies for which no built-in controls exist in ChangeMan ZMF to prevent them.
- ...

Just a complete disable of a function like 'unfreeze' (or utility request 'rename') might be an option in certain cases, which is typically done by just remove it from the ISPF panels. But how can you implement something to restrict access to such functions (without removing them entirely)?

What is needed is a technique to restrict the execution of certain ChangeMan ZMF panel commands to only those users who are authorized for it. Such technique should facilitate some type of (custom) verification related to a (custom) security entity check, which can be added as a minor (1 line?) ISPF panel validation (VER-statement).

The security entity to be checked can be either of those:

- any of the delivered CMNx entities (like CMNxGBAD, CMNxREVRT, etc).
- a brand new entity (e.g. specific to some special CMN/ZMF function to be secured, e.g. CMNxRENAM, CMNxUNFRZ, ...).
- any existing entity (like for promote, approve, ..., etc). Wouldn't it be great if you could make this even more dynamic by indicating "go look up the security entity corresponding to the highest promotion level that this package has been promoted to"?

With such technique in place, it should be possible to implementing custom validations as in these situations:

- Access to the package revert function is restricted to only those people who are authorized to promote to the highest promotion level, which is exactly what is asked for in the question (on

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

Serena.com) about Restricting the revert function at certain promotional levels (though the answer there about CMNxREVR / CMNREVRT is incomplete: that security entity is NOT checked if the approval process has not been started).
- Setting certain user options (on the user options panel) to something different from the default values is restricted to an existing or special entity.
- Unfreeze is only allowed by the highest approval level entity.
- Limit access to the scratch function to (e.g.) only CMN/ZMF administrators (to prevent errors triggered by unexperienced users because of various issues related to using scratch requests, e.g CMN/ZMF's audit function doesn't consider them).
- ...

If you want to implement such technique, consider the approach to address this issue as documented in the Z-Clues (login required).

Source URL (retrieved on *2026-11-15 01:35*):
http://dr.chgman.com/z-factory/z-issues/scmbeans/s003

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