

BaphCorp



SAFCHECK

Guide Utilisateur

Version 3.3

Janvier 2023

Alain Barthélemy

Table des matières

NOUVEAUTES & EVOLUTIONS	4
<i>Version 1.1</i>	<i>4</i>
<i>Version 1.2</i>	<i>4</i>
<i>Version 2.7</i>	<i>4</i>
<i>Version 3.3</i>	<i>4</i>
FONCTIONNEMENT DE SAFCHECK	5
UTILISATION DE SAFCHECK	5
<i>Contraintes de mise en œuvre.....</i>	<i>5</i>
<i>Appel de SAFCHECK sous TSO.....</i>	<i>5</i>
<i>Appel de SAFCHECK en Batch.....</i>	<i>6</i>
SYNTAXE DE SAFCHECK	7
<i>Codes retour.....</i>	<i>7</i>

Figures

<i>Figure 1 : Définition dans IKJTSOxx</i>	<i>5</i>
<i>Figure 2 : SAFCHECK sous TSO.....</i>	<i>5</i>
<i>Figure 3 : SAFCHECK en Batch.....</i>	<i>6</i>

Nouveautés & Evolutions

Les apports des différentes versions sont les suivants :

Version 1.1

- Version d'origine.

Version 1.2

- Ajout de l'accès par défaut.

Version 2.7

- Ajout de la possibilité de valider les droits d'un autre utilisateur.

Version 3.3

- Ajout de la possibilité de valider la classe DATASET.
- Ajout de la possibilité de placer la ressource entre cotes.

Fonctionnement de SAFCHECK

SAFCHECK est un programme assembleur permettant de vérifier si un utilisateur dispose d'un droit précis sur une ressource générale RACF ou sur un Dataset. Le résultat se matérialise par un simple code retour. Il est très utile pour sécuriser un développement Rexx sous TSO et peut servir dans une chaine batch pour activer ou désactiver un traitement.

Utilisation de SAFCHECK

SAFCHECK est prévu pour être utilisé soit sous TSO où il utilise le Command Processor⁽¹⁾, soit par batch avec EXEC PGM=SAFCHECK. Attention, un TSO batch est considéré comme un TSO dès lors qu'il appelle un programme IKJEFTxx.

SAFCHECK nécessite au moins deux paramètres : la classe de ressource et la ressource désirée. Si l'on désire tester un accès supérieur à *Read*, il faut l'indiquer en troisième paramètre. Si l'on veut tester les droits d'un autre utilisateur il faut ajouter le userid à tester après l'accès demandé. Les deux derniers paramètres prennent la valeur par défaut s'ils sont codifiés avec *.

A noter qu'il est possible de définir la ressource entre cotes, mais cela n'a aucun intérêt.

Contraintes de mise en œuvre

- Dans son fonctionnement, SAFCHECK nécessite d'être placé dans une bibliothèque APF.
- Comme SAFCHECK utilise le Command Processor TSO il doit être défini en commande autorisée dans le membre IKJTSoxx

```
AUTHCMD NAMES (          /* AUTHORIZED COMMANDS          */ +
. . . . .                +
  SAFCHECK                +
. . . . .                +
```

Figure 1 : Définition dans IKJTSoxx

Appel de SAFCHECK sous TSO

Ci-dessous, un exemple d'utilisation de SAFCHECK. A noter que la neutralisation de WTPMSG permet d'éviter l'affichage d'un éventuel message d'erreur de l'ESM à l'écran.

```
/* Rexx : Vérification d'un droit */
Say "*** Classe Ressource Accès User : "
Parse pull Classe Rsrc Acces User .
Address TSO
'SAFCHECK' Classe Rsrc Acces User
Select
  When Rc = 0 then say "Ressource autorisée"
  When Rc = 4 then say "Ressource inconnue"
  When Rc = 8 then say "Ressource interdite"
  Otherwise say "Erreur de paramètres"
End
Return 0
```

Figure 2 : SAFCHECK sous TSO

¹ Le programme SAFCHECK doit être défini dans le paragraphe AUTHCMD de l'IKJTSoxx

Appel de SAFCHECK en Batch

Vous avez ci-dessous, un exemple d'utilisation de SAFCHECK en Batch suivi d'un appel en TSO batch. Le deuxième step utilise le command processor comme un appel en TSO interactif.

```
//SAFCHKBT JOB ('BaphCorp'),'Test Pgm',MSGCLASS=H,CLASS=A,  
//          REGION=7M,MSGLEVEL=(1,1),NOTIFY=&SYSUID  
//*  
//*-----*  
//* SAFCHECK en programme *  
//*-----*  
//*  
//SAFCHKBT EXEC PGM=SAFCHECK,  
//          PARM='FACILITY BPX.SUPERUSER read TzssY01'  
//STEPLIB DD DISP=SHR,DSN=TZSSY03.USER.LOADLIB  
//SYSPRINT DD SYSOUT=*  
//SYSUDUMP DD SYSOUT=*  
//*  
//*-----*  
//* SAFCHECK en command processor *  
//*-----*  
//*  
//SAFCHKTS EXEC PGM=IKJEFT1B,  
//          PARM='%SAFCHECK FACILITY BPX.SUPERUSER read'  
//STEPLIB DD DISP=SHR,DSN=TZSSY03.USER.LOADLIB  
//SYSEXEC DD DISP=SHR,DSN=TZSSY03.USER.EXEC  
//SYSTSPRT DD SYSOUT=*  
//SYSUDUMP DD SYSOUT=*  
//SYSTSIN DD DUMMY  
//
```

Figure 3 : SAFCHECK en Batch

Syntaxe de SAFCHECK

SAFCHECK	<i>Classe</i>	<i>Ressource</i>	[<i>Accès</i>	[<i>User</i>]]
----------	---------------	------------------	----------------	-------------------

Les paramètres *Classe*, *Accès* et *User* peuvent être écrits indifféremment en majuscules ou minuscules, seul le paramètre *Ressource* est sensible à la casse.

Les paramètres *Accès* et *User* sont facultatifs.

Descriptif

Classe	Indique la classe de ressource
Ressource	Indique la ressource à tester (<u>Attention</u> à la casse)
Accès	Indique l'accès désiré. Les valeurs acceptées sont : READ, UPDATE, CONTROL ou ALTER En indiquant * on prend la valeur par défaut (READ)
User	Indique le user pour lequel le droit est testé. Indiquer * revient à utiliser le user qui appelle le programme.

Note :

- La ressource peut être définie entre cotes, si elle contient des blancs ils seront conservés et le résultat sera un *Abend 282-5C*

Codes retour

0	Ressource autorisée.
4	Ressource ou classe inconnue.
8	Ressource interdite.
12	Erreur de paramètre.