



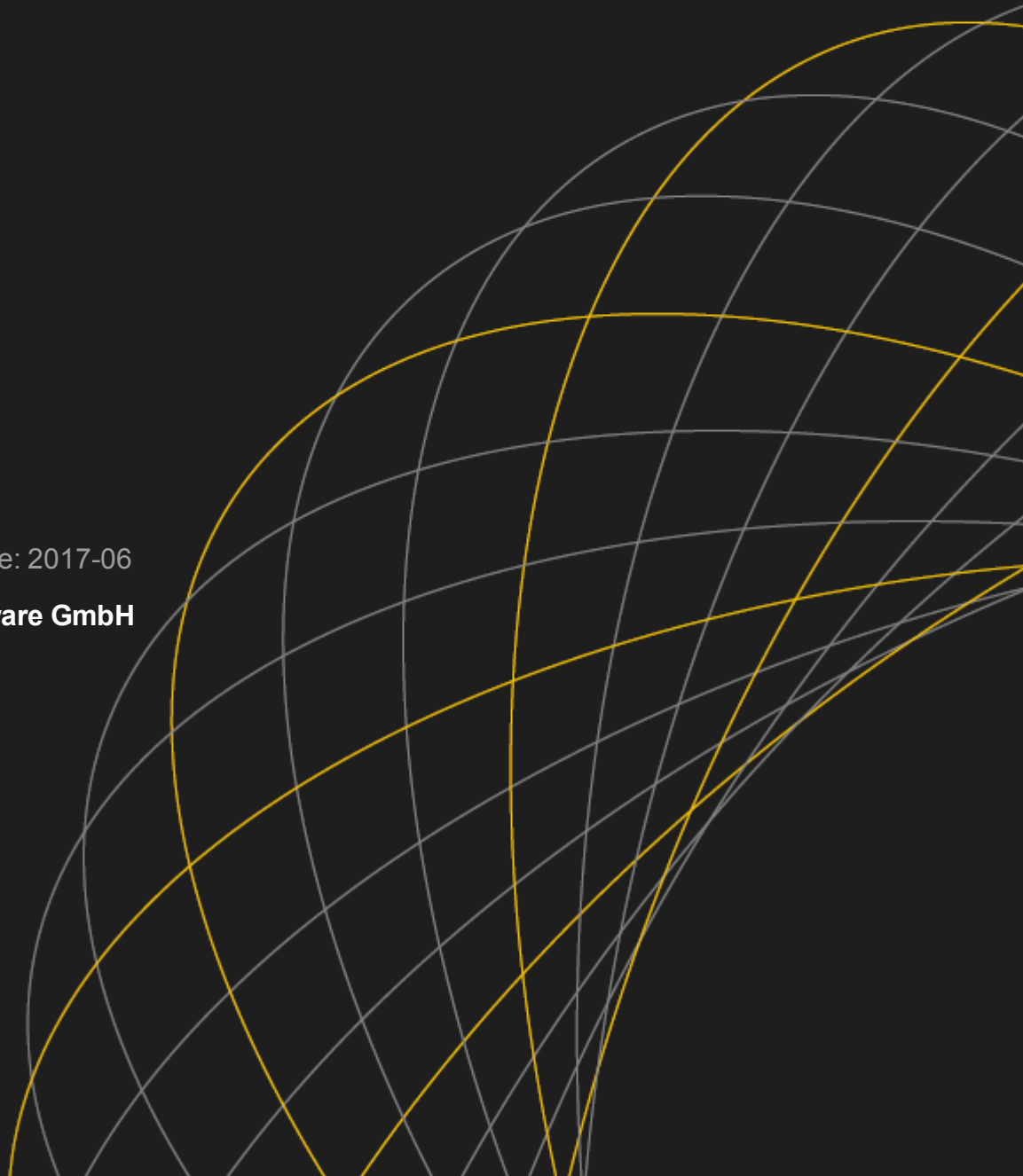
Dollar Universe

Manuel commandes

Version: 6.8.01

Publication Date: 2017-06

Automatic Software GmbH



Les logos Automic® et Automic® sont des marques déposées appartenant à Automic Software GmbH (Automic). L'utilisation de toutes les marques déposées nécessite une autorisation explicite écrite et est soumise aux conditions de la licence. Les logiciels et/ou le programme sont la propriété protégée d'Automic Software. L'accès et l'utilisation de ces derniers sont soumis à des conditions de licence devant être approuvées par écrit.

Les logiciels et/ou le programme sont également protégés par des droits de reproduction par des contrats internationaux ainsi que par les lois nationales et internationales. La copie non autorisée et d'autres formes de reproduction partielle ou totale, la décompilation, la reproduction, la modification, ou le développement de produits dérivés du logiciel sont strictement interdits. Le non-respect de ces conditions peut entraîner des poursuites pénales.

Aucune responsabilité ne sera acceptée pour toute modification, omission, erreur d'impression ou de production. Toute reproduction, totale ou partielle, est strictement interdite.

© Copyright Automic Software GmbH. Tous droits réservés.

Les autres noms peuvent être des marques de leurs propriétaires respectifs.

Table des matières

1 Release Notes	9
2 Introduction	10
2.1 Les commandes de gestion du paramétrage	10
2.2 Les commandes de paramétrage	10
2.3 Les commandes d'exploitation	11
2.4 Les commandes de gestion de batch	12
2.5 Les commandes de transfert FTP	12
2.6 Les commandes DQM	12
2.7 Les commandes de gestion de date	13
2.8 Les commandes de gestion des variables cachées	13
2.9 Les commandes du CL	13
2.10 Les variables	14
2.11 Les commandes spécifiques à OS/400	14
2.12 Les alias sous OpenVMS	14
2.13 Spécifications générales	15
2.13.1 Format d'une commande	15
2.13.2 Utilisation des commandes	17
2.13.3 Quelle syntaxe pour quel système d'exploitation	20
2.13.4 Conventions de nommage	22
3 Commandes de paramétrage	25
3.1 Extraction et insertion	25
3.1.1 Description des items	26
3.1.2 Exemples	27
3.2 Les sociétés	27
3.2.1 Description des items	28
3.3 Les nœuds	29
3.3.1 Description des items	30
3.3.2 Exemples	31
3.4 Les variables de nœuds	31
3.4.1 Description des items	32
3.4.2 Exemples	32
3.5 Les types d'unités de gestion	32

3.5.1 Description des items	33
3.5.2 Exemples	33
3.6 Les unités de gestion	34
3.6.1 Description des items	35
3.6.2 Exemples	36
3.7 Les dépendances des unités de gestion	37
3.7.1 Description des items	37
3.7.2 Exemples	38
3.8 Les variables d'Unités de Gestion	38
3.8.1 Description des items	39
3.8.2 Exemples	39
3.9 Les domaines	39
3.9.1 Description des items	40
3.9.2 Exemples	40
3.10 Les applications	41
3.10.1 Description des items	42
3.10.2 Exemples	42
3.11 Les répertoires application	42
3.11.1 Description des items	43
3.11.2 Exemples	44
3.12 Les répertoires unités de gestion	44
3.12.1 Description des items	45
3.12.2 Exemples	46
3.13 Les profils utilisateurs	46
3.13.1 Description des items	47
3.14 Les utilisateurs	48
3.14.1 Description des items	49
3.14.2 Exemples	50
3.15 Les classes	50
3.15.1 Description des items	51
3.15.2 Exemples	51
3.16 Les ressources	52
3.16.1 Description des items	54
3.16.2 Exemples	56

3.17 Les Uprocs	56
3.17.1 Description des items	68
3.17.2 Exemples	77
3.18 Les sessions	77
3.18.1 Description des items	80
3.18.2 Exemples	82
3.19 Les calendriers	83
3.19.1 Description des items	86
3.19.2 Exemples	87
3.20 Les règles de planification	88
3.20.1 Description des items	90
3.20.2 Exemples	92
3.21 Les tâches	93
3.21.1 Description des items	102
3.21.2 Exemples	108
3.22 Les déclencheurs	110
3.22.1 Description des items	113
3.22.2 Exemples	115
3.23 Les dossiers d'exploitation	115
3.23.1 Description des items	118
3.23.2 Exemples	119
4 Commandes d'exploitation	120
4.1 Les ressources	120
4.1.1 Description des items	122
4.1.2 Exemples	124
4.2 Les tâches	125
4.2.1 Description des items	126
4.2.2 Exemples	127
4.3 Les lancements prévus	127
4.3.1 Description des items	130
4.3.2 Exemples	133
4.4 Les événements	134
4.4.1 Description des items	136
4.4.2 Exemples	138

4.5 Les exécutions	138
4.5.1 Description des items	140
4.5.2 Exemples	143
4.6 Les notes	143
4.6.1 Description des items	145
4.6.2 Exemples	145
4.7 Les plages de maintenance	145
4.7.1 Description des items	147
4.7.2 Exemples	147
4.8 Les exceptions	148
4.8.1 Description des items	150
4.8.2 Exemples	151
4.9 Les automates	152
4.9.1 Description des items	153
4.9.2 Exemples	153
4.10 L'historique des exécutions	153
4.10.1 Description des items	154
4.10.2 Exemples	156
4.11 Le planning prévisionnel	156
4.11.1 Description des items	156
4.11.2 Exemple	157
4.12 Les statistiques	157
4.12.1 Description des items	159
4.12.2 Exemples	159
5 Gestion de batch	160
5.1 Déclenchement de batch uxordre	160
5.2 Surveillance de la durée d'exécution	163
5.2.1 uxsurjob	163
5.2.2 uxspvjob	164
5.2.3 uxalrjob	164
6 Commandes FTP	167
6.1 uxstr FTP GET	167
6.2 uxstr FTP PUT	169
7 Commandes DQM	171

7.1 Créer une queue batch	171
7.2 Modifier les paramètres d'une queue batch	173
7.3 Supprimer une queue batch	174
7.4 Initialiser une queue batch	175
7.5 Démarrer une queue batch	175
7.6 Arrêter une queue batch	176
7.7 Afficher la liste des queues batch	176
7.8 Afficher l'état d'une queue batch	177
7.9 Soumettre un job	178
7.10 Suspendre un job	179
7.11 Libérer un job suspendu	179
7.12 Afficher l'état d'un job	180
7.13 Arrêter un job	181
7.14 Modifier les priorités d'un job	181
8 Gestion de date	183
8.1 Formatage et décalage de dates	183
8.1.1 Description des items	184
8.1.2 Exemples	185
8.2 Formatage de date et décalage d'heure	186
8.2.1 Descriptions des items	187
8.2.2 Exemples	187
9 CL d'exécution	189
9.1 Présentation du CL	189
9.1.1 Normes d'écritures du CL	189
9.1.2 Le code retour du CL	189
9.1.3 Résumé des commandes	190
9.2 Interdire l'arrêt de l'exécution d'une Uproc	190
9.3 Ecrire des jalons	192
9.4 Afficher des informations	192
9.5 Générer des messages	193
9.6 Passer des paramètres	194
9.7 Reconduire des variables	195
9.8 Variables d'héritage	197
9.8.1 Contraintes sur la valeur de la variable	198

9.8.2 Affichage des variables d'héritage	199
9.8.3 Exemples	199
9.9 Variables cachées	200
10 Annexes	203
10.1 Exemple d'utilisation des commandes du CL	203
10.1.1 Les jalons de reprise et les messages dans un script UNIX ksh	203
10.1.2 Les steps de reprise et les messages dans un script Unix CSH.	204
10.2 Annexes AS/400	204
10.2.1 Exemples de programmes sous OS/400	204
10.2.2 Langage de programmation CLP	207
11 A propos d'Automic, de la communauté et des services	213
11.1 La société Automic Software	213
11.2 La communauté Automic	213
11.3 Centre de téléchargement	213
11.4 Support technique	214
11.5 Formation et services	214

1 Release Notes

Le tableau ci-dessous récapitule l'historique des modifications. Les informations sur les corrections de problèmes et sur les problèmes connus sont disponibles dans les Release Notes Dollar Universe ou en contactant le support technique d'Automic.

Date	Fonctionnalités clés
11-2014	Les manuels utilisateurs de Dollar Universe changent de look pour s'intégrer dans le monde Automic.
12-2014	Evolution des Uprocs de type FTP et de la commande uxstr FTP.
03-2015	Contraintes sur les valeurs des variables d'héritage Syntaxe pour les fuseaux horaires Transferts de fichiers récursifs
07-2015	Types d'Uprocs supportant les variables d'héritage
08-2015	Dupliquer une Unité de Gestion Nouvelles variables dans les ressources fichier Variable de type mot de passe
12-2015	Correction sur les filtres de la commande uxlst ctl. Comment changer une tâche multi-journalière en tâche journalière (ou vice versa).
03-2016	Condition d'enchaînement sur statut ENDED (Fini) Chemins "normal", "erreur" ou "code retour" d'une session. Corrections sur les valeurs par défaut dans les conditions Nouveau paramètre nœud pour afficher les listes des exécutions, lancements et événements comme en v5.
05-2016	L'item NODE= est décrit à la section " Environnement " (sauf commandes DQM).
09-2016	Sous OS/400, UXSETVAR est un objet CMD et non PGM. Les Uprocs de type CL_INT, CL_EXT ou CMD peuvent utiliser les shells suivants : POWERSHELL, CSCRIPT (sauf type CMD) et PERL. Description de la commande uxordre corrigée.
03-2017	Correction de la description du format de la commande uxset var Correction des items obligatoires / optionnels des commandes uxins et uxext TSK DOM n'est pas un argument des commandes uxadd et uxupd UPR
06-2017	La commande DQM uxsubjob est supportée Item REGEX pour rechercher une chaîne dans un fichier pour définir le statut avancé d'une Uproc Item IFPLANNED dans les conditions d'enchaînement pour ne vérifier une condition que si le traitement conditionnant est planifié.

2 Introduction

Bienvenue dans le Manuel Commandes de Dollar Universe !

Les commandes de Dollar Universe permettent d'agir sur l'automate depuis un environnement extérieur (applicatif client, shell UNIX, scripts...). Elles se répartissent en groupes présentés dans les chapitres suivants de ce manuel :

- Les commandes de paramétrage qui permettent de définir ou de modifier le paramétrage de Dollar Universe.
- Les commandes d'exploitation qui permettent de consulter et d'intervenir sur l'exploitation en dehors des interfaces graphiques.
- Les commandes de gestion des batchs afin de soumettre et de surveiller des exécutions.
- Les commandes FTP afin de permettre des transferts de données par FTP (sauf OS/400).
- Les commandes DQM de gestion des queues batch et des jobs dans les queues batch.
- Les commandes de gestion de date (sauf OS/400).
- Les commandes du CL pour agir sur l'automate depuis le script d'une Uproc.

Ces commandes sont unifiées pour toutes les plate formes. L'utilisation de ces commandes est régie par des règles qui sont exposées dans la suite de ce chapitre.

2.1 Les commandes de gestion du paramétrage

Ces commandes permettent :

- D'extraire les objets du paramétrage d'une Société v5 ou v6 (y compris les événements jobs) afin d'obtenir un fichier au format XML.
- De vérifier le format du fichier XML.
- D'insérer ce paramétrage dans une Société v6.

 Ces commandes sont livrées dans le kit d'upgrade de Dollar Universe v6 et sont documentées dans le Manuel d'Upgrade de Dollar Universe.

2.2 Les commandes de paramétrage

Par thème, nous décrivons, ci-dessous, sous la forme d'un tableau, la liste des commandes de paramétrage disponibles (correspondant aux options de consultation et de mise à jour des fonctions interactives) :

Commande : ux...	add	upd	dup	dlt	shw	lst	tra	dst	ext	ins	rls	scv
Thème :												
Administration									X	X		
Société	X	X		X	X	X		X				
Nœud					X	X		X				
Var. Nœud	X	X		X		X						
Type UG	X	X		X	X	X						
UG	X	X	X	X	X	X		X				

Elles permettent d'intervenir sur l'exploitation en cours. Elles sont décrites au paragraphe [Commandes d'exploitation](#).

2.4 Les commandes de gestion de batch

Quatre commandes permettent de déclencher l'exécution d'un batch et d'en surveiller la durée d'exécution :

- [Déclenchement de batch uxordre](#) : déclenche une tâche provoquée ou planifiée.
- [uxsurjob](#) : surveillance de la durée d'exécution d'une Uproc et interruption de l'exécution au-delà d'un temps limite.
- [uxspvjob](#) : surveillance de la durée d'exécution d'une Uproc et exécution d'un script au delà d'un temps limite. Précis à la minute.
- [uxalrjob](#) : surveillance de la durée d'exécution d'une Uproc et exécution d'un script au delà d'un temps limite. Précis à la seconde.

2.5 Les commandes de transfert FTP

Ces commandes utilisent les mécanismes de Dollar Universe Manager for File Transfer, intégré à Dollar Universe v6 (sauf OS/400).

- [uxstr FTP GET](#) : pour récupérer un fichier par FTP
- [uxstr FTP PUT](#) : pour envoyer un fichier par FTP

2.6 Les commandes DQM

DQM propose des commandes de gestion des queues batch :

- [uxaddque](#) : [Créer une queue batch](#)
- [uxupdque](#) : [Modifier les paramètres d'une queue batch](#)
- [uxdlque](#) : [Supprimer une queue batch](#)
- [uxresetque](#) : [Initialiser une queue batch](#)
- [uxlstque](#) : [Afficher la liste des queues batch](#)
- [uxstrque](#) : [Démarrer une queue batch](#)
- [uxstpque](#) : [Arrêter une queue batch](#)
- [uxshwque](#) : [Afficher l'état d'une queue batch](#)

Et des commandes de gestion des jobs dans les queues batch :

- [uxsubjob](#) : [Soumettre un job](#) (UNIX / Linux uniquement)
- [uxrlsjob](#) : [Libérer un job suspendu](#)
- [uxhldjob](#) : [Suspendre un job](#)
- [uxshwjob](#) : [Afficher l'état d'un job](#)
- [uxstpjob](#) : [Arrêter un job](#)
- [uxsetpri](#) : [Modifier les priorités d'un job](#)

Sous OpenVMS, les jobs sont lancés dans une queue batch DQM et sont ensuite soumis à VMS Queue Manager lorsque leur statut passe à "Exécution en cours". Les commandes DQM n'impactent que les queues batch DQM et les commandes de VMS Queue Manager n'impactent que les queues batch VMS. Reportez-vous à la documentation OpenVMS pour plus d'informations sur la gestion des queues batch OpenVMS.

 Sous OpenVMS, l'utilisateur doit avoir le privilège OPER pour exécuter les commandes de VMS Queue Manager.

2.7 Les commandes de gestion de date

Sous Windows/UNIX/OpenVMS, une commande permet de formater ou effectuer des décalages sur des dates :

- **uxdat** permet de modifier le format d'une date, de réaliser un décalage et d'obtenir le type ou le rang d'un jour dans le calendrier.
- **uxtim** permet de convertir une date (connue ou non) dans un format donné et/ou d'y ajouter un décalage horaire.

Reportez-vous au paragraphe "[Gestion de date](#)".

2.8 Les commandes de gestion des variables cachées

Ces commandes permettent de définir des variables dont la valeur est cachée à l'utilisateur (interface, logs...).

- **uxhideval ... insert...** : crée une variable avec sa valeur dans le fichier des variables.
- **uxhideval ... list...** : liste les variables du fichier ainsi que leur valeur.
- **uxhideval ... delete...** : supprime une variable et sa valeur du fichier.
- **uxhideval ... extract...** : affiche en clair la valeur de la variable dans la sortie standard.
- **uxexechidden** : (alias EXECHIDDEN sous OS/400) exécute une commande passée en paramètre en utilisant la valeur cachée d'une ou de plusieurs variables.

Reportez-vous au paragraphe "[Variables cachées](#)".

2.9 Les commandes du CL

Les commandes uxset permettent d'agir sur l'automate depuis les scripts des Uprocs :

- **uxset msg** : (alias UXSETMSG sous OS/400) envoi de messages dans la trace automate de l'exécution à des fins de diagnostic : "[Générer des messages](#)".
- **uxset step** : (alias UXSETSTEP sous OS/400) pose de jalons pour suivre l'exécution d'un script : "[Ecrire des jalons](#)".
- **uxset parm** : (alias UXSETPARM sous OS/400) passage de paramètres, au sein d'une session, d'une Uproc à sa suivante : "[Passer des paramètres](#)".
- **uxset var** : (alias UXSETVAR sous OS/400) reconduction de la valeur des variables d'une Uproc au sein d'une session : "[Reconduire des variables](#)".
- **uxset info** : (alias UXSETINFO sous OS/400) affichage d'informations supplémentaires dans le suivi des exécutions : "[Afficher des informations](#)".
- **uxsetabort** : interdit l'arrêt de l'exécution du job par une interface d'exploitation de Dollar Universe : "[Interdire l'arrêt de l'exécution d'une Uproc](#)".
- **uxset hvar** : assigne des valeurs aux variables et spécifie quelles variables seront transmises à l'Uproc suivante : "[Variables d'héritage](#)".
- **uxunset hvar** : arrête la transmission à l'Uproc suivante des variables spécifiées : "[Variables d'héritage](#)".

- **uxset inheritance** : définit si les variables seront transmises à l'Uproc suivante : "[Variables d'héritage](#)".

2.10 Les variables

Un ensemble de variables contextuelles à l'exécution des Uprocs et à la configuration de Dollar Universe est disponible dans le CL, il peut être utilisé pour tester, par exemple, un compte de soumission, le nœud ou l'unité de gestion d'exécution, etc. Reportez-vous au Manuel de Référence de Dollar Universe, paragraphe "Paramètres d'exécution".

Deux méthodes peuvent être utilisées pour générer des variables d'environnement :

- Variables de nœud, d'UG ou d'Uproc : ce sont des variables d'environnement ajoutées à l'aide d'UVC et composées d'un nom et d'une valeur. Elles peuvent être définies au niveau du nœud, de l'Unité de Gestion ou de l'Uproc. Des variables de nœud sont accessibles dans les Paramètres du nœud Dollar Universe > Variables de nœud. Les variables d'UG et d'Uproc sont définies directement dans l'objet.
- Le fichier unienv_specific : les variables d'environnement utilisées pour accéder aux environnements des applications externes doivent être ajoutées au fichier unienv_specific (.bat sous Windows, .ksh sous UNIX, .com sous OpenVMS) dans data/nodefiles ou, sous OS/400, dans le fichier UNIENVSPEC de la librairie Société. Dans ce cas, la variable n'a pas besoin d'être déclarée au niveau du nœud.

 Sous UNIX/Linux et OpenVMS, le nom des variables ne doit pas commencer par un chiffre (restriction du système d'exploitation).

2.11 Les commandes spécifiques à OS/400

Les commandes ci-dessous peuvent être utilisées sur la ligne de commande :

- UXADDFLA, UXUPDFLA et UXDLTFLA pour respectivement créer, modifier ou supprimer un lancement, reportez-vous section "[Les lancements prévus](#)".
- UXRLSRES pour libérer une ressource, reportez-vous section "[Les ressources](#)".
- UXORDRE pour déclencher une tâche existante, reportez-vous section "[Déclenchement de batch uxordre](#)".

Saisissez le nom de la commande + F4 pour afficher les paramètres et saisir leurs valeurs.

2.12 Les alias sous OpenVMS

Les alias suivants ont été créés pour compatibilité :

- UXCD : Se déplacer dans un répertoire
- UXORD : Soumettre des travaux batch.

Sous OpenVMS, les alias suivants sont disponibles :

- CDDAP : Se déplacer dans le répertoire des données de l'Espace Application
 - CDDEX : Se déplacer dans le répertoire des données de l'Espace Exploitation
 - CDDIN : Se déplacer dans le répertoire des données de l'Espace Intégration
 - CDDSI : Se déplacer dans le répertoire des données de l'Espace Simulation
 - CDEXE : Se déplacer dans le répertoire bin
-

- CDLAP : Se déplacer dans le répertoire des journaux de l'Espace Application
- CDLEX : Se déplacer dans le répertoire des journaux de l'Espace Exploitation
- CDLIN : Se déplacer dans le répertoire des journaux de l'Espace Intégration
- CDLOG : Se déplacer dans le répertoire des journaux de la Société
- CDLSI : Se déplacer dans le répertoire des journaux de l'Espace Simulation
- CDMGR : Se déplacer dans le répertoire mgr
- CDPAP : Se déplacer dans le répertoire des scripts d'Uprocs de l'Espace Application
- CDPEX : Se déplacer dans le répertoire des scripts d'Uprocs de l'Espace Exploitation
- CDPIN : Se déplacer dans le répertoire des scripts d'Uprocs de l'Espace Intégration
- CDPSI : Se déplacer dans le répertoire des scripts d'Uprocs de l'Espace Simulation
- CDROOT : Se déplacer dans le répertoire racine de la Société

2.13 Spécifications générales

Pour que la commande puisse être exécutée, il faut que les serveurs d'I/O de la société soient démarrés.

 L'utilisateur qui exécute la commande doit être déclaré dans la liste des règles utilisateurs d'UniViewer Management Serveur (dans UniViewer Console, mode Administration > Sécurité > Règles utilisateurs) et disposer des droits nécessaires à l'exécution de la commande.

2.13.1 Format d'une commande

Une commande se découpe en cinq parties principales :

Sous Windows/UNIX/OpenVMS : <Commande> <thème> <environnement> <items> <sortie> :

- <Commande> : Désignation du verbe de la commande utilisée, par exemple uxadd.
- <thème> : Définition du thème de travail pour la fonction activée, par exemple UPR.
- <environnement> : Sélection de l'environnement de travail : nœud, espace et éventuellement unité de gestion.
- <items> : Sélection des enregistrements de travail, avec positionnement des informations souhaitées. Sous OS/400, tous les items doivent être cités entre apostrophes.
- <sortie> : Définition des fichiers contenant des informations en "entrée" de la fonction (input=fichier) ou de recueil des informations en "sortie" (output=fichier).

Sous OS/400, le programme associé à la commande peut être appelé avec la même syntaxe :

```
CALL PGM(<PROGRAM>) PARM(<thème> <environnement> <items> <sorties>)
```

Commande

Le verbe de la commande doit toujours être cité en premier. Si le répertoire de travail courant n'est pas le répertoire UNI_DIR_EXEC, celui-ci doit être précisé dans la commande (sauf sur OpenVMS).

Par exemple sous UNIX :

```
${UNI_DIR_EXEC}/ux1st UPR
```

Pour obtenir la liste des Uprocs de l'espace par défaut.

Sous OS/400, la librairie Société doit être chargée.

Thème

Le thème doit toujours être cité en second à la suite du verbe de la commande, les items étant décrits par la suite dans un ordre quelconque.

```
${UNI_DIR_EXEC}/uxlst UPR
```

Dans l'exemple ci-dessus le thème est UPR. Il indique que la commande porte sur les Uprocs.

Environnement

Le fichier unienv (dans le répertoire principal de la société) doit être exécuté préalablement pour charger l'environnement par défaut. Les informations ci-dessous permettent de modifier ces valeurs :

NODE= Nom de nœud Dollar Universe.

Si la commande cible un nom de nœud distant, celui-ci doit être déclaré sur le même UVMS que le nœud Dollar Universe local ou, dans le cas d'une architecture "UVMS maître/UVMS subordonnés", sur un UVMS subordonné (du même maître) partageant ses nœuds.

Toutes les commandes ne peuvent concerner que des nœuds d'une même Société.

- Sous Windows/UNIX/OpenVMS, si le nœud n'est pas spécifié, la valeur de la variable S_NODENAME (jusqu'à 64 caractères) sera utilisée par défaut.
- Sous OS/400 en mode affichage, si le nœud n'est pas spécifié, le nœud courant (*CURRENT) est utilisé.

EXP pour l'espace d'exploitation, SIM pour simulation, INT pour intégration et APP pour application.

- Sous Windows/UNIX/OpenVMS, si l'espace n'est pas indiqué c'est la valeur de la variable S_ESPEXE qui sera pris en compte.
- Sous OS/400 en mode affichage, si l'Espace n'est pas spécifié, l'Espace courant (*CURRENT) est utilisé.

Items

Pour chaque commande, une liste d'items est détaillée :

- Les items obligatoires sont présentés en gras,
- "*" (astérisque) souligne la possibilité de saisir une valeur générique pour l'item concerné.

Sous OS/400, les items utilisant le signe = doivent être saisis entre apostrophes. Par exemple : 'MU=MYMU'.

Fichiers d'entrée / sortie

Les items input et output sont principalement utilisés avec les commandes d'extraction et d'insertion uxext et uxins proposées par Dollar Universe afin de permettre une interface facilitée avec un produit de télédistribution par exemple.

L'item output peut également être utilisé par les commandes uxshw ou uxlst pour recueillir les informations dans un fichier à la place de l'affichage à l'écran (valeur par défaut).

Un même fichier de sortie peut être utilisé pour un ensemble de commandes uxext (pouvant porter sur des objets différents), les données extraites étant ajoutées en fin de fichier à celles déjà extraites par les commandes précédentes.

Il est cependant recommandé de ne pas utiliser le même fichier de sortie pour des objets appartenant à des espaces différents, dans ce dernier cas l'utilisation de caractères génériques sur les version d'objets (Uprocs, session et tâches), pourrait générer des confusions. Par exemple, si le même fichier contient le résultat des deux commandes suivantes :

```
uxext UPR APP UPR="*" VUPR="*" output=fichier
uxext UPR INT UPR="*" VUPR="*" output=fichier
```

Si le même couple (Uproc, version) existe dans les deux espaces, et comme l'item REPL n'est pas indiqué seule l'Uproc version de l'espace application sera inséré dans le fichier.

Si l'item REPL était indiqué dans la deuxième commande, seule l'Uproc version de l'espace intégration persistera dans le fichier (en écrasant la définition donnée en espace application).

Code retour

Les codes retournés par les commandes sont les suivants :

- 0 (ou nombre impair sous OpenVMS) : la commande a été exécutée correctement,
- 1 (ou nombre pair sous OpenVMS): la commande n'a pas pu être exécutée correctement, un message explique la raison de l'échec : erreur de syntaxe, serveur d'I/O absent ...
- 255 ou -1 : la commande a été exécutée correctement mais n'avait rien à faire (par exemple: pas d'événements à supprimer).

2.13.2 Utilisation des commandes

Les commandes de paramétrage, citées dans le tableau ci-dessus peuvent être utilisées sous la ligne de commande du système d'exploitation (sous Windows/UNIX/OpenVMS) ou dans une procédure.

Les commandes d'exploitation citées dans le second tableau ainsi que les commandes de gestion de batch peuvent être utilisées sous la ligne de commande du système d'exploitation (sous Windows/UNIX/OpenVMS) ou dans une procédure.

Les commandes du CL ne peuvent être utilisées que dans un CL associé à une Uproc. Elles ne peuvent s'exécuter que dans le contexte fourni par Dollar Universe.

Les particularités liées aux différents modes d'utilisation sont décrites ci-après.

Paramétrage des commandes par UniViewer Console

Les paramètres nœud d'un nœud Dollar Universe v6, catégorie "Réglages de l'interface commande" permettent de définir le nombre de caractères affichés pour les différents noms d'objets.

De ce fait, les noms des objets affichés par une commande peuvent être tronqués en fonction de la valeur de ces paramètres. Les valeurs par défaut sont les suivantes :

Paramètre	Objet	Longueur
Longueur d'affichage du nom de nœud	Nœud	64 c
Longueur d'affichage du nom d'Unité de gestion	UG	64 c
Longueur d'affichage du nom d'utilisateur	Compte utilisateur	64 c
Longueur d'affichage du nom de ressource	Ressource	64 c
Longueur d'affichage du nom d'uproc	Uproc	64 c
Longueur d'affichage du nom de classe	Classe d'incompatibilité	64 c
Longueur d'affichage du nom d'application	Application	64 c
Longueur d'affichage du nom de règle	Règle	64 c
Longueur d'affichage du nom de Session	Session	64 c
Longueur d'affichage du nom de tâche	Tâche	64 c
Longueur d'affichage du nom de dossier d'exploitation	Dossier d'exploitation	64 c
Longueur d'affichage des descriptions	Descriptions	40 c
Longueur d'affichage du nom des variables	Nom d'une variable	64 c

Paramètre	Objet	Longueur
Longueur d'affichage des valeurs des variables	Valeur d'une variable	64 c
Version d'Uproc cible lors de l'insertion en APP/INT	Uproc	001
Version de session cible lors de l'insertion en APP/INT	Session	001

L'affichage des listes des exécutions, des lancements et des événements diffère en v5 et en v6 : en v6 la première colonne est le nom de la tâche associée (qui n'existe pas en v5). Pour conserver le même comportement en v5 et en v6, le paramètre nœud `U_V5_CLI_COMPATIBILITY` doit être défini à Oui.

Reportez-vous au Manuel Utilisateur d'UniViewer pour le nom et la description des paramètres nœud associés.

Règles générales

Pour plus de lisibilité des commandes le caractère "\" (sous Windows/UNIX/) ou le caractère "-" (sous OpenVMS) ou le caractère "+" (sous OS/400) peut être utilisé pour indiquer le passage à la ligne suivante.

Tous les mots clés peuvent être saisis en minuscules ou en majuscules indifféremment, seule l'utilisation différenciée de majuscules et de minuscules doit être respectée dans les commandes et les zones de saisie.

Les objets de Dollar Universe doivent tous être saisis en majuscules sauf pour les nœuds et les utilisateurs où la casse doit être respectée.

Spécificités des commandes DQM

Lors de modifications du script .profile de l'utilisateur root, exécuter les étapes suivantes (DQM travaille principalement en mémoire).

- Arrêter DQM.
- Se déconnecter de la session root.
- Se reconnecter avec l'utilisateur root.
- Relancer DQM.

Format des dates

Toutes les dates saisies ou affichées répondent, depuis la version 4.3 de Dollar Universe à la règle ci-dessous.

Le paramètre nœud `U_FMT_DATE` (catégorie Réglages généraux) permet de définir le format souhaité des dates. Sa valeur doit être construite à partir des composants suivants :

- YYYY : année sur 4 caractères,
- MM : numéro de mois de 01 à 12 (sur deux caractères),
- DD : jour dans le mois de 01 à 31 (sur deux caractères),
- Un séparateur qui peut être choisi parmi les valeurs caractères suivants : / (slash) ou : (deux points) ou - (tiret) ou pas de séparateur. Le séparateur, s'il est utilisé, ne peut servir qu'à séparer les trois composantes de la date et non à les encadrer.

Par exemple si on souhaite exprimer la date du 21 juin 2008 :

- Si U_FMT_DATE vaut DDMMYYYY, la date affichée ou saisie sera le 21062008,
- Si U_FMT_DATE vaut YYYY-DD-MM, la date affichée ou saisie sera le 2008-21-06.

Si le paramètre U_FMT_DATE n'est pas valorisé alors sa valeur par défaut est DD/MM/YYYY en Français, MM/DD/YYYY en Anglais.

 Les commandes OS/400 UX*FLA et UXORDRE utilisent le format de date du système et non le format indiqué par le paramètre U_FMT_DATE.

Aide en ligne

Une aide en ligne est accessible pour toutes les commandes de Dollar Universe (sauf uxset, uxspvjob, uxsurjob et uxalrjob).

 Pour les commandes des variables d'héritage, l'aide en ligne est accessible en tapant la commande : uxset hvar, uxunset hvar ou uxset inheritance.

L'aide en ligne permet d'obtenir la syntaxe de la commande, la liste des items, leur format, éventuellement les valeurs possibles et leur valeur par défaut.

Pour obtenir l'aide en ligne sur une commande, taper la commande suivie de "help".

Par exemple, pour obtenir l'aide en ligne sur la commande uxlst upr :

```
C:\Automic\DUAS\UNIV62_FRWPMDEV08\bin>uxlst upr help
Lists the uprocs.

uxlst UPR
[Node]=          format:64c      default:S_NOEUD or S_NODENAME
[EXP | SIM | INT | APP] default:S_ESPEXE
[UPR]=           format:64c *    default:*
[VUPR]=          format:3n *    default:*
[DOM]=           format:1c *    default:*
[APPL]=          format:64c *    default:*
[INF | FULL]
[OUTPUT]=file_name
```

Les items présentés en crochets : [] sont optionnels, les autres obligatoires.

Deux items séparés par le caractère : | sont incompatibles.

Format : indique le format de saisie de l'item. Le nombre de caractères est suivi de :

- "n" pour une valeur numérique.
- "c" pour chaîne de caractères.
- "*" indique que ce caractère générique peut être saisi en fin de chaîne ou seul.

Default : indique la valeur par défaut (ce peut être la valeur d'une variable, S_ESPEXE pour l'espace par exemple).

U_FMT_DATE : indique que la date doit être saisie au format indiqué par la valeur de ce paramètre nœud.

 Sous OS/400, les paramètres d'une commande spécifique OS/400 peuvent être affichés par la touche F4. L'utilisateur peut alors saisir la valeur des paramètres. Le nom des paramètres est affiché par la touche F11.

2.13.3 Quelle syntaxe pour quel système d'exploitation

Chaque système d'exploitation utilise une liste de caractères spécifiques qui peuvent être utilisés dans une commande de Dollar Universe.

A cette liste, peut être ajoutée la liste des caractères qui ont un sens particulier pour les commandes de Dollar Universe. Ces caractères sont : espace = () { }.

Ces caractères doivent être protégés afin d'être transmis aux commandes de Dollar Universe sans avoir été interprétés par le système d'exploitation. Le type de protection dépend du système d'exploitation.

Utiliser les commandes sous Windows

Espace + - () sont des caractères spécifiques.

Un seul type de protection peut être utilisé : le guillemet, par exemple : "texte ou champ texte".

L'accent circonflexe ^ peut également être utilisé pour protéger des caractères spéciaux individuels, y compris les guillemets. Par exemple : texte^+texte^"texte

Exemples

Correct :

```
uxadd MU MU=CENTER LABEL="Siège société" TNODE=%S_NODENAME%
uxupd MU MU=CENTER LABEL=" Siège société (M)"
uxupd UPR EXP UPR=D_LOAD_FIL TERMS add TERMNO=02 TERM=D_LOAD_FIL
STATUS=ABNORMAL PDATE="(-1,,)"
uxupd UPR EXP UPR=D_LOAD_FIL FORMULA LFORM="=C01 AND =C02"
uxadd MU MU=CENTER LABEL=Company^+center TNODE=%S_NODENAME%
uxupd UPR EXP UPR=D_LOAD_FIL TERMS add TERMNO=02 TERM=D_LOAD_FIL
STATUS=ABNORMAL PDATE=^(^1,,)
```

Incorrect :

```
uxadd MU MU=CENTER LABEL= Siège société TNODE=%S_NODENAME%
uxadd MU MU=CENTER LABEL=' " Siège société "' TNODE=%S_NODENAME%
```

Utiliser les commandes sous UNIX

Espace + - () { } [] sont des caractères spécifiques.

Deux types de protection peuvent être utilisés :

- Le guillemet (si l'espace est le seul caractère spécifique), par exemple : "texte ou champ texte". Les guillemets " protègent tous les caractères spéciaux sauf " \$ ' et \ (guillemets, dollar, apostrophe et antislash)
- L'apostrophe, par exemple : 'texte (ou champ texte)'. L'apostrophe ' protège tous les caractères spéciaux sauf l'apostrophe.

L'antislash \ peut également être utilisé pour protéger un caractère spécifique. Par exemple : texte\\$

Exemples

Correct :

```
uxadd MU MU=CENTER LABEL=" Siège société " TNODE=$S_NODENAME
uxupd MU MU=CENTER LABEL=" Siège société \(M\)"
uxadd MU MU=CENTER LABEL=' Siège société (M)' TNODE=$S_NODENAME
```

```
uxupd UPR EXP UPR=D_LOAD_FIL TERMINs add TERMNO=01 TERM=D_LOAD_FIL
STATUS=ABNORMAL PDATE='(-1,,)'
uxupd UPR EXP UPR=D_LOAD_FIL FORMULA LFORM="=C01 AND =C02"
uxupd UPR EXP UPR=D_LOAD_FIL FORMULA LFORM="=C01 AND =C02"
uxupd UPR EXP UPR=D_LOAD_FIL FORMULA LFORM=C01\ AND\ =C02
```

Incorrect :

```
uxadd MU MU=HEADQUART LABEL=' ' Siège société ' ' TNODE=$S_NODENAME
uxadd MU MU=HEADQUART LABEL= Siège société TNODE=$S_NODENAME
uxadd MU MU=CENTER LABEL=\ ' ' Siège société (M)\ ' ' TNODE=$S_NODENAME
uxupd UPR EXP UPR=D_LOAD_FIL TERMINs add TERMNO=01 TERM=D_LOAD_FIL
STATUS=ABNORMAL PDATE=\ ' '(-1,,)\ ' '
uxupd UPR EXP UPR=D_LOAD_FIL FORMULA LFORM=' '=C01 AND =C02''
uxupd UPR EXP UPR=D_LOAD_FIL FORMULA LFORM=\ =C01 AND =C02
```

⚠ Certains interpréteurs de commandes peuvent avoir d'autres particularités. Ces exemples ont été testés avec le ksh et le csh.

Utiliser les commandes sous OS/400

Les apostrophes doivent être utilisées pour définir un paramètre qui utilise le signe = ainsi que pour préserver la casse.

Par exemple :

```
CALL PGM(UXSHW) PARM(SSES EXP 'SES=D_LOAD_BCK' LNK)
```

Utiliser les commandes sous Open VMS

Caractères spéciaux

+ - () et espace sont des caractères spécifiques.

Un seul type de protection peut être utilisé : le guillemet, par exemple : "texte (ou champ texte)".

L'accent circonflexe ^ peut également être utilisé pour protéger des caractères spéciaux individuels, y compris les guillemets. Par exemple : texte^+texte^"texte"

Exemples :

Correct :

```
uxadd MU MU=CENTER LABEL=" Siège société (M)" TNODE=TEST
uxupd UPR EXP UPR=D_LOAD_FIL TERMINs add TERMNO=02 TERM=D_LOAD_FIL
STATUS=ABNORMAL PDATE="(-1,,)"
uxupd UPR EXP UPR=D_LOAD_FIL FORMULA LFORM="=C01 AND =C02"
uxadd MU MU=CENTER LABEL=Company^ center TNODE=%S_NODENAME%
uxupd UPR EXP UPR=D_LOAD_FIL TERMINs add TERMNO=02 TERM=D_LOAD_FIL
STATUS=ABNORMAL PDATE=^(^ -1,,)
```

Incorrect :

```
uxadd MU MU=CENTER LABEL=" " " Siège société M " " " TNODE=TEST
uxupd UPR EXP UPR=D_LOAD_FIL FORMULA LFORM=" " =C01 AND =C02 " "
```

Respect de la casse

VMS possède deux particularités supplémentaires :

- Lorsque la commande est transmise au système d'exploitation, la casse des arguments est automatiquement modifiée en majuscules.
- Lorsqu'une commande "foreign" (cas des commandes de Dollar Universe) est transmise au système d'exploitation, la casse des arguments est parfois (cela dépend de la version du système d'exploitation) automatiquement modifiée en minuscules.

Les objets de Dollar Universe peuvent être écrits en majuscules (par défaut) ou minuscules sauf le nom de noeud Dollar Universe et le compte utilisateur/compte de soumission qui doivent respecter la casse avec laquelle ils ont été définis (la définition d'un nom de noeud doit être exactement la même sur les différents noeuds).

Pour protéger la casse, le nom de noeud doit toujours être protégé dans toutes les commandes utilisant les items NODE et TNODE, le compte utilisateur doit toujours être protégé dans toutes les commandes utilisant l'item USER.

La protection est réalisée par des guillemets, par exemple : "Noeud".

Il est également recommandé de protéger tous les arguments par des guillemets.

Exemples

```
uxupd NODE TNODE="Node" LABEL="" "Test machine""
uxdlr NODE TNODE="Node"
```

2.13.4 Conventions de nommage

Longueur du nom des objets

La longueur des champs nom et description des objets est différente selon la version de Dollar Universe. Le tableau ci-dessous récapitule ces données :

Version DUAS	< v6		> v6	
	Nom	Description	Nom	Description
Application	2	20	64	64
Business View / Chaîne de traitement	64	256	64	256
Calendrier	-	40	-	64
Classe	6	20	64	64
Compte de soumission	12	-	64	64
Déclencheur	-	-	64	-
Domaine	1	20	1	64
Dossier d'exploitation	-	-	64	64
Modèle (de tâche, de calendrier...)	10	-	64	-
Noeud	10	20	64	64
Plage de maintenance	-	-	-	64
Profil	8	40	-	-
Queue batch	31	-	31	-
Règle	8	40	64	64

Version DUAS	< v6		> v6	
Ressource	20	40	64	64
Session	10	30	64	64
Société	6	20	6	-
Tâche	-	-	64	64
Type d'UG	1	20	1	64
UG	10	20	64	64
Uproc	10	60	64	64

Certains objets n'apparaissent pas dans cette liste : ils sont spécifiques à UVMS.

Caractères autorisés

Avec Dollar Universe v6, le nom des objets peut contenir (par défaut) les caractères suivants :

A-Z 0-9 _ - . [] () espace @ # \$

- Le caractère espace, lorsqu'il est autorisé, ne peut pas être utilisé comme premier caractère.
- Les caractères minuscules sont automatiquement convertis en majuscules sauf pour les noms de nœuds, les comptes utilisateurs et les comptes de soumission.
- Unicode est également supporté.

 Seuls les caractères autorisés (A à Z, 0 à 9 et _) dans le nom de la Société sont supportés. Sous OpenVMS, le caractère "\$" est autorisé pour les noms des queues batch (i.e. SYS\$BATCH). Ces caractères spéciaux doivent être protégés (en utilisant un backslash \ ou en les encadrant de simples quotes '\$' afin d'éviter que ces caractères spéciaux soient interprétés par le langage de commandes.

Objet	Nom		
Application	Défaut		
Business View / Chaîne de traitement	Défaut		
Calendrier	Défaut		1 ^{er} caractère : A-Z 0-9
Classe	Défaut		
Compte de soumission	Défaut		
Déclencheur	Défaut		
Domaine		A-Z	
Dossier d'exploitation	Défaut		
Modèle (de tâche, de calendrier...)	Défaut		
Nœud		A-Z a-z 0-9 _ - .	Espace interdit
Queue batch		A-Z 0-9 _	
Règle	Défaut		
Ressource	Défaut		
Session	Défaut		
Société		A-Z 0-9 _	

Objet	Nom		
Tâche	Défaut		
Type d'UG		A-Z 0-9	
UG	Défaut		1er caractère : A-Z 0-9
Uproc	Défaut		

3 Commandes de paramétrage

Ce chapitre présente les commandes utilisées pour définir l'automatisation de la production.

Les informations affichées par la commande diffèrent entre Dollar Universe v5 et Dollar Universe v6. Il en est de même si un package est déployé de DUAS v5 vers DUAS v6.

3.1 Extraction et insertion

uxext ADM : extrait une table d'administration ou une partie de cette table.

Item	Format	Défaut
TSOC=	6c *	
TNODE=	64c *	
DOM=	1c *	
APPL=	64c *	
CLASSE=	64c *	
TYPUG=	1c *	
UG=	64c *	
TIH		
USER=	64c *	
REPERT		
REPL		
OUTPUT=file_name		
NOVERIF		

Exemple :

```
uxext ADM APPL=B* OUTPUT="c:\temp\adm.ext"
```

uxins ADM : insère des informations issues d'une extraction dans les tables d'administration correspondantes pour la même version de Dollar Universe.

Item	Format	Défaut
TSOC=	6c *	
TNODE=	64c *	
DOM=	1c *	
APPL=	64c *	
CLASSE=	64c *	
TYPUG=	1c *	
UG=	64c *	
TIH		

Item	Format	Défaut
USER=	64c *	
REPERT		
REPL		
INPUT=file_name		

Exemple :

```
uxins ADM APPL=* INPUT="c:\temp\adm.ext"
```

3.1.1 Description des items

TSOC : table des sociétés. Saisie d'un code société (valeur générique acceptée). Format : 6 caractères alphanumériques majuscules.

TNODE : table des nœuds. Saisie d'un code nœud (valeur générique acceptée). Format : 64 caractères alphanumériques.

DOM : table des domaines. Saisie d'un code domaine (valeur générique acceptée). Format : 1 caractère alphabétique.

APPL : table des applications et des répertoires d'application. Saisie d'un code application (valeur générique acceptée). Format : 64 caractères alphanumériques.

- table des applications si REPERT n'est pas mentionné,
- table des répertoires d'applications si REPERT est indiqué; dans ce cas, les répertoires de toutes les applications renseignés dans la table sont extraits (les répertoires de toutes les applications contenus dans le fichier d'entrée sont insérés).

CLASSE : table des classes d'incompatibilités. Format : 64 caractères alphanumériques. Saisie d'une classe d'incompatibilités (valeur générique acceptée).

TYPUG : table des types d'unité de gestion. Format : 1 caractère alphanumérique. Saisie d'un code type d'unité de gestion (valeur générique acceptée).

UG : table des unités de gestion et des répertoires d'unités de gestion. Saisie d'un code unité de gestion (valeur générique acceptée). Format : 64 caractères alphanumériques.

- table des unités de gestion si REPERT n'est pas mentionné,
- table des répertoires d'unités de gestion si REPERT est indiqué. Toutes les applications sont considérées dans l'extraction ou l'insertion.

TIH : table du réseau des unités de gestion. L'extraction se fait de manière globale (pas de saisie de valeur et donc pas de sélection possible).

USER : tables des utilisateurs. Saisie d'un compte de soumission (valeur générique acceptée). Format : 64 caractères alphanumériques.

 Les items précédents sont incompatibles (extraction d'une seule table à la fois) hormis APPL et UG utilisés avec REPERT pour indiquer la table des répertoires d'application ou d'unité de gestion. L'un au moins de ces items est requis.

REPERT : répertoires d'application ou d'unité de gestion. Se combine avec APPL et/ou UG pour déterminer la table à traiter.

REPL : option de remplacement. Indique si l'enregistrement extrait (inséré) annule et remplace l'enregistrement déjà existant dans le fichier de destination; par défaut, l'extraction (l'insertion) est refusée pour chaque enregistrement déjà existant (signalé par un message pour chaque occurrence).

OUTPUT : fichier de sortie pour la commande uxext. Le nom du fichier est obligatoire.

INPUT : fichier d'entrée pour la commande uxins. Le nom du fichier est obligatoire.

NOVERIF : peut être utilisé avec la commande uxext uniquement. Cette option permet de ne pas vérifier le fichier l'extraction avant d'y écrire de nouveaux enregistrements et d'améliorer ainsi les performances de cette commande. Cet item ne doit être utilisé que pour un nouveau fichier.

3.1.2 Exemples

```
uxext ADM APPL=BK output=c:\temp\DOC.ext
```

Extrait l'application BK de la société courante et archive les données dans le fichier c:\temp\DOC.ext.

```
uxins ADM APPL=* input=c:\temp\DOC.ext
```

Insère toutes les applications du fichier c:\temp\DOC.ext dans l'environnement courant (société et nœud).

```
uxext ADM UG=S* output=c:\temp\DOC.ext
uxext ADM UG=A* output=c:\temp\DOC.ext
```

Extrait toutes les unités de gestion de type S et A et archive les données dans le fichier c:\temp\DOC.ext.

```
uxins ADM UG=* input=c:\temp\DOC.ext
```

Insère toutes les unités de gestion du fichier c:\temp\DOC.ext dans l'environnement courant (société et nœud).

3.2 Les sociétés

Ces commandes sont présentes pour compatibilité uniquement. Elles sont sans effet.

uxadd SOC : crée une société dans la table des sociétés.

Item	Format	Défaut
TSOC=	6c	
LABEL=	20c	
MASTER=	64c	
LOCK UNLOCK		UNLOCK
DIR=	60c	

Exemple :

```
uxadd SOC TSOC=EXAMPL LABEL=\"new company\" DIR=\"c:\EXAMPL\"
```

uxupd SOC : modifie les caractéristiques d'une société.

Item	Format	Défaut
TSOC=	6c *	
LABEL=	20c	

Item	Format	Défaut
MASTER=	64c	
LOCK UNLOCK		
DIR=	60c	

Exemple :

```
uxupd SOC TSOC=EXAMPL DIR="d:\EXAMPL"
```

uxshw SOC : affiche les caractéristiques d'une société.

Item	Format	Défaut
TSOC=	6c *	
OUTPUT=file_name		

Exemple :

```
uxshw SOC TSOC=*
```

uxdlt SOC : supprime une société de la table sociétés.

Item	Format	Défaut
TSOC=	6c *	

Exemple :

```
uxdlt SOC TSOC=EXAMPL
```

uxlst SOC : liste la table des sociétés.

Item	Format	Défaut
TSOC=	6c *	*
INF FULL		FULL
OUTPUT=file_name		

Exemple :

```
uxlst SOC
```

3.2.1 Description des items

TSOC : Code société. Format : 6 caractères alphanumériques majuscules. Saisie d'un code société (valeur générique acceptée sauf pour uxadd).

LABEL : Libellé descriptif de la société. Format : 20 caractères alphanumériques.

LOCK : Verrouillage des Uprocs et des sessions lors des opérations de transfert. Défaut : UNLOCK.

MASTER : Nœud maître auquel est rattaché la société. Format : 64 caractères alphanumériques. Défaut : nœud courant.

TPU : Editeur utilisé sous OpenVMS uniquement pour le CL des Uprocs et les logs. Défaut : EDT.

DIR : Répertoire d'installation de la société. Obligatoire pour la commande uxadd. Format : 60 caractères alphanumériques.

INF : Option d'affichage. Limite l'affichage au nom des objets. Autre valeur : FULL (par défaut). Affiche toutes les informations disponibles sur la liste des objets.

OUTPUT : Stockage des résultats. Si l'item n'est pas renseigné la sortie se fait à l'écran,

3.3 Les nœuds

uxadd NODE : crée un nœud dans la table des nœuds. Cette commande est présente pour compatibilité uniquement. Elle est sans effet. Les nœuds DUAS v6 sont créés par UVMS.

Item	Format	Défaut
TNODE=	64c	
LABEL=	64c	
MONI NOMONI		NOMONI
DEV NODEV		DEV
PROD NOPROD		PROD
DIR=	60c	
XDIR=	60c	
SDIR=	60c	
IDIR=	60c	
ADIR=	60c	
DDIR=	60c	
PDIR=	60c	

Exemple :

```
uxadd NODE TNODE=Asterix LABEL=\"Test machine\" DIR=\"C:\\UNIV54\"
XDIR=\"C:\\UNIV54\\exp\\data\" SDIR=\"C:\\UNIV54\\sim\\data\"
IDIR=\"C:\\UNIV54\\int\\data\" ADIR=\"C:\\UNIV54\\app\\data\" DDIR=\"C:\\UNIV54\\exec\"
PDIR=\"C:\\UNIV54\\exec\"
```

uxupd NODE : modifie les caractéristiques d'un nœud. Cette commande est présente pour compatibilité uniquement. Elle est sans effet. Les nœuds DUAS v6 sont synchronisés par UVMS.

Item	Format	Défaut
TNODE=	64c *	
LABEL=	64c	
MONI NOMONI		
DEV NODEV		
PROD NOPROD		
DIR=	60c	
XDIR=	60c	
SDIR=	60c	
IDIR=	60c	
ADIR=	60c	

Item	Format	Défaut
DDIR=	60c	
PDIR=	60c	

Exemple :

```
uxupd NODE TNODE=Asterix LABEL=\"Test node\"
```

uxshw NODE : affiche les caractéristiques d'un nœud.

Item	Format	Défaut
TNODE=	64c *	
OUTPUT=file_name		

Exemple :

```
uxshw NODE TNODE=A*
```

uxlst NODE : liste la table des nœuds.

Item	Format	Défaut
TNODE=*	64c *	*
INF FULL		FULL
OUTPUT=file_name		

Exemple :

```
uxlst NODE OUTPUT="c:\temp\LstNode.txt"
```

3.3.1 Description des items

TNODE : Code nœud. Saisie d'un code nœud (valeur générique acceptée sauf pour uxadd). Format : 64 caractères alphanumériques.

LABEL : Libellé descriptif du nœud. Format : 64 caractères alphanumériques.

MONI : Nœud de contrôle centralisé. Défaut : NOMONI.

DEV : Site de développement. Défaut : DEV.

- DEV : le nœud peut être utilisé dans les espaces application et intégration.
- NODEV : le nœud ne peut pas être utilisé dans les espaces application et intégration.

PROD : Site de production. Défaut : PROD.

- PROD : le nœud peut être utilisé dans les espaces simulation et exploitation.
- NOPROD : le nœud ne peut pas être utilisé dans les espaces simulation et exploitation.

DIR : Répertoire d'installation de la table société. Obligatoire pour uxadd. Format : 60 caractères alphanumériques.

ADIR : Répertoire pour l'espace application. Format : 60 caractères alphanumériques.

IDIR : Répertoire pour l'espace intégration. Format : 60 caractères alphanumériques.

SDIR : Répertoire pour l'espace simulation. Format : 60 caractères alphanumériques.

XDIR : Répertoire pour l'espace exploitation. Format : 60 caractères alphanumériques.

DDIR : Répertoire des exécutables en développement. Format : 60 caractères alphanumériques.

PDIR : Répertoire des exécutables en production. Format : 60 caractères alphanumériques.

INF : Option d'affichage. Limite l'affichage au nom des objets. Autre valeur : FULL (par défaut). Affiche toutes les informations disponibles sur la liste des objets.

OUTPUT : Stockage des résultats. Si l'item n'est pas renseigné, la sortie se fait à l'écran.

3.3.2 Exemples

```
uxshw NODE TNODE=Asterix
```

Affiche la définition du nœud Asterix.

3.4 Les variables de nœuds

Les variables de nœud sont accessibles dans UniViewer Console en sélectionnant le mode Administration > Nœuds > Nœuds : paramètres nœud du nœud Dollar Universe v6, catégorie Variables de nœud.

uxadd NVAR : ajoute une variable au nœud spécifié.

Item	Format	Défaut
VARNAME=	256c	
VVALUE=	1024c	
OVERWRITE=Y N	1c	

Exemple :

```
uxadd nvar varname=var1 vvalue=val1
```

uxupd NVAR : modifie la valeur d'une variable de nœud.

Item	Format	Défaut
VARNAME=	256c *	
VVALUE=	1024c	
OVERWRITE=Y N	1c	Y

Exemple :

```
uxupd nvar varname=var1 vvalue=new overwrite=Y
```

uxdlt NVAR : supprime une variable du nœud spécifié.

Item	Format	Défaut
VARNAME=	256c *	

Exemple :

```
uxdlt nvar varname=VAR1
```

uxlst NVAR : liste les variables de nœuds.

Item	Format	Défaut
VARNAME=	256c *	*
OUTPUT=file_name		

Exemple :

```
uxlst NVAR OUTPUT="c:\temp\LstVarNode.txt"
```

3.4.1 Description des items

NODE : code nœud. Format : 64 caractères alphanumériques.

VARNAME : nom de la variable de nœud. Format : 256 caractères alphanumériques. Obligatoire sauf pour `uxlst`. Valeur générique acceptée sauf pour `uxadd`.

VVALUE : valeur de la variable. Obligatoire pour `uxadd` et `uxupd`.

OVERWRITE : écrasement de la valeur de la variable si elle existe déjà.

- Y : (par défaut) la valeur existante est écrasée par la nouvelle valeur.
- N : la valeur existante n'est pas écrasée par la nouvelle valeur et la commande se termine en erreur (si la variable existe déjà).

OUTPUT : stockage des résultats. Si l'item n'est pas renseigné, la sortie se fait à l'écran.

3.4.2 Exemples

```
uxadd NVAR VARNAME=NEW_VAR VVALUE="value"
```

Crée la variable de nœud `NEW_VAR` ayant comme valeur "value" pour le nœud courant.

```
uxupd NVAR VARNAME=NEW_VAR VVALUE="new_value" OVERWRITE=Y
```

Modifie la variable de nœud `NEW_VAR` avec la valeur "new_value" pour le nœud courant.

3.5 Les types d'unités de gestion

uxadd MUT : crée un type d'unité de gestion dans la table des types d'UG.

Item	Format	Défaut
MUT=	1c	
LABEL=	20c	

Exemple :

```
uxadd MUT MUT=A LABEL=Agencies
```

uxupd MUT : modifie les caractéristiques d'un type d'unité de gestion.

Item	Format	Défaut
MUT=	1c *	
LABEL=	20c	

Exemple :

```
uxupd MUT MUT=S LABEL=Stocks
```

uxshw MUT : affiche les caractéristiques d'un type d'unité de gestion.

Item	Format	Défaut
MUT=	1c *	
OUTPUT=file_name		

Exemple :

```
uxshw MUT MUT=A
```

uxdlt MUT : supprime un type d'unité de gestion de la table des types.

Item	Format	Défaut
MUT=	1c *	

Exemple :

```
uxdlt MUT MUT=T
```

uxlst MUT : liste la table des types d'unité de gestion.

Item	Format	Défaut
MUT=	1c *	*
INF FULL		FULL
OUTPUT=file_name		

Exemple :

```
uxlst MUT OUTPUT="c:\temp\LstMut.txt"
```

3.5.1 Description des items

MUT : Code du type de l'unité de gestion. Format : 1 caractère alphanumérique. Saisie d'un code type (valeur générique acceptée sauf pour uxadd).

LABEL : Libellé descriptif du type de l'unité de gestion. Format : 20 caractères alphanumériques.

INF : Option d'affichage. Limite l'affichage au nom des objets. Autre valeur : FULL (par défaut). Affiche toutes les informations disponibles sur la liste des objets.

 INF et FULL sont incompatibles entre eux.

OUTPUT : Stockage des résultats. Si l'item n'est pas renseigné la sortie se fait à l'écran.

3.5.2 Exemples

```
uxadd MUT MUT=A LABEL=Agences
uxadd MUT MUT=S LABEL=Siège
```

Crée les types d'unités de gestion "A" de libellé "Agences" et "S" de libellé "Siège".

```
uxlst MUT
```

Liste l'ensemble des types d'unités de gestion définis avec les libellés associés.

3.6 Les unités de gestion

uxadd MU : crée une unité de gestion dans la table des unités de gestion.

Item	Format	Défaut
MU=	64c	
TNODE=	64c	
LABEL=	64c	
DEV NODEV		DEV
PROD NOPROD		PROD
TZ=	+/-4n	+0000
USER=	64c	

Exemple :

```
uxadd MU MU=HEADQUART LABEL=\"Company headquarters\" TNODE=%S_NODENAME%
```

uxdup MU : duplique une Unité de Gestion avec toutes ses caractéristiques.

Item	Format	Default
MU=	64c	
TMU=	64c	
LABEL=	64c	

Exemple :

```
uxdup MU MU=HEADQUART TMU=LOCAL LABEL=\"Bureau local\"
```

uxupd MU : modifie les caractéristiques d'une unité de gestion.

Item	Format	Défaut
MU=	64c *	
TNODE=	64c	
LABEL=	64c	
DEV NODEV		
PROD NOPROD		
TZ=	+/-4n	
USER=	64c	

Exemple :

```
uxupd MU MU=%S_NODENAME% LABEL=\"Local node\"
```

uxshw MU : affiche les caractéristiques d'une unité de gestion.

Item	Format	Défaut
MU=	64c *	
TNODE=	64c *	*

Item	Format	Défaut
TZ=	+/-4n	*
USER=	64c *	*
OUTPUT=file_name		

Exemple :

```
uxshw MU MU=A_TOURS
```

uxdlit MU : supprime une unité de gestion de la table des unités de gestion.

Item	Format	Défaut
MU=	64c *	
TNODE=	64c *	*
TZ=	+/-4n	*
USER=	64c *	*

Exemple :

```
uxdlit MU MU=S_TOURS
```

uxlst MU : liste la table des unités de gestion.

Item	Format	Défaut
MU=	64c *	*
TNODE=	64c *	*
TZ=	+/-4n	*
INF FULL LOCAL		INF
USER=	64c *	
OUTPUT=file_name		

Exemple :

```
uxlst MU OUTPUT="c:\temp\LstMu.txt"
```

3.6.1 Description des items

MU : Code de l'unité de gestion. Format : 64 caractères alphanumériques. Saisie d'un code unité de gestion (valeur générique acceptée sauf pour uxadd et uxdup).

TMU : saisissez le code de l'Unité de Gestion cible de la duplication. Format : 64 caractères alphanumériques.

LABEL : Libellé descriptif de l'unité de gestion. Format : 64 caractères alphanumériques.

DEV : Site de développement. Défaut : DEV.

- DEV : l'unité de gestion peut être utilisée dans les espaces application et intégration.
- NODEV : l'unité de gestion ne peut pas être utilisée dans les espaces application et intégration.

PROD : Site de production. Défaut : PROD.

- PROD : l'unité de gestion peut être utilisée dans les espaces simulation et exploitation.
- NOPROD : l'unité de gestion ne peut pas être utilisée dans les espaces simulation et exploitation.

TNODE : Nœud de résidence de l'unité de gestion. Format : 64 caractères alphanumériques. Obligatoire avec uxadd. Critère de sélection avec uxshw et uxlst.

TZ : Optionnel, décalage entre l'heure de l'Unité de Gestion et l'heure système. Critère de sélection avec uxshw et uxlst. Format :

- +/-HHMM : "+" ou "-" suivi du nombre d'heures sur 2 caractères et du nombre de minutes sur 2 caractères.
- Ou "Continent/Ville", la syntaxe est celle de l'IANA (<http://www.iana.org/time-zones>). Par exemple : "America/Montreal", "Europe/Paris", "Asia/Hong_Kong"... La sélection d'un fuseau horaire pour une UG permet d'avoir un décalage horaire qui varie avec l'heure été / hiver sans intervention de l'utilisateur. De plus, il n'y a aucune dépendance avec le fuseau horaire de la machine.

 Ce décalage est systématiquement prise en compte par les automates : calcul de la planification, création d'un lancement, reprise d'une exécution...

USER : Compte de soumission par défaut associé à l'unité de gestion. Format : 64 caractères alphanumériques.

INF : Option d'affichage par défaut. Limite l'affichage au nom et à la description des objets. Autre valeur : FULL. Affiche toutes les informations disponibles sur la liste des objets: nom, description, nœud, décalage heure UG. Si LOCAL est indiqué, seules les unités de gestion du nœud local seront listées.

 Les items INF, FULL et LOCAL sont incompatibles entre eux.

OUTPUT : Stockage des résultats. Si l'item n'est pas renseigné la sortie se fait à l'écran.

3.6.2 Exemples

```
uxadd MU mu=A_TOURS label=\ "Agence de Tours\ " TNODE=Asterix
uxadd MU MU=A_LENS label=\ "Agence de Lens\ " TNODE=%S_NODENAME%
uxadd MU MU=SIEGE label=\ "Siege societe\ " TNODE=%S_NODENAME%
```

Crée l'unité de gestion "A_TOURS", de type "A", de libellé "Agence de Tours" définie sur le nœud distant Asterix, crée les unités de gestion A_LENS et SIEGE définies sur le nœud local.

```
uxadd MU MU=MU_NY label=\ "MU for NY\ " TNODE=%S_NODENAME% TZ=-0600
uxadd MU MU=MU_NDL label=\ "MU for NDL\ " TNODE=%S_NODENAME% TZ=+0300
```

MU_NY et MU_NDL sont situées sur un serveur basé à Paris. New York a six heures de moins que Paris, aussi le décalage de l'unité de gestion MU_NY sera de "moins 6 heures". New Delhi a 3 heures de plus que Paris, aussi le décalage de l'unité de gestion MU_NDL sera de "plus 3 heures". Une tâche planifiée à 20h sur MU_NY sera calculée pour être lancée à 02h00 heure de Paris. La même tâche planifiée à 20h sur MU_NDL sera calculée pour être lancée à 17h00 heure de Paris.

```
uxupd MU MU=MU_NY USER=administrator
```

Modifie l'unité de gestion MU_NY en définissant le compte de soumission par défaut : administrator.

```
uxshw MU MU=A_TOURS
```

Affiche la définition de l'unité de gestion A_TOURS.

```
uxshw MU mu=* tz=+0500
```

Affiche toutes les unités de gestion ayant un décalage de "plus 5 heures".

```
uxlst MU LOCAL
```

Liste les unités de gestions résidant sur le nœud local, A_TOURS n'y figure donc pas.

3.7 Les dépendances des unités de gestion

uxadd HDP : crée une dépendance dans la table des dépendances des unités de gestion.

Item	Format	Défaut
MU=	64c	
DEPMU=	64c	

Exemple :

```
uxadd HDP MU=HEADQUART DEPMU=A_TOURS
```

uxshw HDP : affiche les caractéristiques d'une dépendance.

Item	Format	Défaut
MU=	64c *	
DEPMU=	64c *	
OUTPUT=file_name		

Exemple :

```
uxshw HDP MU=HEADQUART DEPMU=*
```

uxdlit HDP : supprime une dépendance de la table des dépendances des unités de gestion.

Item	Format	Défaut
MU=	64c *	
DEPMU=	64c *	

Exemple :

```
uxdlit HDP MU=A_TOURS DEPMU=A_LENS
```

uxlst HDP : liste la table des dépendances des unités de gestion.

Item	Format	Défaut
MU=	64c *	*
DEPMU=	64c *	*
OUTPUT=file_name		

Exemple :

```
uxlst HDP OUTPUT=c:\temp\LstHdp.txt
```

3.7.1 Description des items

MU : unité de gestion origine de la dépendance. Format : 64 caractères alphanumériques. Saisie d'un code unité de gestion (valeur générique acceptée sauf pour uxadd).

DEPMU : unité de gestion dépendante de l'unité de gestion origine. Format : 64 caractères alphanumériques. Saisie d'un code unité de gestion (valeur générique acceptée sauf pour uxadd).

OUTPUT : Stockage des résultats. Si l'item n'est pas renseigné la sortie se fait à l'écran.

3.7.2 Exemples

```
uxadd HDP MU=SIEGE DEPMU=A_TOURS
uxadd HDP MU=SIEGE DEPMU=A_LENS
```

Crée une dépendance entre l'unité de gestion SIEGE et les unités de gestion A_TOURS et A_LENS.

```
uxlst HDP MU=SIEGE
```

Affiche la liste des dépendances de l'unité de gestion SIEGE.

3.8 Les variables d'Unités de Gestion

uxadd MUVAR : crée une variable pour une Unité de gestion.

Item	Format	Défaut
MU=	64c	
VARNAME=	256c	
VVALUE=	1024c	
OVERWRITE=Y N	1c	

Exemple :

```
uxadd muvar mu=HEADQUART varname=VAR1 vvalue=val1
```

uxupd MUVAR : modifie la valeur d'une variable d'une Unité de Gestion.

Item	Format	Défaut
MU=	64c	
VARNAME=	256c *	
VVALUE=	1024c	
OVERWRITE=Y N	1c	Y

Exemple :

```
uxupd muvar mu=HEADQUART varname=VAR1 vvalue=val2 overwrite=Y
```

uxdlt MUVAR : supprime une variable d'une Unité de Gestion.

Item	Format	Défaut
MU=	64c	
VARNAME=	256c *	

Exemple :

```
uxdlt nvar mu=HEADQUART varname=VAR1
```

uxlst MUVAR : liste les variables d'une Unité de Gestion.

Item	Format	Défaut
MU=	64c	
VARNAME=	256c *	*
OUTPUT=file_name		

Exemple :

```
uxlst muvar mu=HEADQUART OUTPUT="c:\temp\LstMuVar.txt"
```

3.8.1 Description des items

MU : nom de l'Unité de Gestion sur laquelle porte la variable. Format : 64 caractères alphanumériques.

VARNAME : nom de la variable d'UG. Format : 256 caractères alphanumériques. Obligatoire sauf pour uxlst. Valeur générique acceptée sauf pour uxadd.

VVALUE : valeur de la variable. Obligatoire pour uxadd et uxupd.

OVERWRITE : écrasement de la valeur de la variable si elle existe déjà.

- Y : par défaut, la valeur existante est écrasée par la nouvelle valeur.
- N : la valeur existante n'est pas écrasée par la nouvelle valeur et la commande se termine en erreur (si la variable existe déjà).

OUTPUT : stockage des résultats. Si l'item n'est pas renseigné, la sortie se fait à l'écran.

3.8.2 Exemples

```
uxadd MUVAR MU=HEADQUART VARNAME=NEW_VAR VVALUE="value"
```

Crée la variable d'UG NEW_VAR ayant comme valeur "value" pour l'Unité de Gestion HEADQUART.

```
uxupd MUVAR MU=HEADQUART VARNAME=NEW_VAR VVALUE="new_value" OVERWRITE=Y
```

Modifie la variable d'UG NEW_VAR avec la valeur valeur "new_value" pour d'Unité de Gestion HEADQUART.

3.9 Les domaines

uxadd DOM : crée un domaine dans la table des domaines.

Item	Format	Défaut
DOM=	1c	
LABEL=	64c	" "

Exemple :

```
uxadd DOM DOM=T LABEL=\"Technical activities\"
```

uxupd DOM : modifie les caractéristiques d'un domaine.

Item	Format	Défaut
DOM=	1c *	

Item	Format	Défaut
LABEL=	64c	

Exemple :

```
uxupd DOM DOM=V LABEL=\"General services\"
```

uxshw DOM : affichage les caractéristiques d'un domaine.

Item	Format	Défaut
DOM=	1c *	
OUTPUT=file_name		

Exemple :

```
uxshw DOM DOM=*
```

uxdlt DOM : supprime un domaine de la table des domaines.

Item	Format	Défaut
DOM=	1c *	

Exemple :

```
uxdlt DOM DOM=V
```

uxlst DOM : liste la table des domaines.

Item	Format	Défaut
DOM=	1c *	*
INF FULL		FULL
OUTPUT=file_name		

Exemple :

```
uxlst DOM OUTPUT=\"c:\\temp\\LstDom.txt\"
```

3.9.1 Description des items

DOM : Code du domaine. Saisie d'un code domaine (valeur générique acceptée sauf pour uxadd). Format : 1 caractère alphabétique..

LABEL : Libellé descriptif du domaine. Format : 64 caractères alphanumériques.

INF : Option d'affichage. Limite l'affichage au nom des objets. Autre valeur : FULL (par défaut). Affiche toutes les informations disponibles sur la liste des objets.

OUTPUT : Stockage des résultats. Si l'item n'est pas renseigné la sortie se fait à l'écran.

3.9.2 Exemples

```
uxadd DOM DOM=T LABEL=\"Activites techniques\"
```

Crée un domaine T de libellé "Activités techniques".

```
uxlst DOM
```

Liste les domaines définis et leurs libellés.

3.10 Les applications

uxadd APPL : crée une application dans la table des applications.

Item	Format	Défaut
APPL=	64c	
LABEL=	64c	
DOM=	1c	

Exemple :

```
uxadd APPL APPL=DB DOM=T LABEL=\"Data Base\"
```

uxupd APPL : modifie les caractéristiques d'une application.

Item	Format	Défaut
APPL=	64c *	
LABEL=	64c	
DOM=	1c	

Exemple :

```
uxupd APPL APPL=BK DOM=T
```

uxshw APPL : affiche les caractéristiques d'une application.

Item	Format	Défaut
APPL=	64c *	
OUTPUT=file_name		

Exemple :

```
uxshw APPL APPL=BK
```

uxdlt APPL : supprime une application de la table des applications.

Item	Format	Défaut
APPL=	64c *	

Exemple :

```
uxdlt APPL APPL=DB
```

uxlst APPL : liste la table des applications.

Item	Format	Défaut
APPL=	64c	*
INF FULL		FULL
OUTPUT=file_name		

Exemple :

```
uxlst APPL OUTPUT="c:\temp\LstApp1.txt"
```

3.10.1 Description des items

APPL : Code de l'application. Format : 64 caractères alphanumériques. Saisie d'un code application (valeur générique acceptée sauf pour uxadd).

LABEL : Libellé descriptif de l'application. Format : 64 caractères alphanumériques.

DOM : Code du domaine auquel l'application est associée. Format : 1 caractère alphanumérique.

INF : Option d'affichage. Limite l'affichage au nom des objets. Autre valeur : FULL (par défaut). Affiche toutes les informations disponibles sur la liste des objets.

 INF et FULL sont incompatibles entre eux.

OUTPUT : Stockage des résultats. Si l'item n'est pas renseigné la sortie se fait à l'écran.

3.10.2 Exemples

```
uxadd APPL APPL=BK DOM=T LABEL=Backup
```

Crée une application BK de libellé "Backup" appartenant au domaine T.

```
uxshw APPL APPL=BK
```

Affiche la définition de l'application Backup.

3.11 Les répertoires application

uxadd APD : crée un répertoire application dans la table des répertoires application.

Item	Format	Défaut
APPL=	64c	
TNODE=	64c	
DDIR=	60c	
PDIR=	60c	

Exemple :

```
uxadd APD APPL=BK TNODE=Asterix DDIR="d:\temp\dev" PDIR="d:\temp\expl"
```

uxupd APD : modifie les caractéristiques d'un répertoire application.

Item	Format	Défaut
APPL=	64c *	
TNODE=	64c *	
DDIR=	60c	
PDIR=	60c	

Exemple :

```
uxupd APD APPL=BK TNODE=%S_NODENAME% PDIR="c:\temp\prod"
```

uxshw APD : affiche les caractéristiques d'un répertoire application.

Item	Format	Défaut
APPL=	64c *	
TNODE=	64c *	
OUTPUT=file_name		

Exemple :

```
uxshw APD APPL=BK TNODE=*
```

uxdlt APD : supprime un répertoire application de la table des répertoires application.

Item	Format	Défaut
APPL=	64c *	
TNODE=	64c *	

Exemple :

```
uxdlt APD APPL=BK TNODE=Asterix
```

uxlst APD : liste les répertoires application.

Item	Format	Défaut
APPL=	64c *	*
TNODE=	64c *	*
DEV PROD ALL		ALL
OUTPUT=file_name		

Exemple :

```
uxlst APD OUTPUT="c:\temp\LstApd.txt"
```

3.11.1 Description des items

APPL : Code de l'application. Format : 64 caractères alphanumériques. Saisie d'un code application (valeur générique acceptée sauf pour uxadd).

TNODE : Code du nœud. Format : 64 caractères alphanumériques. Saisie d'un code nœud (valeur générique acceptée sauf pour uxadd).

DEV : Univers de développement. Saisie incompatible avec PROD ou ALL. Permet l'affichage des répertoires pour l'univers de développement.

PROD : Univers de production. Saisie incompatible avec DEV ou ALL. Permet l'affichage des répertoires pour l'univers de production.

ALL : Tous les univers. Saisie incompatible avec DEV ou PROD. Permet l'affichage des répertoires de tous les univers.

DDIR : Répertoire pour l'univers de développement. Format : 60 caractères alphanumériques.

PDIR : Répertoire pour l'univers de production. Format : 60 caractères alphanumériques.

OUTPUT : Stockage des résultats. Si l'item n'est pas renseigné la sortie se fait à l'écran.

3.11.2 Exemples

```
uxadd APD APPL=BK TNODE=%S_NODENAME% DDIR=c:\temp\dev PDIR=c:\temp\expl
```

Crée des répertoires pour l'application BK sur le nœud local, le répertoire pour l'univers de développement est "c:\temp\dev" et pour l'univers de production est "c:\temp\expl".

```
uxshw APD APPL=BK TNODE=*
```

Affiche la définition des répertoires de l'application BK.

```
uxupd APD APPL=BK TNODE=%S_NODENAME% PDIR=c:\temp\prod
```

Modifie le répertoire de l'application BK sur le nœud local pour l'univers de production en "c:\temp\prod".

```
uxlst APD
```

Liste les répertoires application définis.

3.12 Les répertoires unités de gestion

uxadd MUD : crée un répertoire unité de gestion dans la table des répertoires unité de gestion.

Item	Format	Défaut
MU=	64c	
APPL=	64c	
XDIR=	60c	
SDIR=	60c	
IDIR=	60c	
ADIR=	60c	

Exemple :

```
uxadd MUD MU=HEADQUART APPL=BK XDIR="c:\temp\BK\exp" SDIR="c:\temp\BK\sim"
IDIR="c:\temp\BK\int" ADIR="c:\temp\BK\app"
```

uxupd MUD : modifie les caractéristiques d'un répertoire UG.

Item	Format	Défaut
MU=	64c *	
APPL=	64c *	
XDIR=	60c	
SDIR=	60c	
IDIR=	60c	
ADIR=	60c	

Exemple :

```
uxupd MUD MU=HEADQUART APPL=BK XDIR="c:\temp\BK\prod"
```

uxshw MUD : affiche les caractéristiques d'un répertoire UG.

Item	Format	Défaut
MU=	64c *	
APPL=	64c *	
APP INT SIM EXP		
OUTPUT=file_name		

Exemple :

```
uxshw MUD MU=HEADQUART APPL=*
```

uxdlit MUD : supprime un répertoire UG de la table des répertoires UG. L'effacement est réalisé pour tous les espaces.

Item	Format	Défaut
MU=	64c *	
APPL=	64c *	

Exemple :

```
uxdlit MUD MU=HEADQUART APPL=PR
```

uxlst MUD : affiche, par espace, les répertoires des unités de gestion.

Item	Format	Défaut
MU=	64c *	*
APPL=	64c *	*
APP INT SIM EXP ALL		ALL
OUTPUT=file_name		

Exemple :

```
uxlst MUD OUTPUT="c:\temp\LstMud.txt"
```

3.12.1 Description des items

MU : Code de L'unité de gestion. Format : 64 caractères alphanumériques. Saisie d'un code unité de gestion (valeur générique acceptée sauf pour uxadd).

APPL : Code de l'application. Format : 64 caractères alphanumériques. Saisie d'un code application (valeur générique acceptée sauf pour uxadd).

APP : Espace application. Saisie incompatible avec INT, SIM, EXP ou ALL. Permet l'affichage des répertoires pour l'espace application.

INT : Espace intégration. Saisie incompatible avec APP, SIM, EXP ou ALL. Permet l'affichage des répertoires pour l'espace intégration.

SIM : Espace simulation. Saisie incompatible avec APP, INT, EXP ou ALL. Permet l'affichage des répertoires pour l'espace simulation.

EXP : Espace exploitation. Saisie incompatible avec APP, INT, SIM ou ALL. Permet l'affichage des répertoires pour l'espace exploitation.

ALL : Tous les espace. Saisie incompatible avec APP, INT, SIM ou EXP. Permet l'affichage des répertoires pour tous les espaces.

ADIR : Répertoire pour l'espace application. Format : 60 caractères alphanumériques.

IDIR : Répertoire pour l'espace intégration. Format : 60 caractères alphanumériques.

SDIR : Répertoire pour l'espace simulation. Format : 60 caractères alphanumériques.

XDIR : Répertoire pour l'espace exploitation. Format : 60 caractères alphanumériques.

OUTPUT : Stockage des résultats. Si l'item n'est pas renseigné la sortie se fait à l'écran.

3.12.2 Exemples

```
uxadd MUD MU=SIEGE APPL=BK XDIR=c:\temp\exp SDIR=c:\temp\sim  
IDIR=c:\temp\int ADIR=c:\temp\app
```

Crée des répertoires pour l'unité de gestion SIEGE pour l'application BK, le répertoire pour l'espace application est "c:\temp\app", le répertoire pour l'espace intégration est "c:\temp\int", le répertoire pour l'espace simulation est "c:\temp\sim" et le répertoire pour l'espace exploitation est "c:\temp\exp".

```
uxshw MUD MU=SIEGE APPL=*
```

Affiche la définition des répertoires de l'unité de gestion SIEGE pour toutes les applications.

3.13 Les profils utilisateurs

Ces commandes sont présentes pour compatibilité uniquement. Elles sont sans effet avec Dollar Universe v6.

uxadd PROF : crée un profil utilisateur dans la table des profils.

Item	Format	Défaut
PROF=	8c	
LABEL=	40c	

Exemple :

```
uxadd PROF PROF=PROFECA
```

```
uxdlst PROF PROF=PROFECA
```

uxlst PROF : liste les profils utilisateurs.

Item	Format	Défaut
PROF=	8c *	
OUTPUT=file_name		

Exemple :

```
uxlst PROF PROF=*
```

uxext AUT : extrait la définition d'un profil.

Item	Format	Défaut
AUT		
PROFIL=	8c *	
OUTPUT=file_name		
REPL		
NOVERIF		

Exemple :

```
uxext AUT PROFIL=* OUTPUT="c:\temp\Profil.ext" REPL
```

uxins AUT : insère la définition d'un profil dans les fichiers correspondants de Dollar Universe pour la même version de Dollar Universe.

Item	Format	Défaut
AUT		
PROFIL=	8c *	
INPUT=file_name		
REPL		

Exemple :

```
uxins AUT PROFIL=* INPUT="c:\temp\Profil.ext"
```

3.13.1 Description des items

PROFIL : code du profil. Format : 8 caractères alphanumériques maximum. Saisie d'un code profil (valeur générique acceptée sauf pour uxadd).

LABEL : libellé descriptif du profil. Format : 40 caractères alphanumériques.

REPL : option de remplacement. Indique si l'enregistrement extrait (inséré) annule et remplace l'enregistrement déjà existant dans le fichier de destination; par défaut, l'extraction (l'insertion) est refusée pour chaque enregistrement déjà existant (signalé par un message pour chaque occurrence).

OUTPUT : stockage des résultats. Le nom du fichier est obligatoire pour la commande uxext. Si l'item n'est pas renseigné la sortie se fait à l'écran.

INPUT : fichier d'entrée pour la commande uxins. Le nom du fichier est obligatoire.

NOVERIF : peut être utilisé avec la commande `uxext` uniquement. Cette option permet de ne pas vérifier le fichier l'extraction avant d'y écrire de nouveaux enregistrements et d'améliorer ainsi les performances de cette commande. Cet item ne doit être utilisé que pour un nouveau fichier.

3.14 Les utilisateurs

Les commandes utilisant l'objet `USER` ciblent uniquement les comptes de soumission dans Dollar Universe. Les utilisateurs systèmes correspondants ne sont pas gérés par ces commandes. Les droits de connexion des utilisateurs (UVC, mode commandes) sont définis par UVMS, reportez-vous au Manuel d'Administration d'UniViewer pour la description de ces commandes.

uxadd USER : crée un compte de soumission dans Dollar Universe. Sous Windows le service utilisateur est également créé et démarré (si le mot de passe est fourni).

Item	Format	Défaut
USER=	64c	
CODE=	3c	
PROF=	8c	
DEV PROD BOTH		
TYPE=WIN OTH		
SYSUSER=	256c	
SYSPWD=	64c	
LABEL=	64c	

Exemple :

```
uxadd USER USER=eca TYPE=WIN SYSUSER=Administrateur SYSPWD=123456
```

uxupd USER : modifie les caractéristiques d'un compte de soumission.

Item	Format	Défaut
USER=	64c *	
PROF=	8c	
SYSUSER=	256c	
SYSPWD=	64c	
LABEL=	64c	

Exemple :

```
uxupd USER USER=eca SYSUSER=LocalSystem
```

uxshw USER : affiche les caractéristiques d'un compte de soumission.

Item	Format	Défaut
USER=	64c *	
TYPE=WIN OTH		*
CODE=	3c	*
DEV PROD BOTH		

Item	Format	Défaut
OUTPUT=file_name		

Exemple :

```
uxshw USER USER=eca
```

uxdlit USER : supprime un compte de soumission. Sous Windows, le service utilisateur correspondant est également supprimé .

Item	Format	Défaut
USER=	64c *	

Exemple :

```
uxdlit USER USER=eca
```

uxlst USER : liste les comptes de soumission.

Item	Format	Défaut
USER=	64c *	*
CODE=	3c *	
DEV PROD BOTH		
INF FULL		FULL
OUTPUT=file_name		

Exemple :

```
uxlst USER OUTPUT="c:\temp\LstUser.txt"
```

3.14.1 Description des items

USER : Nom du compte de soumission dans Dollar Universe. Format : 64 caractères alphanumériques (valeur générique acceptée sauf pour uxadd).

TYPE : type d'utilisateur. WIN pour les systèmes Windows, OTH pour les autres systèmes. Valeur par défaut :

- Sous Windows : WIN
- Pour les autres systèmes : OTH

SYSUSER : Nom de l'utilisateur système, il doit exister sur le système. Format 256 caractères. Valeur par défaut :

- Sous UNIX/OpenVMS : identique à USER
- Sous Windows : LocalSystem ou nom du compte utilisateur

SYSPWD : sous Windows 32 uniquement, mot de passe du service utilisateur. Obligatoire si SYSUSER n'est pas LocalSystem.

LABEL: Description. Format 64 caractères max.

CODE : Pour compatibilité uniquement. Code auteur associé à l'utilisateur. Format : 3 caractères de 0 à 9 et de A à Z (majuscules).

PROF : Pour compatibilité uniquement. Profil associé à l'utilisateur. Format : 8 caractères alphanumériques.

DEV : Pour compatibilité uniquement. Type du compte de soumission. Pour la commande `uxadd`, une des trois valeurs suivantes est obligatoire. Il n'y a pas de valeur par défaut.

- **DEV** : l'utilisateur ne peut être utilisé comme compte de soumission que dans les espaces application et intégration.
- **PROD** : l'utilisateur ne peut être utilisé comme compte de soumission que dans les espaces simulation et exploitation.
- **BOTH** : l'utilisateur peut être utilisé comme compte de soumission dans tous les espaces.

INF : Option d'affichage. Limite l'affichage au nom des objets. Autre valeur : **FULL** (par défaut). Affiche toutes les informations disponibles sur la liste des objets.

OUTPUT : Stockage des résultats. Si l'item n'est pas renseigné la sortie se fait à l'écran.

3.14.2 Exemples

```
uxadd user user=eca type=WIN sysuser=localsystem
```

Crée le compte de soumission "eca" sous Windows et le service utilisateur associé en tant que LocalSystem.

```
uxadd USER USER=admin
```

Crée le compte de soumission "admin"

```
uxshw USER USER=admin
```

Affiche la définition du compte de soumission admin :

```
Command : uxshw user user=admin
```

```
TYPE | user
ITEMS | user      : admin
      | type      : WIN
      | prof      : PROFADM
      | both
      | code      : ad1
      | sysuser   : LocalSystem
      | syspwd    : *****
      | user service : STARTED (for LocalSystem)
```

3.15 Les classes

uxadd

uxupd CLASS : modifie les caractéristiques d'une classe.

Item	Format	Défaut
CLASS=	64c *	
LABEL=	64c	

Exemple :

```
uxupd CLASS CLASS=EDIT LABEL=\"General printouts\"
```

uxshw CLASS : affiche les caractéristiques d'une classe.

Item	Format	Défaut
CLASS=	64c *	
OUTPUT=file_name		

Exemple :

```
uxshw CLASS CLASS=EDIT
```

uxdlt CLASS : supprime une classe de la table des classes.

Item	Format	Défaut
CLASS=	64c *	

Exemple :

```
uxdlt CLASS CLASS=UPDATE
```

uxlst CLASS : liste la table des classes.

Item	Format	Défaut
CLASS=	64c *	*
INF FULL		FULL
OUTPUT=file_name		

Exemple :

```
uxlst CLASS OUTPUT=\"c:\temp\LstClass.txt\"
```

3.15.1 Description des items

CLASS : Code de la classe. Saisie d'un code classe (valeur générique acceptée sauf pour uxadd). Format : 64 caractères alphanumériques.

LABEL : Libellé descriptif de la classe. Format : 64 caractères alphanumériques.

INF : Option d'affichage. Limite l'affichage au nom des objets. Autre valeur : FULL (par défaut). Affiche toutes les informations disponibles sur la liste des objets.

OUTPUT : Stockage des résultats. Si l'item n'est pas renseigné la sortie se fait à l'écran.

3.15.2 Exemples

```
uxadd CLASS CLASS=EDIT LABEL=Editions
```

```
uxadd CLASS CLASS=UPDATE LABEL=\"Mises a jour\"
```

Crée les classes EDIT de libellé "Editions" et UPDATE de libelle "Mises à jour".

```
uxlst CLASS
```

Liste les classes définies.

3.16 Les ressources

uxadd RES : crée une ressource dans la liste des ressources.

Item	Format	Défaut
RES=	64c	
LABEL=	64c	" "
NATURE= FIL LOG SAE SAJ SCR SYS SPM	3c	
FNAME=file_name	60c*	
DIR=dir_name	256c*	
LOCK UNLOCK		UNLOCK
CLEAR NOCLEAR		CLEAR
REGEXP NOREGEXP		NOREGEXP
QT1=	4n	0
QT2=	4n	0
FREQ=	30...86400	30
PARAM=	1024c	
VALIDITY=	5n	60

Exemples :

```
uxadd RES RES=TESTFILE LABEL=\"Awaiting test file\" NATURE=FIL
DIR=c:\\temp\\ FNAME=TestFile FREQ=30
uxadd RES RES=D_BASE LABEL=\"Base protection\" NATURE=LOG FNAME=!UG!!ESP!
QT1=10 QT2=10 NOCLEAR
```

uxupd RES : modifie les caractéristiques d'une ressource.

Item	Format	Défaut
RES=	64c *	
LABEL=	64c	
NATURE=	FIL LOG SAE SAJ SCR SYS SPM	3c
FNAME=file_name	60c*	
DIR=dir_name	256c*	
LOCK UNLOCK		
CLEAR NOCLEAR		
REGEXP NOREGEXP		
QT1=	4n	

Item	Format	Défaut
QT2=	4n	
FREQ=	30...86400	
PARAM=	1024c	
VALIDITY=	5n	

Exemple :

```
uxupd RES RES=D_FIL_PAY FNAME=pay_!DTRAIT!
```

uxshw RES : affiche la définition et l'allocation d'une ressource. Reportez-vous également au paragraphe "[Les ressources](#)".

Item	Format	Défaut
RES=	64c *	
ESP=X S I A *	1c * " "	*
MU=	64c * " "	*
TSOC=	6c * " "	*
PDATE=	AAAAMMJJ * " "	*
INF RSV FULL		INF
OUTPUT=file_name		

Exemple :

```
uxshw RES RES=D_*
```

uxdlt RES : supprime une ressource.

Item	Format	Défaut
RES=	64c *	

Exemple :

```
uxdlt RES RES=TESTFILE
```

uxlst RES : liste les ressources.

Item	Format	Défaut
RES=	64c *	*
INF FULL		FULL
OUTPUT=file_name		

Exemple :

```
uxlst RES OUTPUT="c:\temp\LstRes.txt"
```

uxext RES : extrait la définition complète d'une ressource.

Item	Format	Défaut
RES=*	64c *	
REPL		

Item	Format	Défaut
OUTPUT=file_name		
NOVERIF		

Exemple :

```
uxext RES RES=D_* OUTPUT="c:\temp\Res.ext"
```

uxins RES : insère la définition complète d'une ressource dans les fichiers correspondants de Dollar Universe pour la même version de Dollar Universe.

Item	Format	Défaut
RES=*	64c *	
REPL		
INPUT=file_name		

Exemple :

```
uxins RES RES=* INPUT="c:\temp\Res.ext"
```

3.16.1 Description des items

RES : référence de la ressource. Format : 64 caractères alphanumériques. Requiert la saisie de la référence de la ressource (valeur générique acceptée sauf pour uxadd).

LABEL : libellé descriptif de la ressource. Format : 64 caractères alphanumériques.

NATURE : définit la nature de la ressource. Sur UNIX, Windows et OpenVMS, les valeurs possibles sont FIL (fichier), LOG (logique), SAE (SAP event), SAJ (SAP job), SCR (script), SYS (Système). SPM (SP Monitor) est seulement disponible sous UNIX et Windows..

FNAME : nom du fichier ou de la ressource. Format: 60 caractères max. Permet la saisie du nom du fichier associé à une ressource de type FIL ou SCR ou, dans le cas d'une ressource logique, permet la saisie d'un nom de "fichier virtuel" sur lequel repose la vérification de la ressource.

Ce nom peut contenir les variables !UG! ou !MU!, !DTRAIT! ou !PDATE!, !ESP! ou !AREA! et !SOC! ou !COMP! qui seront analysées lors du pilotage de l'Uproc contenant une condition sur cette ressource (en fonction de son contexte d'exécution). Il peut contenir également le caractère générique "**"

Si la nature de la ressource est FIL :

- Les variables !APP!, !SESS! et !UPR! peuvent également être utilisées.
- Le nom du fichier peut contenir des variables saisies entre points d'exclamation, par exemple : !VARIABLE!.
- Utilisé avec le mot clé REGEXP, ce champ correspond à une expression régulière (bibliothèque Tcl/Tk version 8.1 avec l'option "ERE" signifiant que seule la syntaxe 8.0 est supportée; suivez cette URL https://tcl.tk/doc/howto/regexp81.html#part1_2 pour plus de détails).
Les caractères suivants sont supportés pour les expressions régulières :
 - Générique : pour remplacer n'importe quel caractère "."
 - Répétition : zéro ou plusieurs occurrences du caractère ou de la classe précédant "*"
 - Position dans la ligne : début de ligne "^"
 - Position dans la ligne: fin de ligne "\$"
 - Classe de caractères: chaîne qui contient l'un des caractères. Exemple : [clas]

- Classe inversée : chaîne qui ne contient pas l'un des caractères. Exemple : `[^clas]`
- Plage de caractères : tous les caractères dans la plage définie. Exemple : `[0-9]`
- Échappement : pour rechercher le caractère spécial "x". Exemple : `\x`
- Position dans le mot : début du mot. Exemple : `\<xyz`
- Position dans le mot : fin du mot. Exemple : `xyz\>`

REGEXP | **NOREGEXP** : indique qu'une expression régulière est utilisée pour analyser un ou des fichiers. Valeur par défaut **NOREGEXP**.

FREQ : cycle de vérification par le surveillant. Durée de la période de vérification : de 30 à 86400 secondes. La valeur par défaut est 30s.

PARAM : (pour une ressource de type Script uniquement) correspond à la liste des paramètres qui doivent être transmis au script lors de son exécution.

- Les paramètres doivent être séparés par des espaces.
- Ils peuvent contenir des éléments variables de Dollar Universe (!UG!, !DTRAIT!, !SOC! et !ESP!).
- Ils peuvent contenir des variables connues de l'environnement d'exécution de l'Uproc.
- La longueur maximum est de 60 caractères. Une fois les variables traduites, elle ne doit pas dépasser 1024 caractères.

VALIDITY : pour une ressource de type Script (SCR), la fréquence de validité correspond au délai pendant lequel le code retour du script sera considéré comme valide pour la vérification de la condition de ressource. La validité doit être supérieure au cycle de vérification du surveillant.

DIR : localisation du fichier. Format: 256 caractères max. Permet la saisie, pour une ressource de type FIL uniquement, du chemin d'accès au fichier.

Ce nom peut contenir les variables !UG!, !DTRAIT!, !ESP et !SOC! qui seront analysées lors du pilotage de l'Uproc contenant une condition sur cette ressource (en fonction de son contexte d'exécution). Il peut contenir également le caractère générique "*".

UNLOCK : allocation de la ressource. Indique que l'Uproc conditionnée pourra s'attribuer tout ou partie des quotas de cette ressource.

Si le mot clé LOCK est utilisé, l'Uproc conditionnée ne pourra s'attribuer que la totalité de la ressource durant son exécution et aucune autre Uproc, conditionnée par cette ressource ne pourra s'exécuter.

 Usage de LOCK incompatible avec QT1 ou QT2.

QT1 ou **QT2** : quotas d'allocation de niveau 1 et 2 respectivement. Format : de 0 à 9999. Dans les commandes `uxadd` ou `uxupd`, ils permettent de définir les valeurs maximum de quotas qui pourront être alloués lors de l'exécution d'une commande `uxhld` ou lors de l'exécution d'une Uproc portant une condition sur cette ressource. L'utilisation de QT1 et QT2 est incompatible avec LOCK.

CLEAR : libération. Permet de libérer la ressource sélectionnée. Dans les commandes `uxadd` ou `uxupd`, la libération ne sera effective que lors de la terminaison de l'Uproc conditionnée. L'Uproc conditionnée ne libère que les quotas qu'elle a alloués.

NOCLEAR indique que la ressource ne sera pas libérée en fin d'exécution de l'Uproc conditionnée, la commande `uxrls` devra être exécutée pour permettre la libération de la ressource.

INF : informations techniques. Permet l'affichage des informations de définition de la ressource. Autres valeurs : FULL et RSV : cf. chapitre "Les commandes d'exploitation" paragraphe "[Les ressources](#)".

 Les trois items sont incompatibles entre eux.

REPL : option de remplacement. Indique si l'enregistrement extrait (inséré) annule et remplace l'enregistrement déjà existant dans le fichier de destination; par défaut, l'extraction (l'insertion) est refusée pour chaque enregistrement déjà existant (signalé par un message pour chaque occurrence).

OUTPUT : stockage des résultats. Le nom du fichier est obligatoire pour la commande `uxext`. Si l'item n'est pas renseigné la sortie se fait à l'écran.

INPUT : fichier d'entrée pour la commande `uxins`. Le nom du fichier est obligatoire.

NOVERIF : peut être utilisé avec la commande `uxext` uniquement. Cette option permet de ne pas vérifier le fichier l'extraction avant d'y écrire de nouveaux enregistrements et d'améliorer ainsi les performances de cette commande. Cet item ne doit être utilisé que pour un nouveau fichier.

3.16.2 Exemples

```
uxadd RES RES=D_FIL_PAY LABEL=\"Attente fichier paye\" nature=FILE FNAME=paye_!DTRAIT! DIR=c:\temp\ LOCK CLEAR
uxadd RES RES=D_BASE LABEL=\"Protection base\" nature=LOG FNAME=!UG!!ESP! DIR=c:\temp QT1=10 QT2=10 NOCLEAR
```

Crée deux ressources :

- la ressource `D_fil_pay` portant sur l'arrivée du fichier `c:\temp\paye_!DTRAIT!`, la ressource est exclusive et la libération est automatique en fin d'exécution de l'Uproc conditionnée,
- la ressource `D_BASE` portant sur une protection d'accès à une base, la ressource est logique, elle est spécifique à l'espace et à l'unité de gestion d'exécution de l'Uproc conditionnée, elle peut être partagée et devra être libérée par une commande `uxrls` dans le script d'une Uproc.

```
uxadd RES RES=D_FILE LABEL=\"file search\" nature=FILE FNAME= a[1-8][1-8][a|b].[.]txt DIR=c:\temp
```

Crée la ressource `D_FILE` qui utilise une expression régulière pour analyser les fichiers dans le repertoire `c:\temp`. La ressource analyse les fichiers qui commencent par "a", suivi de 2 chiffres, chacun compris entre 1 et 8, "a" ou "b", n'importe quel caractère puis l'extension du fichier `.txt`.

Les fichier suivants correspondent à ces critères : `a34br.txt` et `a51a8.txt`.

Les fichier suivants ne correspondent pas à ces critères : `a81a8.txt` et `a34de.txt`.

```
uxshw RES RES=D_*
```

Affiche la définition des ressources créées précédemment.

```
uxext RES RES=D_* output=c:\temp\DOC.ext
```

Extrait toutes les ressources commençant par `D_` de l'environnement courant et archive les données dans le fichier `c:\temp\DOC.ext`.

```
uxins RES RES=* input=c:\temp\DOC.ext
```

Insère toutes les ressources du fichier `c:\temp\DOC.ext` dans l'environnement courant (Société, nœud).

3.17 Les Uprocs

uxadd UPR : crée une Uproc dans la liste des Uprocs : seules les informations générales sont définies. Les informations complémentaires sont saisies par l'intermédiaire d'une commande de mise à jour de l'Uproc (`uxupd`).

Item	Format	Défaut
EXP SIM INT APP	3c	S_ESPEXE
UPR=	64c	
VUPR=	3n	
LABEL=	64c	" "
FPERIOD=N D W T F M 2 3 Q S Y	1c	N
UPT=CL_INT CL_EXT CMD FTP_GET FTP_PUT...	10c	CL_INT
CLFIL=file_name	128c	
CL_INT=filename	256c	
COMMAND=	256c	
EXECDIR=	256c	
SHELL=	512c	
SHELLARG=	256c	
SHELLTYPE=POWERSHELL PERL CSCSCRIPT DEFAULT	10c	DEFAULT
APPL=	64c	
CLASS=	64c	
MEMO=N O A	1c	N
NBPER=	2n	0
RETRY=(NNN,MM)	(3n,2n)	
DEFINFO=	50c	« »
DEFSEV=	4n	« » ou 0
INTERACTIVE NOINTERACTIVE		NOINTERACTIVE

Exemples :

```
uxadd UPR EXP UPR=D_LOAD_FIL LABEL=\"Loading of file\" FPERIOD=W APPL=BK
MEMO=0 NBPER=1 UPT=CL_EXT CLFIL=\"c:\temp\D_LOAD_FIL.bat\"
uxadd UPR EXP UPR=CMD_UPR LABEL=\"My Command Uproc\" UPT=CMD
COMMAND=netstat EXECDIR=/temp FPERIOD=D APPL=EJ MEMO=0 NBPER=1
```

uxupd UPR : modifie les informations générales d'une Uproc, ajoute des caractéristiques complémentaires telles que des conditions, des consignes, des incompatibilités.

Item	Format	Défaut
EXP SIM INT APP	3c	S_ESPEXE
UPR=	64c *	
VUPR=	3n *	

Une seule des familles décrites ci-dessous peut être utilisée dans une commande uxupd.

Famille	Description
GENINF	Permet la mise à jour des informations générales de l'Uproc.
CTLDUR	Permet la mise à jour des informations de contrôle de la durée d'exécution de l'Uproc

Famille	Description
UPR_STATUS	Permet la définition de l'état de terminaison de l'Uproc.
UPR_NOTIFY	Envoie une notification par e-mail lors de la terminaison du Lancement ou de l'exécution de l'Uproc.
LAUNCHVAR	Permet la création et la suppression des variables de l'Uproc et leur valorisation.
INCCLASS	Permet de définir des classes d'incompatibilité pour l'Uproc. La classe d'appartenance de l'Uproc est définie dans les informations générales.
TERMINES	Permet de définir des consignes de terminaisons qui seront effectuées lors de la terminaison de l'Uproc.
DEPCON	Permet de définir une condition d'enchaînement à l'Uproc ainsi que les informations de détail associées.
RESCON	Permet de définir une condition de ressource à l'Uproc ainsi que les informations de détail associées.
INCCON	Permet de définir une condition de non simultanéité à l'Uproc ainsi que les informations de détail associées.
FORMULA	Permet la mise à jour de la formule de lancement. Cette famille doit être utilisée dès lors qu'une condition a été ajoutée ou supprimée à l'Uproc pour que les conditions soient effectivement examinées lors du pilotage.
SUCCESSOR	Permet de définir l'ordre d'examen des successeurs de l'Uproc courante
FTP_GET	Permet la définition d'une Uproc de type FTP_GET
FTP_PUT	Permet la définition d'une Uproc de type FTP_PUT

GENINF : informations générales

Item	Format	Défaut
LABEL=	64c	
FPERIOD=N D W T F M 2 3 Q S Y	1c	
UPT=CL_INT CL_EXT CMD FTP_GET FTP_PUT...	10c	
CLFIL=file_name	128c	
CL_INT=filename	256c	
COMMAND=	256c	
EXECDIR=	256c	
SHELL=	256c	
SHELLARG=	256c	
SHELLTYPE=POWERSHELL PERL CSCRIPT DEFAULT	10c	
APPL=	64c	
CLASS=	64c	
MEMO=N O A	1c	
NBPER=	2n	
RETRY=(NNN,MM)	(3n,2n)	
DEFINFO=	50c	

Item	Format	Défaut
DEFSEV=	4n	
INTERACTIVE NOINTERACTIVE		
DIR_FILES=	256c	

Exemple :

```
uxupd UPR EXP UPR=D_LOAD_FIL GENINF FPERIOD=D
```

UPR_STATUS : définit l'attribution de l'état de terminaison de l'Uproc.

Item	Format	Défaut
ENABLE=Y N y n	1c	
SET=C A	1c	C
RETCODE=valeur[,EQ LO GT NE]	4n[,2c]	
EXT_RETCODE=n1;n2;[n3,n4];[n5,n6]/n7	64c	
STRING=	128c	
REGEXP NOREGEXP		NOREGEXP
SEARCH_FILE=	256c	Log du Job

Exemples :

```
uxupd UPR EXP UPR=STATUS_UPR UPR_STATUS ENABLE=Y SET=A RETCODE=0, LO
uxupd UPR EXP UPR=STATUS_UPR UPR_STATUS ENABLE=N
uxupd UPR EXP UPR=STATUS_UPR UPR_STATUS ENABLE=Y SET=C STRING=Execution_OK
SEARCH_FILE=/data/result_file.txt
```

CTLDUR : contrôle de la durée d'exécution de l'Uproc

Item	Format	Défaut
MAXDUR=	(HHHMMSS nnn,K F C=256c) NONE	
MINDUR=	(HHHMMSS nnn,R C=256c) NONE	
MAXWDUR=	(HHHMMSS nnn,F A C=256c) NONE	

Exemples :

```
uxupd upr upr=MY_UPROC ctldur maxdur="(119,C=hello)"
uxupd upr upr=MY_UPROC ctldur maxdur="(0010203,F)"
uxupd upr upr=MY_UPROC ctldur maxdur="(0010203,R)"
uxupd upr upr=MY_UPROC ctldur maxdur="(0010203,C=C:\my\own\script.bat)"
uxupd upr upr=MY_UPROC ctldur maxdur=NONE
```

NOTIFY : Définit la notification par e-mail lors de la terminaison du Lancement ou de l'exécution de l'Uproc.

Item	Format	Défaut
MAIL=Y N	1c	N
COMPLETED=Y N	1c	N
ABORTED=Y N	1c	N
REFUSED=Y N	1c	N

Item	Format	Défaut
TIMEOVERRUN=Y N	1c	N
JOBLOG=Y N valeur	1c 4n	N
RECIPIENT="email1;email2..."	256c	vide

Exemple :

```
uxupd UPR EXP UPR=DEF_UPR NOTIFY MAIL=Y ABORTED=Y REFUSED=Y JOBLOG=400
recipient="admin@Automic.com;admin_du@Automic.com"
```

LAUNCHVAR add : ajoute une variable.

Item	Format	Format	Format	Format
VARNAME=	20c	20c	20c	20c
VTYPE=D Q T P	D	Q	P	T (défaut)
VFORMAT=	"YYYY/MM/DD" "MM/DD/YYYY" "DD/MM/YYYY"		3n (longueur)	3n (longueur)
VALMIN=		12n		
VALMAX=		12n		
VVALUE=	"10c"	12n	256c	256c

Exemples :

```
uxupd UPR EXP UPR=D_LOAD_FIL LAUNCHVAR add VARNAME=VARDATE VTYPE=D
VFORMAT="YYYY/MM/DD" VVALUE="2002/07/01"
uxupd UPR EXP UPR=D_LOAD_FIL LAUNCHVAR add VARNAME=VARNUM VTYPE=Q
VVALUE=123
uxupd UPR EXP UPR=D_LOAD_FIL LAUNCHVAR add VARNAME=VARTXT VVALUE="\Last
var\"
uxupd UPR EXP UPR=D_LOAD_FIL LAUNCHVAR add VARNAME=PASSWD VTYPE=P
VVALUE=MyPassword
```

LAUNCHVAR dlt : supprime une variable.

Item	Format	Format
VARNAME=	20c	

Exemple :

```
uxupd UPR EXP UPR=D_LOAD_FIL LAUNCHVAR dlt VARNAME=VARTXT
```

INCCLASS add : ajoute une classe d'incompatibilité.

Item	Format	Défaut
XCLASS=	6c	

Exemple :

```
uxupd UPR EXP UPR=D_LOAD_FIL INCCLASS add XCLASS=BACKUP
```

INCCLASS dlt : supprime une classe d'incompatibilité.

Item	Format	Défaut
XCLASS=	6c	

Exemple :

```
uxupd UPR EXP UPR=D_LOAD_FIL INCCLASS dlt XCLASS=BACKUP
```

TERMINs add : ajoute une consigne de terminaison.

Item	Format	Défaut
TERMNO=	2n	
TERM=	64c	
PDATE=	ANY SAME (±n,n,n) (a, ,n)	SAME
MU=	SAME "mu(MU_CODE)" "mut(T)" "hdp({XY})"	SAME
SESS=	any same samerun expl(SESS_CODE)	ANY
USER=	any same	ANY
STATUS=	terminated abnormal	TERMINATED

Exemple :

```
uxupd UPR EXP UPR=D_LOAD_FIL TERMINs add TERMNO=01 TERM=D_LOAD_FIL PDATE="(-1, , )" STATUS=ABNORMAL
```

TERMINs dlt : supprime une consigne de terminaison.

Item	Format	Défaut
TERMNO=	2n	

Exemple :

```
uxupd UPR EXP UPR=D_LOAD_FIL TERMINs dlt TERMNO=01
```

DEPCON add : ajoute une condition d'enchaînement.

Item	Format	Défaut
CONDNO=	2n	
DEP=	64c	
IFPLANNED		""
PDATE=	ANY SAME (±n,n,n) (a, ,n)	SAME
DTYPE=	A W C H	
MU=	SAME "mu(MU_CODE)" "mut(T)" "hdp({XY})"	SAME
ONE ALL		ALL
SESS=	any same samerun expl(SESS_CODE)	ANY
USER=	ANY SAME	ANY
STATUS=	TERMINATED ABNORMAL ENDED ABSENT	TERMINATED
EXPECTED EXCLUDED		EXPECTED
NOFATAL FATAL		NOFATAL

Exemples :

```
uxupd UPR EXP UPR=D_BACKUP DEPCON add CONDNO=01 DEP=D_LOAD_FIL MU="mu
(SATURN)"
uxupd UPR EXP UPR=D_BACKUP DEPCON add CONDNO=02 DEP=D_LOAD_FIL MU="hdp
({AH})"
```

DEPCON dlt : supprime une condition d'enchaînement. La formule de lancement doit tout d'abord être mise à jour avec la nouvelle formule de lancement complète.

Item	Format	Défaut
CONDNO=	2n	

Exemple :

```
uxupd UPR EXP UPR=D_BACKUP DEPCON dlt CONDNO=01
```

INCCON add : ajoute une condition de non simultanéité.

Item	Format	Défaut
CONDNO=	2n	
INC=	64c	
PDATE=	SAME ANY	SAME
MU=	SAME "mu(MU_CODE)" "mut(T)" "hdp({XY})"	SAME
SESS=	any same samerun expl(SESS_CODE)	ANY
USER=	ANY SAME	ANY
EXPECTED EXCLUDED		EXPECTED
NOFATAL FATAL		NOFATAL

Exemple :

```
uxupd UPR EXP UPR=D_BACKUP INCCON add CONDNO=01 INC=D_BACKUP PDATE=ANY
```

INCCON dlt : supprime une condition de non simultanéité. La formule de lancement doit tout d'abord être mise à jour avec la nouvelle formule de lancement complète.

Item	Format	Défaut
CONDNO=	2n	

Exemple :

```
uxupd UPR EXP UPR=D_BACKUP INCCON dlt CONDNO=02
```

RESCON add : ajoute une condition de ressource.

 Sous Solaris (intel et sparc), la commande /usr/xpg4/bin/grep doit être accessible pour que l'attribut PARSING puisse être utilisé dans la condition de ressource.

Item	Format	Défaut
CONDNO=	2n	
RES=	64c	
PDATE=	ANY SAME (±n,n,n) (a, ,n)	SAME
DTYPE=	A W C H	
MU=	SAME "mu(MU_CODE)" "mut(T)" "hdp({XY})"	SAME

Item	Format	Défaut
ONE ALL		ALL
VERIF NOVERIF		VERIF
UNLOCK LOCK		UNLOCK
EXPECTED EXCLUDED		EXPECTED
NOFATAL FATAL		NOFATAL
ATTRIB=	EXIST SIZE DATEUNCHANGE SIZEUNCHANGE DATCREA DATMOD GROUP PARSING	
OPER=	# = < > <= >=	
VALUE=	255c	
QT1=	4n	0
QT2=	4n	0

Exemple :

```
uxupd UPR EXP UPR=D_LOAD_FIL RESCON add CONDNO=01 RES=D_FIL_PAY
```

RESCON dlt : supprime une condition de ressource. La formule de lancement doit tout d'abord être mise à jour avec la nouvelle formule de lancement complète.

Item	Format	Défaut
CONDNO=	2n	

Exemple :

```
uxupd UPR EXP UPR=D_LOAD_FIL RESCON dlt CONDNO=03
```

SUCCESSOR add : crée un successeur de l'Uproc

Item	Format	Default
SUCC_UPR=	64c	

Exemple:

```
uxupd UPR EXP UPR=UPROC_1 SUCCESSOR add SUCC_UPR=D
```

SUCCESSOR dlt : supprime un successeur de l'Uproc

Item	Format	Default
SUCC_UPR=	64c	

Exemple:

```
uxupd UPR EXP UPR=UPROC_1 SUCCESSOR dlt SUCC_UPR=D
```

FORMULA : définit la formule de lancement.

Item	Format	Défaut
LFORM=	255c	

Exemple :

```
uxupd UPR EXP UPR=D_LOAD_FIL FORMULA LFORM=\"=C01 AND =C02\"
```

FTP_GET : récupération de fichier par FTP.

Item	Format	Défaut
PARTNER=	16c	
ORIGIN_FILENAME=	256c	""
ORIGIN_PATHNAME=	256c	
TARGET_FILENAME=	256c	""
TARGET_PATHNAME=	256c	
LOGON=	32c	anonymous
PASSWORD=	32c	
MODE=ASCII BINARY		ASCII
ACTION=REPLACE ABORT NONE		ABORT
CRLF NOCRLF		NOCRLF
DELETE NODELETE		NODELETE
NEWER NONEWER		NONEWER
RECURSIVE NORECURSIVE		NORECURSIVE
BUFFER_SIZE=	2n	32
RETRY=	1c	
UMASK=	3c	

Exemple :

```
uxupd UPR EXP UPR=GET_FILES FTP_GET PARTNER="WinXP" ORIGIN_PATHNAME="/tmp/"  
CRLF TYPE=BINARY UMASK=700 ACTION=REPLACE RETRY=3
```

FTP_PUT : envoi de fichier par FTP.

Item	Format	Défaut
PARTNER=	16c	
ORIGIN_FILENAME=	256c	""
ORIGIN_PATHNAME=	256c	
TARGET_FILENAME=	256c	""
TARGET_PATHNAME=	256c	
LOGON=	32c	anonymous
PASSWORD=	32c	
MODE=ASCII BINARY		ASCII
ACTION=REPLACE ABORT APPEND NONE		ABORT
DELETE NODELETE		NODELETE
NEWER NONEWER		NONEWER
RECURSIVE NORECURSIVE		NORECURSIVE

Item	Format	Défaut
BUFFER_SIZE=	2n	32
RETRY=	1c	

Exemple :

```
uxupd UPR EXP UPR=PUT_FILE FTP_PUT PARTNER="192.168.1.100" ORIGIN_
FILENAME="data.txt" TARGET_PATHNAME="/tmp/" ACTION=ABORT
```

uxdup UPR : duplique une Uproc ainsi que l'ensemble de ses caractéristiques. Si cette Uproc a un script interne, le script est également dupliqué.

Item	Format	Défaut
EXP SIM INT APP	3c	S_ESPEXE
UPR=	64c	
VUPR=	3n	
TUPR=	64c	
TVUPR=	3n	
TLABEL	64c	

Exemple :

```
uxdup UPR EXP UPR=IU_TEST VUPR=000 TUPR=IU_TEST2 TVUPR=000 TLABEL="\Uproc
de test\"
```

uxshw UPR : affiche les caractéristiques d'une Uproc, son script (s'il est interne), ses variables (les valeurs des variables de type mot de passe sont affichées sous la forme d'étoiles *****), sa formule de lancement et le détail des conditions. Pour ne pas afficher le script (interne) le paramètre nœud U_UPR_SCRIPT_NODISP peut être défini. Reportez-vous au Manuel Utilisateur d'UniViewer.

Item	Format	Défaut
EXP SIM INT APP	3c	S_ESPEXE
UPR=	64c *	
VUPR=	3n *	
DOM=	1c *	
APPL=	64c *	
SCRIPT		
OUTPUT=file_name		

Exemple :

```
uxshw UPR EXP UPR=D_LOAD_FIL
```

uxdlt UPR : supprime une Uproc.

Item	Format	Défaut
EXP SIM INT APP	3c	S_ESPEXE
UPR=	64c *	

Item	Format	Défaut
VUPR=	3n *	

Exemple :

```
uxdlt UPR UPR=TEST*
```

uxlst UPR : liste les Uprocs.

Item	Format	Défaut
EXP SIM INT APP	3c	S_ESPEXE
UPR=	64c *	*
VUPR=	3n *	*
DOM=	1c *	*
APPL=	64c *	*
INF FULL		FULL
OUTPUT=file_name		

Exemple :

```
uxlst UPR OUTPUT="c:\temp\lstUpr.txt"
```

uxtra UPR : transfère les Uprocs dans un autre espace de la même société sur le même nœud.

Item	Format	Défaut
APP INT SIM	3c	S_ESPEXE
UPR=	64c *	
VUPR	3n *	
TESP=	I S X	

Exemple :

```
uxtra UPR APP UPR=D_* TESP=X
```

uxscv UPR : permet de définir la version courante d'une Uproc (espace application et intégration uniquement).

Item	Format	Défaut
INT APP	3c	S_ESPEXE
UPR=	64c *	
VUPR=	3n *	

Exemple :

```
uxscv UPR APP UPR=D_* VUPR=002
```

uxrls UPR : permet de libérer une Uproc.

Item	Format	Défaut
EXP SIM INT APP	3c	S_ESPEXE
UPR=	64c *	

Item	Format	Défaut
VUPR=	3n *	

Exemple :

```
uxrls UPR UPR=*
```

uxext UPR : extrait la définition complète d'une Uproc (avec son CL si celui-ci est déclaré interne).

Item	Format	Défaut
EXP SIM INT APP	3c	S_ESPEXE
UPR=	64c *	
VUPR=	3n *	
REPL		
ENABLE NOENABLE		ENABLE
OUTPUT=file_name		
NOVERIF		

Exemple :

```
uxext UPR EXP UPR=D_* OUTPUT="c:\temp\upr.ext"
```

uxins UPR : insère la définition complète d'une Uproc dans les fichiers correspondants de Dollar Universe (avec son éventuel CL) pour le même espace et la même version de Dollar Universe que l'extraction.

Item	Format	Défaut
EXP SIM INT APP	3c	S_ESPEXE
UPR=	64c *	
VUPR=	3n *	
REPL		
ENABLE NOENABLE		ENABLE
INPUT=file_name		

Exemple :

```
uxins UPR EXP UPR=* INPUT=c:\temp\upr.ext
```

Si des Uprocs ont été extraites des espaces exploitation ou simulation (version 000), elles peuvent être insérées dans les espaces intégration ou application en définissant au préalable la nouvelle valeur de la version (paramètre nœud S_VUPR_REINSERT, catégorie Réglages de l'interface commande):

```
S_VUPR_REINSERT=XYZ
```

XYZ étant la nouvelle valeur de la version de l'Uproc soit au minimum 001 pour pouvoir être insérée dans les espaces intégration ou application.

 Toutes les commandes d'insertion suivantes utiliseront ce paramètre pour définir la version des Uprocs insérées (quelque soit l'espace cible).

3.17.1 Description des items

UPR : Code Uproc. Format : 64 caractères alphanumériques. Requiert la saisie d'un code Uproc (valeur générique acceptée sauf pour uxadd).

TUPR : Code de l'Uproc résultat de la duplication. Format : 64 caractères alphanumériques.

VUPR : Version d'Uproc. Format : 3 caractères numériques. Permet la saisie d'une version d'Uproc (valeur explicite). Par défaut, pour la commande uxext, toutes les versions des Uprocs renseignées sont extraites (inutile avec SIM et EXP). Pour les autres commandes la version est obligatoire dans les espaces APP et INT, elle est facultative dans les espaces SIM et EXP (systématiquement à 000).

TVUPR : Version de l'Uproc résultat de la duplication. Format : 3 caractères numériques.

DOM : Code domaine. Format : 1 caractère alphanumérique. Permet la sélection des Uprocs rattachées à un domaine.

APPL : Code application. Format : 64 caractères alphanumériques. Permet la sélection des Uprocs rattachées à une application (valeur générique acceptée).

LABEL : Libellé descriptif de l'Uproc. Format : 64 caractères alphanumériques.

TLABEL : Libellé descriptif de l'Uproc résultat de la duplication. Format : 64 caractères alphanumériques.

FPERIOD : Période fonctionnelle. Indique le format de la date de traitement qui sera associée à l'exécution de l'Uproc. Les valeurs possibles sont : N : sans, D : jour, W : semaine, T : décade, F : quinzaine, M : mois, 2 : bimestre, 3 : trimestre, Q : quadrimestre, S : semestre, Y : année.

UPT : Type de l'Uproc. Pour une Uproc à CL interne indiquer CL_INT, pour une Uproc à CL externe sélectionner CL_EXT, pour une Uproc de type Commande indiquer CMD. Sélectionnez FTP_GET ou FTP_PUT pour les transferts de fichiers. D'autres types peuvent être disponibles selon les Managers Dollar Universe installés.

CLFIL : CL externe associé à l'Uproc. Ne peut être utilisé qu'avec le type d'Uproc CL_EXT. Format : 128 caractères maximum. Indique le chemin d'accès et le nom de la procédure associée à l'Uproc.

CL_INT : chemin ciblant un fichier unique dont le contenu sera copié en tant que script interne pour une Uproc de type CL_INT.

COMMAND : Ligne de commande à exécuter. Ne peut être utilisé qu'avec le type d'Uproc CMD. Format : 256 caractères maximum. La ligne de commande doit contenir :

- Le chemin d'accès à la commande (sinon, la commande est recherchée dans un des répertoires de la variable PATH).
- La commande et ses arguments.

Le chemin et les arguments de la commande peuvent contenir des variables selon la syntaxe : !VARIABLE!. Se reporter au Manuel de Référence de Dollar Universe pour la manipulation de ces variables.

EXECDIR : Répertoire d'exécution de la ligne de commande. Ne peut être utilisé qu'avec le type d'Uproc CMD. Par défaut ce répertoire est le répertoire home du compte de soumission qui exécute l'Uproc. Format : 258 caractères maximum.

SHELL : Contient le shell utilisé pour exécuter la commande de l'Uproc. Par exemple : "/bin/sh". Cet item n'est utilisable que pour les Uprocs de type CMD. Par exemple : SHELL=/usr/bin/ruby

SHELLARG : Contient les arguments optionnels de ce shell. Cet item n'est utilisable que pour les Uprocs de type CMD. Par exemple : SHELLARG="-e !CMD!"

Lorsque l'Uproc CMD s'exécute, la commande devient alors : "/usr/bin/ruby -e !CMD!"

(!CMD! étant la commande définie dans l'uproc).

Si la commande est saisie dans l'Uproc est : sleep(10), la ligne de commande sera :
/usr/bin/ruby -e sleep(10)

SHELLTYPE : Contient le type de shell pour une Uproc de type CL_INT, CL_EXT ou CMD uniquement sur un nœud en version 6.7 minimum. La valeur par défaut est DEFAULT : le type de shell est indiqué si nécessaire dans le script. Les autres valeurs sont : POWERSHELL, CSCRIPT (sauf type CMD) ou PERL.

DOM : Code domaine. Format : 1 caractère alphanumérique. Indique à quel domaine l'Uproc est rattachée.

APPL : Code application. Format : 64 caractères alphanumériques. Indique à quelle application l'Uproc est rattachée.

CLASS : Classe d'appartenance de l'Uproc. Format : 64 caractères alphanumériques. Indique à quelle classe l'Uproc est associée.

MEMO : Indique le mode de mémorisation de l'Uproc dans le fichier des événements jobs. La valeur par défaut est : N. Trois valeurs sont possibles :

- N : sans mémorisation,
- O : une exécution,
- A : toutes les exécutions.

Dans le cas où l'Uproc est mémorisée (choix O ou A), cette information doit être complétée par l'item NBPER.

NBPER : Nombre de périodes de mémorisation de l'Uproc. Format : 2 caractères numériques. La valeur par défaut est : 0 (sans limitation). Indique le nombre de périodes (fonctionnelles) sur lesquelles les événements seront conservés dans la base des événements.

RETRY : Redémarrage automatique. Si le mot clé est utilisé: si l'état de terminaison déterminé par le lanceur est Incidenté, le lanceur réitérera l'exécution de l'Uproc jusqu'à obtenir un état Terminé. Format: (NNN,MM):

- NNN: nombre maximum de tentatives d'exécution pour obtenir un état final Terminé. Maximum 999.
- MM: intervalle en minutes entre chaque tentative. Entre la fin de l'exécution précédente et le redémarrage de la nouvelle exécution. La valeur "0" signifie un redémarrage immédiat. Maximum 59 minutes.

DEFINFO : Valeur par défaut du paramètre Information de l'Uproc qui sera affiché dans le suivi des exécutions. Texte libre tronqué à 50 caractères s'il est plus long. Optionnel, vide par défaut.

DEFSEV : Valeur par défaut du paramètre Sévérité de l'Uproc qui sera affiché dans le suivi des exécutions. Nombre entier de 0000 à 9999 sur 4 caractères. Optionnel, vide par défaut.

INTERACTIVE : sous Windows uniquement, le Job est soumis en mode interactif par le biais du service "Interactive Desktop".

NOINTERACTIVE : le job est soumis en mode batch, par le biais du Service Utilisateur du compte de soumission. Si aucun service utilisateur n'est disponible, le système tente de soumettre le Job en mode interactif par le biais du service "Interactive Desktop".

DIR_FILES : chemin ciblant un répertoire dont le contenu sera copié en tant que fichiers de l'Uproc.

ENABLE : Active ou désactive la gestion de l'état de terminaison de l'Uproc. Ne peut être utilisé qu'avec la famille STATUS. Obligatoire. Format : 1 caractère : Y, y, N, n.

- Y ou y : active la gestion de l'état de terminaison
- N ou n : désactive la gestion de l'état de terminaison

SET : Nouvel état de terminaison de l'Uproc si la règle définie est vérifiée. Ne peut être utilisé qu'avec la famille STATUS. Format : 1 caractère : C (état Terminé, valeur par défaut) ou A (état Incidenté).

RETCODE : Test sur le code retour du script de l'Uproc (sous UNIX/OpenVMS) ou sur la variable RESEX (sous Windows). Ne peut être utilisé qu'avec la famille STATUS. Incompatible avec EXT_RETCODE, STRING et SEARCH_FILE. Obligatoire. Format 4n[,2c].

- Valeur : Code retour attendu. Nombre entier positif ou négatif. Obligatoire.
- Direction : Opérateur de test de la valeur. Optionnel, la valeur par défaut est EQ (égal). Les valeurs possible sont : EQ (égal), GT (supérieur à), LO (inférieur à) ou NE (différent de).

EXT_RETCODE : Test sur le code retour du script de l'Uproc (sous UNIX/OpenVMS) ou sur la variable RESEX (sous Windows). Ne peut être utilisé qu'avec la famille STATUS. Incompatible avec RETCODE, STRING et SEARCH_FILE. Obligatoire. Format : n1;n2;[n3,n4];[n5,n6]/n7 sur 64 caractères maximum. La liste des valeurs peut être composée:

- D'une valeur simple : format : * ou nombre entier positif ou négatif (de -32767 à 32768), par exemple: -2
- D'une plage de valeurs entières positives ou négatives (de -32767 à 32768) : format : [min,max], par exemple: [-10,10] : de -10 à 10 inclus.
- De valeurs calculées à partir d'un multiple. Format : N/m (N positif ou négatif, m positif uniquement), par exemple: */3 : tous les multiples de 3.
- De valeurs calculées dans une plage (de -32767 à 32768) selon le multiple spécifié (m positif uniquement) : format [min,max]/m, par exemple: [1,10]/4 : 4 et 8. Chacune des valeurs ou des plages doit être séparée par un point virgule. La liste ne doit pas dépasser 64 caractères. Par exemple: -2;[-10,10];*/3;[1,10]/4

STRING : Test sur la présence d'une chaîne de caractères dans le fichier spécifié par l'item SEARCH_FILE. Ne peut être utilisé qu'avec la famille STATUS. Incompatible avec RETCODE et EXT_RETCODE. Obligatoire. Format : 128 caractères. Si la chaîne contient des espaces, il faut l'entourer de guillemets. Utilisé avec le mot clé REGEXP, ce champ correspond à une expression régulière (bibliothèque Tcl/Tk version 8.1 avec l'option "ERE" signifiant que seule la syntaxe 8.0 est supportée; suivez cette URL https://tcl.tk/doc/howto/regexp81.html#part1_2 pour plus de détails).

Les caractères suivants sont supportés pour les expressions régulières :

- Générique : pour remplacer n'importe quel caractère "."
- Répétition : zéro ou plusieurs occurrences du caractère ou de la classe précédant "*"
- Position dans la ligne : début de ligne "^"
- Position dans la ligne: fin de ligne "\$"
- Classe de caractères: chaîne qui contient l'un des caractères. Exemple : [clas]
- Classe inversée : chaîne qui ne contient pas l'un des caractères. Exemple : [^clas]
- Plage de caractères : tous les caractères dans la plage définie. Exemple : [0-9]
- Échappement : pour rechercher le caractère spécial "x". Exemple : \x
- Position dans le mot : début du mot. Exemple : \<xyz
- Position dans le mot: fin du mot. Exemple : xyz\>

REGEXP | **NOREGEXP** : indique que STRING contient une expression régulière. Valeur par défaut NOREGEXP.

SEARCH_FILE : Fichier dans lequel la chaîne de caractères STRING sera recherchée. Ne peut être utilisé qu'avec la famille STATUS. Incompatible avec RETCODE et EXT_RETCODE. Optionnel, par défaut le fichier log de l'Uproc est utilisé. Format : 256 caractères. Si la chaîne contient des espaces, il faut l'entourer de guillemets.

MAXDUR : contrôler une durée d'exécution trop longue. Format : (HHHMMSS|nnn,K | F | C=256c) ou NONE pour effacer la valeur. Si la durée indiquée (HHHMMSS ou nnn%) est atteinte avant la terminaison de l'Uproc, l'action choisie sera exécutée :

- K: arrête l'Uproc
- F: force le statut à Terminé
- C: exécute le script indiqué : renseignez le chemin et le nom du script à exécuter (maximum 256 caractères)

MINDUR : contrôler une durée d'exécution trop courte. Format: (HHHMMSS|nnn,R | C=256c) ou NONE pour effacer la valeur. Si l'Uproc se termine avant la durée indiquée (HHHMMSS ou nnn%), l'action choisie sera exécutée :

- R: reprise de l'Uproc
- C: exécute le script indiqué : renseignez le chemin et le nom du script à exécuter (maximum 256 caractères)

MAXWDUR : contrôler une durée d'attente trop longue avant l'exécution. Format : (HHHMMSS|nnn,F | A | C=256c) ou NONE pour effacer la valeur. Si le temps d'attente (HHHMMSS ou nnn%) avant l'exécution est dépassé, l'action choisie sera exécutée.

- F: force le statut à Terminé
- A: force le statut à Incidenté
- C: exécute le script indiqué : renseignez le chemin et le nom du script à exécuter (maximum 256 caractères)

REPL : Option de remplacement. Indique si l'enregistrement extrait (inséré) annule et remplace l'enregistrement déjà existant dans le fichier de destination; par défaut, l'extraction (l'insertion) est refusée pour chaque enregistrement déjà existant (signalé par un message pour chaque occurrence).

ENABLE : Option de libération. Indique si l'Uproc sera insérée dans l'état libéré. L'action de libération sera nécessaire avant toute modification d'Uproc si l'objet est inséré avec le qualifieur NOENABLE. Défaut : ENABLE.

MAIL : Active la notification e-mail pour cette Uproc. Format : Y ou N.

- Y (yes) : active la notification par e-mail lorsque l'un des états sélectionnés (ci-dessous) est atteint.
- N (no) : désactive la notification par e-mail. Valeur par défaut.

COMPLETED : Format : Y ou N (par défaut). Indique si l'état Terminé doit déclencher une notification par e-mail (valeur Y : yes).

ABORTED : Format : Y ou N (par défaut). Indique si l'état Incidenté doit déclencher une notification par e-mail (valeur Y : yes).

REFUSED : Format : Y ou N (par défaut). Indique si l'état Refusé doit déclencher une notification par e-mail (valeur Y : yes).

TIMEOVERRUN : Format : Y ou N (par défaut). Indique si l'état "Horaire dépassé" doit déclencher une notification par e-mail (valeur Y : yes).

JOBLOG : Indique si le log de l'Uproc doit être attaché à l'e-mail. Format :

- N : valeur par défaut, le log de l'Uproc n'est pas attaché à l'e-mail envoyé.
- Y : le log de l'Uproc est attaché à l'e-mail envoyé quelle que soit sa taille.
- 4 caractères numériques : le log de l'Uproc n'est attaché à l'e-mail que si la taille ne dépasse pas la valeur indiquée en kilo octets.

RECIPIENT : Liste des adresses e-mail des destinataires de la notification. Sur Unix, la liste de plus d'une adresse doit être encadrée de guillemets " " (optionel sous Windows). Format : 256 caractères maximum : "email1;email2;..."

VARNAME : Nom de la variable. Format : 20 caractères alphanumériques. Requiert la saisie du nom de la variable de l'Uproc, saisie obligatoire.

VTYPE : Type de la variable. Format : Q pour quantité, T pour texte, P pour mot de passe ou D pour date (choix exclusif). La variable peut être de type texte, mot de passe, quantité ou date. Du type dépend le contrôle du format de la variable. Saisie obligatoire.

VFORMAT : Format de la variable. Saisie obligatoire sauf pour une variable de type quantité pour laquelle cette information ne présente pas de sens.

- Pour une variable de type Date, le format doit être choisi entre les trois valeurs suivantes : AAAA/MM/JJ ou JJ/MM/AAAA ou MM/JJ/AAAA.
- Pour une variable de type texte ou mot de passe, le format exprime la longueur maximum du texte saisi (guillemets non compris) : au maximum et par défaut 255 caractères.

VALMIN, VALMAX : Plage pour une variable de type Quantité uniquement. Saisie non obligatoire. Format : 12 caractères numériques. Valeurs par défaut : 0 et 999999999999. VALMIN et VALMAX doivent indiquer dans quelle plage la valeur de la variable devra être saisie. VALMIN indique la valeur inférieure, VALMAX la valeur supérieure de cette plage.

VVALUE : Valeur de la variable. Saisie obligatoire. La valeur d'une variable mot de passe doit être saisie en clair mais elle ne sera pas affichée dans la sortie de la commande. La valeur de la variable doit être compatible avec son type, son format et éventuellement sa plage, dans le cas contraire la commande sera refusée. Si la valeur contient des caractères non alphanumériques (espaces, /...) elle doit être encadrée de guillemets.

XCLASS : Classe d'incompatibilité de l'Uproc. Indique que l'Uproc est incompatible avec une Uproc de cette classe. Format : entre parenthèses, les différentes classes d'incompatibilité sont séparées par des virgules, les espaces sont interdits. Par exemple : XCLASS=(CLASSE1,CLASSE2) : l'Uproc est déclarée incompatible avec les classes CLASSE1 et CLASSE2.

TERMNO : Numéro désignant la consigne de terminaison. Format : 2 caractères numériques. Un numéro est associé à chaque consigne, deux consignes différentes ne peuvent pas porter le même numéro.

TERM : Consigne de terminaison. Format : 64 caractères alphanumériques. Indique le code de l'Uproc correspondant à l'événement qui doit être purgé par l'exécution de la consigne de terminaison.

SUCC_UPR : Successeur de l'Uproc. Format: 64 caractères. Spécifie l'Uproc successeur de l'Uproc courante. Les Uprocs successeurs sont examinées dans l'ordre dans lequel elles sont ajoutées et ne peuvent être ajoutées qu'une par une. Exemple: uxupd UPR EXP UPR=UPROC1 SUCCESSOR add UPROC1_SUCCESSOR_1.

CONDNO : Numéro désignant la condition. Format : 2 caractères numériques. Un numéro est associé à chaque condition (quel que soit le type), deux conditions différentes ne peuvent pas porter le même numéro.

DEP : Condition d'enchaînement. Format : 64 caractères alphanumériques. Indique le code de l'Uproc correspondant à la condition d'enchaînement.

INC : Condition de non simultanété. Format : 64 caractères alphanumériques. Indique le code de l'Uproc correspondant à la condition de non simultanété.

RES : Condition de ressource. Format : 64 caractères alphanumériques. Indique la référence de la ressource correspondant à la condition de ressource.

IFPLANNED : Indicateur pour que la condition ne soit examinée par le lanceur que si l'Uproc conditionnante est planifiée / exécutée, en tenant compte des éléments : compte de soumission / session / UG et date de traitement. Si l'événement n'est pas présent et si le lancement n'est pas planifié, la condition est complètement ignorée (évite d'attendre un traitement qui n'aura pas lieu). Reportez-vous au Manuel de Référence pour la description complète.

PDATE : Contrôle sur la date de traitement. Indique le décalage recherché entre la date de traitement de l'Uproc conditionnée et de l'Uproc ou de la ressource conditionnante. Format : il dépend de la période fonctionnelle de l'Uproc conditionnante, il s'exprime en "+" ou "-" un nombre d'unités calendaire (cf. ci-dessous) entre parenthèses, séparés par des virgules.

- Si une des unités calendaire contrôlée doit être la même, aucune valeur n'est requise. Par exemple PDATE=(+1,-1,) pour une période fonctionnelle jour peut être traduit par : jour +1, mois -1, même année.
- Si le contrôle doit porter sur la même date de traitement (globalement), le mot clé SAME peut être utilisé, si le contrôle doit porter sur la même unité calendaire un caractère blanc peut être utilisé dans le triplet (cf. Exemple ci-dessus),
- Si le contrôle ne tient pas compte de la date de traitement le mot clé ANY peut être utilisé, si le contrôle ne doit pas tenir compte d'une unité calendaire particulière la lettre a peut être utilisée dans le triplet, par exemple PDATE=(a, ,). peut être traduit par : quel que soit le jour pour le même mois et la même année.

Le tableau suivant résume les unités calendaire proposées en fonction de la période fonctionnelle de l'Uproc :

	Unités calendaires									
Période fonctionnelle	Jour	Sem.	Décad.	Quinz.	mois	2 M.	3 M.	4 M.	6 M.	12 M.
Sans										
Jour	*				*					*
Semaine		*								*
Décade			*		*					*
Quinzaine				*	*					*
Mois					*					*
Bimestre						*				*
Trimestre							*			*
Quadrimestre								*		*
Semestre									*	*
Année										*

DTYPE : Contrôle sur la date de traitement. Indique le type de jour (indiqué par PDATE) à considérer pour les Uprocs à période fonctionnelle jour. Format:

- A: (any) quelque que soit le type de jour : ouvré, chômé, férié : le nombre indiqué par PDATE est calculé en jours calendaires,
- W: (worked) le nombre indiqué par PDATE est calculé en jours ouvrés (en relation avec le calendrier de référence de l'unité de gestion considérée),
- C: (closed) le nombre indiqué par PDATE est calculé en jours chômés,
- H: (holiday) le nombre indiqué par PDATE est calculé en jours fériés.

MU : Contrôle sur les unités de gestion. Indique sur quel ensemble d'unités de gestion la condition doit être recherchée.

- Si le contrôle doit porter sur la même unité de gestion, le mot clé SAME peut être utilisé.
- Si le contrôle doit porter sur une unité de gestion spécifique, l'expression du contrôle sera mu(MU_CODE), où MU_CODE représente le code de l'unité de gestion visée (64 caractères alphanumériques).
- Si le contrôle doit porter sur un type d'unité de gestion, l'expression du contrôle sera mut(T), où T représente le type d'unité de gestion (1 caractère alphanumérique).
- Si le contrôle doit porter sur des dépendances d'unité de gestion, l'expression du contrôle sera hdp({XY}), où X représente le type des unités de gestion visées (1 caractère alphanumérique) et Y le type d'unité de gestion origine de la dépendance (1 caractère alphanumérique).

ONE : Unités de gestion visées. La valeur par défaut est ALL. Utilisé uniquement pour les conditions d'enchaînement ou de ressource, dans le cas où le contrôle des unités de gestion porte sur un type d'unité de gestion ou un TIH.

- ONE signifie que la condition est attendue sur au moins une des unités de gestion visées.
- ALL signifie que la condition est attendue sur toutes les unités de gestion visées.

SESS : Contrôle sur la session. Indique sur quelle session la condition doit être recherchée.

- Si le contrôle ne prend pas en compte la notion de session, le mot clé any peut être utilisé.
- Si le contrôle porte sur la même session, le mot clé same peut être utilisé.
- Si le contrôle porte sur la même session et la même exécution, le mot clé samerun peut être utilisé.
- Si le contrôle porte sur une session spécifique, l'expression du contrôle sera expl(COD_SESS), où COD_SESS est le code de la session visée (64 caractères alphanumériques).

STATUS : Etat recherché. Indique quel est état recherché de l'Uproc. Pour les conditions d'enchaînement ou consignes de terminaison :

- Si l'état recherché est "TERMINE" le mot clé TERMINATED peut être utilisé (valeur par défaut).
- Si l'état recherché est "INCIDENTE" le mot clé ABNORMAL peut être utilisé.

Pour les conditions d'enchaînement uniquement :

- Si l'état recherché est "TERMINE" ou "INCIDENTE" le mot clé ENDED peut être utilisé.
- Si l'état recherché est "ABSENT" le mot clé ABSENT peut être utilisé.

USER : Compte de soumission. Indique avec quel type de contrôle sur le compte de soumission la condition doit être recherchée.

- Si le contrôle porte sur le même compte utilisateur le mot clé SAME peut être utilisé.
- Si le contrôle ne prend pas en compte la notion de compte de soumission le mot clé ANY peut être utilisé (valeur par défaut).

EXPECTED : logique booléenne. Indique que la condition (ou la consigne) exprimée est recherchée vraie (valeur par défaut). Si la condition ou la consigne recherchée doit être le contraire de ce qui est exprimé, le mot clé EXCLUDED peut être utilisé.

NOFATAL : comportement du lanceur vis à vis de l'absence de la condition.

- Si la condition exprimée n'est pas vérifiée lors du premier pilotage de l'Uproc, le Lanceur placera le lancement en attente d'événement. L'Uproc sera à nouveau pilotée lors de l'arrivée de l'événement attendu (valeur par défaut).
- Si la condition exprimée n'est pas vérifiée lors du premier pilotage de l'Uproc et que le mot clé FATAL est utilisé, le pilotage sera abandonné par le Lanceur et le lancement sera placé dans l'état "refusé au pilotage".

VERIF : vérification du fichier. Indique si l'automate doit vérifier la présence physique de la ressource lors du pilotage de l'Uproc portant la condition (valeur par défaut). Si l'automate ne doit pas vérifier la présence

physique de la ressource (elle ne sera vue alors que comme une ressource logique) le mot clé NOVERIF peut être utilisé.

UNLOCK : allocation de la ressource. Indique que l'Uproc conditionnée ne s'attribue que les quotas définis dans la condition (valeur par défaut). D'autres Uprocs conditionnées sur cette ressource pourront s'exécuter dans la mesure des quotas disponibles.

- Si le mot clé LOCK est utilisé, l'Uproc conditionnée s'attribue la totalité de la ressource durant son exécution et aucune autre Uproc, conditionnée par cette ressource ne pourra s'exécuter.
- Si la condition de ressource est déclarée UNLOCK mais que la ressource a été définie "LOCK" le comportement sera identique.

ATTRIB : attribut à vérifier. Permet la saisie de l'attribut de la nature de la ressource (les différents attributs sont décrits dans le Manuel de Référence de Dollar Universe, paragraphe "condition de ressource").

OPER : opérateur. Utilisé pour les ressources de nature FIL.

VALUE : valeur. Utilisé pour les ressources de nature FIL.

QT1 et **QT2** : quotas nécessaires. Valeur numérique comprise entre 0 et 9999, 0 par défaut. Dans le cas d'une condition et d'une ressource déclarées UNLOCK, le Lanceur examine, au moment du pilotage, si les quotas exprimés dans la condition sont effectivement disponibles. S'ils le sont, l'Uproc sera soumise à l'exécution et bloquera les quotas correspondants. Plusieurs Uprocs peuvent ainsi se partager une même ressource.

LFORM : formule de lancement. Exprime la totalité de la formule de lancement pour l'Uproc visée et l'action en cours. Sa longueur maximum doit être de 400 caractères.

- Cnn : indique la condition numéro nn,
- = : indique que la condition est recherchée vraie,
- # : indique que la condition est recherchée fausse,
- AND ou ET : indique que les deux conditions doivent être vérifiées,
- OR ou OU : indique que l'une ou l'autre des deux conditions doivent être vérifiées.

"AND" a priorité sur "OR", des parenthèses peuvent être utilisées pour définir des groupes de conditions réunies par des "OR". Par exemple : LFORM="((=C01 AND (=C02 OR #C03)))"

INF : Option d'affichage. Limite l'affichage au nom des objets. Autre valeur : FULL (par défaut). Affiche toutes les informations disponibles sur la liste des objets.

 INF et FULL sont incompatibles entre eux.

OUTPUT : Stockage des résultats. Le nom du fichier est obligatoire pour la commande uxext. Si l'item n'est pas renseigné la sortie se fait à l'écran.

INPUT : fichier d'entrée pour la commande uxins. Le nom du fichier est obligatoire.

TESP : code de l'espace cible du transfert : X pour Exploitation, S pour Simulation ou I pour intégration.

NOVERIF : peut être utilisé avec la commande uxext uniquement. Cette option permet de ne pas vérifier le fichier l'extraction avant d'y écrire de nouveaux enregistrements et d'améliorer ainsi les performances de cette commande. Cet item ne doit être utilisé que pour un nouveau fichier.

SCRIPT : peut être utilisé avec la commande uxshw upr uniquement. Cette option permet d'afficher en sortie le script de l'Uproc si le paramètre nœud U_UPR_SCRIPT_NODISP (catégorie Réglages avancés) a été défini à "Oui". Reportez-vous au Manuel Utilisateur d'UniViewer.

 **Rappel** : par défaut cette variable n'est pas définie et le script de l'Uproc est affiché même si l'item SCRIPT est absent.

PARTNER : Nom physique de la machine cible ou adresse IP.

ORIGIN_FILENAME : Nom du fichier à transférer. Par défaut tous les fichiers du répertoire

ORIGIN_PATHNAME : Nom du répertoire contenant les fichiers à transférer. Par défaut, répertoire de connexion de l'utilisateur FTP. Caractère * interdit.

TARGET_FILENAME : Nom du fichier cible. Inchangé par défaut.

TARGET_PATHNAME : Nom du répertoire de réception des fichiers. Par défaut, répertoire de connexion du compte de soumission de l'Uproc. Caractère * interdit.

LOGON : Utilisateur de connexion FTP.

PASSWORD : Mot de passe de l'utilisateur de connexion FTP. Le mot de passe est invisible dans l'interface graphique.

MODE : Type de transfert : ASCII ou BINARY

ACTION : Comportement si le fichier cible existe déjà : REPLACE (remplacer), ABORT (annuler erreur), APPEND (concaténer, FTP_PUT uniquement) ou NONE (ne rien faire).

CRLF : Tous les caractères CR sont convertis en LF durant le transfert ASCII. Par défaut pas de conversion sauf pour les soumissions. L'option CRLF n'est pas disponible pour FTP_PUT.

DELETE : Supprime le fichier origine si le transfert est réussi. Par défaut pas de suppression.

NEWER : Transfère le fichier si sa date de modification est plus récente que celle du fichier cible.

RECURSIVE : Transfère tous les fichiers et sous-répertoires contenus dans le chemin ORIGIN_PATHNAME (les paramètres ORIGIN_FILENAME et TARGET_FILENAME ne doivent plus être saisis). Si cette option est utilisée, le nombre de niveaux de répertoires ne doit pas dépasser 150. NORECURSIVE par défaut.

 L'option RECURSIVE utilise en interne la commande FTP LIST pour obtenir la liste des fichiers et des répertoires du serveur distant. La réponse attendue du serveur doit être formatée de telle sorte que chaque ligne doit commencer par les droits des fichiers au format UNIX. Cette réponse variant d'un serveur à un autre, la fonction ne pourra pas être utilisée si la réponse a un format différent. Dans ce cas, un fonctionnement dégradé peut être utilisé : chaque fichier du serveur distant est testé à l'aide d'une commande FTP CWD pour vérifier l'existence d'un répertoire. Ce fonctionnement dégradé est moins performant mais il peut être activé à l'aide du paramètre noeud U_FT_GET_DIR_FAILOVER dans la catégorie "Paramètres noeud – Paramétrage FTP".

BUFFER_SIZE : Définit la taille du buffer de communication en blocs de 512 octets. Si la valeur est définie, le format est un nombre entier, valeur minimum 1, valeur maximum 64.

UMASK : Masque appliqué aux permissions du fichier local. Format : 3 caractères numériques.

- Uprocs FTP_GET : Si UMASK n'est pas défini ou est défini à 000, les fichiers sont téléchargés par défaut avec les permissions 666 (pour un utilisateur root) ou 640 (pour un utilisateur non-root). Sinon, les permissions du fichier sont obtenues en soustrayant UMASK de la valeur par défaut (666 ou 640).
 - Uprocs FTP_PUT : UMASK n'est pas utilisable. Sur systèmes UNIX, les fichiers sont téléchargés avec leurs permissions d'origine. Sous Windows, les fichiers sont téléchargés avec les permissions 755 (les permissions octal n'existant pas nativement sous Windows).
-

3.17.2 Exemples

```
uxadd UPR EXP UPR=D_LOAD_FIL LABEL=\"Chargement du fichier\" FPERIOD=d
UPT=CL_EXT CLFIL=c:\temp\D_LOAD_FIL.bat APPL=\"BK\" MEM0=\"0\" NBPER=1
```

Crée l'Uproc D_LOAD_FIL version 000 label "Chargement du fichier" dans l'environnement courant (société, nœud) en espace exploitation, période fonctionnelle : jour, fichier de commande : c:\temp\D_LOAD_FIL.bat, application BK (domaine T), mémorisée 1 exécution 1 période.

```
uxadd UPR EXP UPR=DEF_UPR APPL=U_ UPT=CL_EXT CLFIL=C:\script.bat
DEFINFO=\"Default information for this Uproc\" DEFSEV=0001
```

Crée l'Uproc DEF_UPR de type CL_EXT (script : "C:\script.bat"), avec les valeurs par défaut des paramètres Information à "Default information for this Uproc" et Sévérité à 1.

```
uxupd UPR EXP UPR=D_LOAD_FIL RESCON add CONDNO=01 RES=D_FIL_PAY
uxupd UPR EXP UPR=D_LOAD_FIL RESCON add CONDNO=02 RES=D_BASE LOCK
uxupd UPR EXP UPR=D_LOAD_FIL FORMULA LFORM=\"=C01 ET =C02\"
```

Modifie l'Uproc D_LOAD_FIL pour lui définir une formule de lancement composée de deux conditions de ressources en ET, la 1^{ère} sur la présence de la ressource fichier D_FIL_PAY et la 2^{nde} sur la disponibilité totale de la ressource logique D_BASE.

```
uxadd UPR EXP UPR=D_BACKUP LABEL=Sauvegarde FPERIOD=d UPT=CL_EXT
CLFIL=c:\temp\D_BACKUP.bat appl=BK
```

Crée l'Uproc D_BACKUP en espace exploitation label "Sauvegarde" dans l'environnement courant, période fonctionnelle : jour, fichier de commande : c:\temp\D_BACKUP.bat, application BK (domaine T).

```
uxupd UPR EXP UPR=D_BACKUP DEPCON add CONDNO=01 DEP=D_LOAD_FIL
uxupd UPR EXP UPR=D_BACKUP FORMULA LFORM=\"=C01\"
```

Modifie l'Uproc D_BACKUP pour lui définir une formule de lancement composée d'une condition d'enchaînement sur l'Uproc D_LOAD_FIL (dans l'état Terminé par défaut). Cette Uproc libérera dans son script la ressource D_BASE allouée en totalité par l'Uproc D_LOAD_FIL.

```
uxupd UPR UPR=D_LOAD_FIL LAUNCHVAR add VARNAME=VARTXT VTYPE=T VVALUE=\"XY_
VARTXT\" VFORMAT=\"30\"
```

Crée pour l'Uproc D_LOAD_FIL une variable VARTXT de type texte, d'une longueur maximum de 30 caractères, de valeur par défaut "XY_VARTXT".

```
uxupd UPR UPR=D_LOAD_FIL LAUNCHVAR add VARNAME=VARDATE VTYPE=D
VVALUE=\"1999/12/31\" VFORMAT=\"AAAA/MM/JJ\"
```

Crée pour l'Uproc D_LOAD_FIL une variable de type date au format année/mois/jour, de valeur par défaut 1999/12/31.

```
uxext UPR EXP UPR=D* output=c:\temp\DOC.ext
```

Extrait toutes les Uprocs commençant par D_ de l'espace exploitation et archive les données dans le fichier c:\temp\DOC.ext.

```
uxins UPR EXP UPR=* input=c:\temp\DOC.ext
```

Insère toutes les Uprocs du fichier c:\temp\DOC.ext dans l'environnement courant (société, nœud) en espace exploitation sauf si elles existent déjà.

3.18 Les sessions

uxadd SES : crée une session dans la liste des sessions.

Item	Format	Défaut
EXP SIM INT APP	3c	S_ESPEXE
SES=	64c	
VSES=	3n	
LABEL=	64c	" "
HEADER=	64c	
FATHER=	Uproc_code(nnn)	
SONOK=	((Uproc_code(nnn),{XY}) (Uproc_code(nnn),{XY}) (Uproc_code(nnn),ug_code)...))	
SONRC	((Uproc_code(nnn),ug_code,returnCode) (Uproc_code(nnn),{XY},returnCode) (Uproc_code(nnn),{XY},returnCode) ...)	
SONKO=	((Uproc_code(nnn),{XY}) (Uproc_code(nnn),{XY}) (Uproc_code(nnn),ug_code)...))	

Exemples :

```
uxadd SES EXP SES=SES_TEST LABEL="\Create session\" HEADER=IU_TEST
FATHER=IU_TEST SONOK="((IU_TEST2)(IU_TEST3(001)))" SONKO="((IU_ALARM
(002)))" FATHER=IU_TEST2 SONOK="((IU_TEST3(003),{A })))" FATHER=IU_TEST3\
(003\ SONOK="((D_BACKUP))" SONKO="((IU_ALARM(004)))"
uxadd SES EXP SES=D_LOAD_BCK label="\Load file\" HEADER=D_LOAD_FIL
FATHER=D_LOAD_FIL SONOK="((D_BACKUP))"
uxadd ses ses=MYSESSION LABEL="\Create session\" HEADER=HEADER
FATHER=HEADER SONOK=((MYSONOK)) SONRC=((MYSONCR4,,4) (MYSONANY))"
```

uxdup SES : duplique une session.

Item	Format	Défaut
EXP SIM INT APP	3c	S_ESPEXE
SES=	64c	
VSES=	3n	
TSES=	64c	
TVSES=	3n	
TLABEL	64c	

Exemple :

```
uxdup SES EXP SES=TEST VSES=000 TSES=TEST2 TVSES=000 TLABEL="\Session de
test\"
```

uxshw SES : affiche les caractéristiques d'une session.

Item	Format	Défaut
EXP SIM INT APP	3c	S_ESPEXE
SES=	64c *	
VSES=	3n *	

Item	Format	Défaut
LST LNK		LST
OUTPUT=file_name		

Exemple :

```
uxshw SES EXP SES=SES_TEST LNK
```

uxdlit SES : supprime une session.

Item	Format	Défaut
EXP SIM INT APP	3c	S_ESPEXE
SES=	64c *	
VSES=	3n *	

Item	Format	Défaut
REPL		
OUTPUT=file_name		
NOVERIF		

Exemple :

```
uxext SES EXP SES=D_* output=c:\temp\ses.ext
```

uxins SES : insère la définition complète d'une session dans les fichiers correspondants de Dollar Universe pour le même espace et la même version de Dollar Universe que l'extraction.

Item	Format	Défaut
EXP SIM INT APP	3c	S_ESPEXE
SES=	64c *	
VSES=	3n *	
REPL		
ENABLE NOENABLE		ENABLE
INPUT=file_name		

Exemple :

```
uxins SES EXP SES=* INPUT=c:\temp\ses.ext
```

Si des sessions ont été extraites de l'espace exploitation ou simulation (version 000), elles peuvent être insérées dans les espaces intégration ou application en définissant au préalable la nouvelle valeur de la version (paramètre nœud S_VSES_REINSERT, catégorie Réglages de l'interface commande) :

```
S_VSES_REINSERT=XYZ
```

XYZ étant la nouvelle valeur de la version de la session soit au minimum 001 pour pouvoir être insérée dans les espaces intégration ou application.

 Toutes les commandes d'insertion suivantes utiliseront ce paramètre pour définir la version des sessions insérées (quelque soit l'espace cible).

3.18.1 Description des items

SES : code session. Format : 64 caractères alphanumériques. Requiert la saisie d'un code session (valeur générique acceptée sauf pour uxadd).

TSES : code de la session résultat de la duplication. Format : 64 caractères alphanumériques.

VSES : version de session. Format : 3 caractères numériques. Permet la saisie d'une version de session (valeur explicite). Par défaut, toutes les versions des sessions renseignées sont extraites (inutile avec EXP ou SIM).

TVSES : version de la session résultat de la duplication. Format : 3 caractères numériques.

LABEL : libelle descriptif de la session. Format : 64 caractères alphanumériques.

TLABEL : libelle descriptif de la session résultat de la duplication. Format : 64 caractères alphanumériques.

HEADER : Uproc entête de la session. Code de l'Uproc en tête de la session.

FATHER : description de l'origine de la branche décrite. Code de l'Uproc parent de la branche.

 Ce code peut être suivi du numéro d'occurrence de l'Uproc au sein de la session (3 caractères numériques, ce qui permet de distinguer deux citations de la même Uproc dans la session) selon la syntaxe : CodeUproc(nnn).

SONOK : Description de l'Uproc suivante en chemin normal. Code de l'Uproc enfant par rapport au FATHER précédent.

 Même remarque que précédemment pour le numéro d'occurrence : cela permet de distinguer plusieurs citations de la même Uproc.

Plusieurs Uprocs peuvent être lancées en parallèle, les codes sont alors présentés entre parenthèses, l'ensemble du chemin est lui-même également décrit entre parenthèses, soit par exemple : ((codeUproc(nnn),{XY}) (codeUproc(nnn),CodeUg) ...).

- codeUproc est le code de l'Uproc à lancer
- nnn est le numéro d'occurrence de l'Uproc dans la session (facultatif)
- {XY} indique l'expression d'un TIH (facultatif) ou CodeUg est le code d'une unité de gestion explicite (facultatif), même UG par défaut.

SONKO : Description de l'Uproc suivante en chemin erreur. Code de l'Uproc enfant par rapport au FATHER précédent, mêmes remarques que précédemment.

SONRC : Description de l'Uproc suivante en chemin code retour. Code de l'Uproc enfant par rapport au FATHER précédent.

 Même remarque que précédemment pour le numéro d'occurrence : cela permet de distinguer plusieurs citations de la même Uproc.

Plusieurs Uprocs peuvent être lancées en parallèle, les codes sont alors présentés entre parenthèses, l'ensemble du chemin est lui-même également décrit entre parenthèses, soit par exemple : ((CodeUproc1(nnn),{XY},CodeRetour1) (CodeUproc2(nnn),CodeUg,CodeRetour2) ...).

- CodeUproc est le code de l'Uproc à lancer.
- nnn est le numéro d'occurrence de l'Uproc dans la session (facultatif).
- {XY} indique l'expression d'un TIH (facultatif) ou CodeUg est le code d'une unité de gestion explicite (facultatif), même UG par défaut.
- CodeRetour : code retour de l'Uproc parent qui doit lancer l'Uproc fille. Vous pouvez saisir une expression du même type que celle utilisée pour déterminer le code retour étendu [EXT_RETCODE](#) de l'Uproc. * par défaut. Format : n1;n2;[n3,n4];[n5,n6]/n7 sur 64 caractères maximum. La liste des valeurs peut être composée:

- D'une valeur simple : format : * ou nombre entier positif ou négatif (de -32767 à 32768), par exemple: -2

- D'une plage de valeurs entières positives ou négatives (de -32767 à 32768) : format : [min,max], par exemple: [-10,10] : de -10 à 10 inclus.

- De valeurs calculées à partir d'un multiple. Format : N/m (N positif ou négatif, m positif uniquement), par exemple: */3 : tous les multiples de 3.

- De valeurs calculées dans une plage (de -32767 à 32768) selon le multiple spécifié (m positif uniquement) : format [min,max]/m, par exemple: [1,10]/4 : 4 et 8. Chacune des valeurs ou des plages doit être séparée par un point virgule. La liste ne doit pas dépasser 64 caractères. Par exemple: -2;[-10,10];*/3;[1,10]/4.

- Si le second champ (CodeUg) utilise la valeur par défaut, la virgule doit être saisie si un code retour doit être spécifié.

REPL : Option de remplacement. Indique si l'enregistrement extrait (inséré) annule et remplace l'enregistrement déjà existant dans le fichier de destination; par défaut, l'extraction (l'insertion) est refusée pour chaque enregistrement déjà existant (signale par un message pour chaque occurrence).

ENABLE : Option de libération. Valeur par défaut. Indique si la session sera insérée dans l'état libéré. L'action de libération (06) sera nécessaire avant toute modification de la session si l'objet est inséré avec le qualifieur NOENABLE.

LST : Liste des Uprocs de la session. Valeur par défaut de la commande uxshw. Indique que le résultat de la commande uxshw doit être la liste des Uprocs de la session, si une Uproc est citée deux fois ou plus dans la session, elle n'apparaîtra qu'une seule fois dans cette liste.

 Incompatible avec l'utilisation de LNK.

LNK : Liste des Uprocs de la session. Indique que le résultat de la commande uxshw doit présenter la session sous la forme d'un tableau décrivant les liens entre les Uprocs et le chemin normal ou dégradé.

 Incompatible avec l'utilisation de LST.

INF : Option d'affichage. Limite l'affichage au nom des objets. Autre valeur : FULL (par défaut). Affiche toutes les informations disponibles sur la liste des objets.

OUTPUT : Stockage des résultats. Le nom du fichier est obligatoire pour la commande uxext. Si l'item n'est pas renseigné la sortie se fait à l'écran.

INPUT : Fichier d'entrée pour la commande uxins. Le nom du fichier est obligatoire.

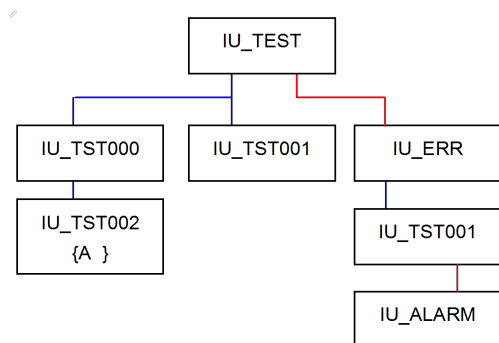
TESP : code de l'espace cible du transfert : X pour Exploitation, S pour Simulation ou I pour intégration.

NOVERIF : peut être utilisé avec la commande uxext uniquement. Cette option permet de ne pas vérifier le fichier l'extraction avant d'y écrire de nouveaux enregistrements et d'améliorer ainsi les performances de cette commande. Cet item ne doit être utilisé que pour un nouveau fichier.

3.18.2 Exemples

```
uxadd SES EXP SES=SES_TEST label=\"Creation session\" HEADER=IU_TEST
FATHER=IU_TEST SONOK=((IU_TST000)(IU_TST001(001))) SONKO=((IU_ERR))
FATHER=IU_TST000 SONOK=((IU_TST002,{A})) FATHER=IU_ERR SONOK=((IU_TST001
(002))) FATHER=IU_TST001(002) SONKO=((IU_ALARM))
```

Crée une session "SES_TEST" dans l'environnement courant (société, nœud) selon le schéma suivant :



```
uxdlr SES EXP SES=SES_TEST
```

Supprime la session "SES_TEST" dans l'environnement courant (société, nœud).

```
uxadd SES EXP SES=D_LOAD_BCK label=\"Chargement du fichier\" HEADER=D_LOAD_
FIL FATHER=D_LOAD_FIL SONOK=((D_BACKUP))
```

Crée la session D_LOAD_BCK composée de l'entête D_LOAD_FIL suivie de l'Uproc D_BACKUP en chemin normal sur la même unité de gestion d'exécution.

```
uxshw SES EXP SES=D_LOAD_BCK LNK
```

Affiche la session créée précédemment avec les Uprocs qui la compose et les liens entre elles.

```
uxext SES EXP SES=D_* output=c:\temp\DOC.ext
```

Extrait toutes les sessions commençant par D_ de l'espace exploitation et archive les données dans le fichier c:\temp\DOC.ext.

```
uxins SES EXP SES=* input=c:\temp\DOC.ext
```

Insère toutes les sessions du fichier c:\temp\DOC.ext dans l'environnement courant (société, nœud) en espace exploitation sauf si elles existent déjà.

```
uxshw ses ses=MYSESSION lnk vses=*
TYPE .| ses
ITEMS .| ses : MYSESS
.| vses : 000
.| label :
.| header : HEADER
.| links :
.|
.| | NumF | NumS | Path | Father .| Son | HDP/MU | Return Code ||
.| | 0001 | 0002 | OK | HEADER | MYSONOK | { } | ||
.| | 0001 | 0003 | CR | | MYSONCR4 | { } | 4 ||
.| | 0001 | 0004 | CR | | MYSONANY | { } | * ||"
```

La colonne "Return Code" affiche le code retour attendu pour déclencher cette Uproc dans le cas d'un lien. Code retour entre l'Uproc parent et l'Uproc fille : code retour 4 pour l'Uproc MYSONCR4, code retour * pour l'Uproc MYSONANY.

3.19 Les calendriers

uxadd CAL : crée un calendrier dans la liste des calendriers ou une ou plusieurs années sur un calendrier existant.

Item	Format	Défaut
EXP SIM INT APP	3c	S_ESPEXE
MU=	64c	
LABEL=	64c	" "
NOMODEL MODEL		NOMODEL
FRHOLI NOFRHOLI		FRHOLI
SINCE=	YYYY	
BEFORE=	YYYY	
WEEK= W C H	7c	WWWWWCC

Exemple :

```
uxadd CAL EXP MU=\" 00\" LABEL=\"General calendar\" SINCE=2001 BEFORE=2011
```

uxupd CAL : modifie les caractéristiques d'un calendrier.

Item	Format	Défaut
EXP SIM INT APP	3c	S_ESPEXE
MU=	64c *	
LABEL=	64c	
NOMODEL MODEL		
DSINCE=	U_FMT_DATE	
DBEFORE=	U_FMT_DATE	
ATTR=W C H ALL		
TO=W C H		
DAY=MON TUE WED THU FRI SAT SUN ALL		ALL

Exemple :

```
uxupd CAL EXP MU=HEADQUART DSINCE=20021226 DBEFORE=20021231 ATTR=ALL TO=C
```

uxshw CAL : affiche les caractéristiques d'un calendrier.

Item	Format	Défaut
EXP SIM INT APP	3c	S_ESPEXE
MU=	64c *	
NOMODEL MODEL		
SINCE=	YYYY	
BEFORE=	YYYY	
OUTPUT=file_name		

Exemple :

```
uxshw CAL EXP MU=HEADQUART SINCE=2002 BEFORE=2002
```

uxdlit CAL : supprime un calendrier.

Item	Format	Défaut
EXP SIM INT APP	3c	S_ESPEXE
MU=	64c *	
NOMODEL MODEL		NOMODEL
SINCE=	YYYY	
BEFORE=	YYYY	

Exemple :

```
uxdlit CAL EXP MU=HEADQUART SINCE=1997 BEFORE=2000
```

uxlst CAL : liste les calendriers.

Item	Format	Défaut
EXP SIM INT APP	3c	S_ESPEXE
MU=	64c *	*

Item	Format	Défaut
INF FULL		FULL
OUTPUT=file_name		

Exemple :

```
uxlst CAL OUTPUT=c:\temp\LstCal.txt
```

uxtra CAL : transfère les calendriers dans un autre espace de la même société sur le même nœud.

Item	Format	Défaut
APP INT SIM	3c	S_ESPEXE
MU=	64c *	
MODEL NOMODEL		NOMODEL
TESP=	I S X	

Exemple :

```
uxtra CAL app MU=D_* MODEL TESP=X
```

uxext CAL : extrait la définition d'un (modèle de) calendrier (étendue complète).

Item	Format	Défaut
EXP SIM INT APP	3c	S_ESPEXE
UG= MU=	64c *	
MODEL		
REPL		
OUTPUT=file_name		
NOVERIF		

Exemple :

```
uxext CAL EXP MU=HEADQUART output=c:\temp\cal.ext
```

uxins CAL : insère la définition d'un calendrier dans les fichiers correspondants de Dollar Universe (étendue complète) pour le même espace et la même version de Dollar Universe que l'extraction.

Item	Format	Défaut
EXP SIM INT APP	3c	S_ESPEXE
UG= MU=	64c *	
MODEL		
CIBLE=	(CODE_UG,...)	
REPL		
INPUT=file_name		

Exemple :

```
uxins CAL EXP MU=* CIBLE=HEADQUART INPUT=c:\temp\cal.ext
```

3.19.1 Description des items

MU (ou **UG**) : Code de l'unité de gestion du Calendrier. Format : 64 caractères alphanumériques. Permet la saisie d'un modèle ou d'une unité de gestion (valeur générique acceptée sauf pour uxadd). Le calendrier général peut être manipulé par l'intermédiaire de l'unité de gestion " 00" (espace 0 0) ou " " (espace).

MODEL : Type du calendrier. Valeur par défaut : NOMODEL. Un calendrier de type MODEL devra être extrait / inséré pour devenir opérationnel. "UG" exprime alors le nom du modèle.

LABEL : Libellé descriptif du calendrier. Format : 64 caractères alphanumériques.

SINCE : Première année du calendrier. Format : 4 caractères numériques. Indique l'année à partir de laquelle le calendrier devra débuter (valeur minimum : année courante moins une).

BEFORE : Dernière année du calendrier. Format : 4 caractères numériques. Indique la dernière année créée (incluse) du calendrier (valeur minimum : année courante plus deux). L'écart entre la première et la dernière année du calendrier ne peut dépasser 24 ans.

FRHOLI : Avec jours fériés français. Indique que le calendrier sera créé avec les jours fériés français. Si le mot clé NOFRHOLI est indiqué, le calendrier sera créé sans les jours fériés français (valeur par défaut).

WEEK : Masque de création. Indique quel masque de création pour la semaine est utilisé pour créer le calendrier. Format : pour les sept jours de la semaine (du lundi au dimanche) le type du jour peut être :

- w : ouvré
- c : chômé
- h : férié

Soit par exemple : week=(wwwwwwcc) : ouvré du lundi au vendredi, samedi et dimanche chômés (valeur par défaut).

DSINCE : jour de début de la modification. Format : indiqué par le paramètre U_FMT_DATE. Indique le premier jour à partir de laquelle la modification est appliquée.

DBEFORE : jour de fin de la modification. Format : indiqué par le paramètre U_FMT_DATE. Indique le dernier jour (inclus) jusqu'à laquelle la modification est appliquée.

ATTR : sélection sur le type des jours. Indique quel type de jours doit être affecté par la modification. Format : w (ouvré), c (chômé), h (fériel) ou all (valeur par défaut).

TO : Nouveau type. Indique le nouveau type des jours sélectionnés. Format : w (ouvré), c (chômé), h (fériel).

DAY : Sélection en fonction du jour de la semaine. Indique quel jour dans la semaine (ou quel ensemble de jours) doit être

- *: accepté.
- Un code d'Unité de Gestion, si seulement une Unité de Gestion est ciblée.
- (CODE_UG, CODE_UG, ...) si plusieurs Unités de Gestion sont ciblées, CODE_UG étant le code de l'Unité de Gestion.

Permet d'indiquer, pour les calendriers modèles sélectionnés, les unités de gestion cibles de l'insertion (démultiplication du modèle en autant de calendriers que d'unités de gestion indiquées dans la cible).

Si deux modèles distincts, par le biais des cibles décrites, donnent lieu à la génération d'un même calendrier (éventuellement dans la même commande), la deuxième insertion est rejetée, tout comme si le calendrier existait au préalable (un message d'avertissement est fourni pour chaque rejet). Il convient donc d'utiliser l'option REPL pour palier ces rejets (c'est alors la dernière insertion qui subsiste).

Si CIBLE=*, tous les calendriers non modèles contenus dans le fichier d'entrée seront insérés avec leur Unité de Gestion d'origine, qui doit exister sur la machine cible. Si l'UG n'existe pas le calendrier ne sera pas inséré.

Cette fonctionnalité permet d'implémenter aisément des opérations de sauvegarde/restauration pour toute la planification en utilisant les quatre commandes listées ci-dessous. Il est recommandé d'exporter/importer les calendriers modèles et non modèles séparément.

```
uxext CAL MU=* OUTPUT=c:\temp\cal_nomod.ext
uxins CAL MU=* CIBLE=* INPUT=c:\temp\cal_nomod.ext
uxext CAL MU=* MODEL OUTPUT=c:\temp\cal_mod.ext
uxins CAL MU=* MODEL CIBLE=* INPUT=c:\temp\cal_mod.ext
```

REPL : Option de remplacement. Indique si l'enregistrement extrait (inséré) annule et remplace les années déjà existantes dans le fichier (ou le calendrier) de destination; par défaut, l'extraction (l'insertion) est refusée pour chaque enregistrement déjà existant (signale par un message pour chaque occurrence).

INF : Option d'affichage. Limite l'affichage au nom des objets. Autre valeur : FULL (par défaut). Affiche toutes les informations disponibles sur la liste des objets.

OUTPUT : Stockage des résultats. Le nom du fichier est obligatoire pour la commande uxext. Si l'item n'est pas renseigné la sortie se fait à l'écran.

INPUT : Fichier d'entrée pour la commande uxins. Le nom du fichier est obligatoire.

TESP : code de l'espace cible du transfert : X pour Exploitation, S pour Simulation ou I pour intégration.

NOVERIF : peut être utilisé avec la commande uxext uniquement. Cette option permet de ne pas vérifier le fichier l'extraction avant d'y écrire de nouveaux enregistrements et d'améliorer ainsi les performances de cette commande. Cet item ne doit être utilisé que pour un nouveau fichier.

3.19.2 Exemples

```
uxadd CAL EXP MU="\ 00\" SINCE=2001 BEFORE=2015 LABEL="\Calendrier
general\"
uxadd CAL EXP MU=SIEGE LABEL="\Calendrier du siege\" SINCE=1995 BEFORE=2015
```

Ajoute (ou remplace) les années 2001 à 2015 au calendrier général dans l'environnement courant (société, nœud et espace) et crée le calendrier de l'unité de gestion SIEGE dans l'environnement courant (société, nœud et espace).

```
uxupd CAL EXP MU=SIEGE DSINCE=19991227 DBEFORE=19991231 ATTR=ALL T0=C
```

Modifie le calendrier de l'unité de gestion SIEGE de l'environnement courant (société, nœud et espace) pour rendre chômée la période du 27/12/99 au 31/12/99 si le paramètre U_FMT_DATE vaut YYYYMMDD.

```
uxdlit CAL EXP MU=SIEGE SINCE=1995 BEFORE=1997
```

Supprime les années 1995, 1996 et 1997 du calendrier de l'unité de gestion SIEGE de l'environnement courant (société, nœud et espace).

```
uxext CAL EXP MU=MOD output=c:\temp\DOC.ext
```

Extrait le calendrier modèle "MOD" de l'espace exploitation et archive les données dans le fichier c:\temp\DOC.ext.

```
uxins CAL EXP MODEL MU=MOD CIBLE=MOD input=c:\temp\DOC.ext
```

Insère le calendrier modèle "MOD" du fichier c:\temp\DOC.ext en tant que calendrier modèle en espace d'exploitation de l'environnement courant (société, nœud) sauf s'il existe déjà.

```
uxins CAL EXP MU=MOD CIBLE=SIEGE input=c:\temp\DOC.ext
```

Insère le calendrier modèle "MOD" du fichier c:\temp\DOC.ext, comme calendrier de l'unité de gestion SIEGE dans l'environnement courant (société, nœud) en espace exploitation sauf s'il existe déjà.

3.20 Les règles de planification

uxadd RUL : crée une règle dans la liste des règles.

Item	Format	Défaut
RUL=	64c	
LABEL=	64c	" "
PERIOD=(n[d w m q y h wd cd mo tu we th fr sa su fm fq fy])	4c	(1d)
POS=(±n[w c d wd cd h mo tu we th fr sa su])	5c	(+1d)
AUTHO=(ynn,ynn,ynn,ynn,ynn,ynn,ynn)	29c	(ynn,ynn,ynn,ynn,ynn,ynn,ynn)
OFFSET=N F P	1c	N
AUTHOD=±(dn)	86c	+ (1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24, 25,26,27,28,29, 30,31)
AUTHOM=(mn)	28c	(1,2,3,4,5,6,7,8,9,10,11,12)
TODW=A E S	1c	A
SHIFTW=(±n[d w c h])	1c	
TODC=A E S	1c	A
SHIFTC=(±n[d w c h])	1c	
TODH=A E S	1c	A
SHIFTH=(±n[d w c h])	1c	

Exemples :

```
uxadd RUL RUL="LWDM" LABEL="\last working day of the month\" PERIOD="(1M)" POS="(-1W)"
```

```

uxadd RUL RUL="5DM0+" LABEL="\the 5th of the month offset +" PERIOD="(1M)"
POS="(+5C)" OFFSET=F
uxadd RUL RUL="NO_SUN" LABEL="\except Sundays\" AUTH0="
(YYY,YYY,YYY,YYY,YYY,YYY,NNN)"
uxadd RUL RUL="1-15wd" LABEL="\every working day from 1 to 15\" PERIOD="(1d)
POS=(+1w) AUTHOD=+(1,2,3,4,5,6,7,8,9,10,11,12,13,14,15)

```

uxupd RUL : modifie les caractéristiques d'une règle.

Item	Format	Défaut
RUL=	64c *	
LABEL=	64c	
PERIOD=(n[d w m q y wd cd mo tu we th fr sa su fm fq fy])	4c	

```
uxshw RUL RUL="5DM0+"
```

uxdlit RUL : supprime une règle.

Item	Format	Défaut
RUL=	64c *	

Exemple :

```
uxdlit RUL RUL=NO_SUN
```

uxlst RUL : liste les règles.

Item	Format	Défaut
RUL=	64c *	*
INF FULL		FULL
OUTPUT=file_name		

Exemple :

```
uxlst RUL OUTPUT=c:\temp\LstRu1.txt
```

uxext RUL : extrait la définition d'une règle.

Item	Format	Défaut
RUL=	64c *	
REPL		
OUTPUT=file_name		
NOVERIF		

Exemple :

```
uxext RUL RUL=* output=c:\temp\ru1.ext
```

uxins RUL : insère la définition d'une règle dans les fichiers correspondants de Dollar Universe pour la même version de Dollar Universe que l'extraction.

Item	Format	Défaut
RUL=	64c *	
REPL		
INPUT=file_name		

Exemple :

```
uxins RUL RUL=* INPUT=c:\temp\ru1.ext
```

3.20.1 Description des items

RUL : Code de la règle. Format : 64 caractères : alphanumériques plus les caractères "+" et "-". Permet la saisie du code de la règle (valeur générique acceptée sauf pour uxadd).

LABEL : Libellé descriptif de la règle. Format : 64 caractères alphanumériques.

PERIOD : Périodicité de la règle. Indique quelle est la période de la règle. Format : un multiple entier de

- d : jour
- w : semaine
- m : mois
- q : trimestre
- y : année
- h : jour férié
- wd : jour ouvré
- cd : jour chômé
- mo : lundi
- tu : mardi
- we : mercredi
- th : jeudi
- fr : vendredi
- sa : samedi
- su : dimanche
- fm : mois fiscal
- fq : trimestre fiscal
- fy : année fiscale

POS : Position dans la période. Format : entre parenthèses, 2 caractères numériques signés suivis du type de jour. Par exemple : (-1wd) : moins un jour ouvré. Indique quel numéro de jour est recherché (la numérotation commence à +1 pour le premier jour, à -1 pour le dernier) et le type du jour :

- w : ouvré (pour compatibilité; égal à "wd")
- c : calendrier (pour compatibilité; égal à "d")
- d : jour
- wd : jour ouvré
- cd : jour chômé
- h : jour férié
- mo : lundi
- tu : mardi
- we : mercredi
- th : jeudi
- fr : vendredi
- sa : samedi
- su : dimanche

AUTHO : Masque d'autorisation de la semaine. Valeurs : y (oui) ou n (non). Indique pour chaque jour de la semaine (du lundi au dimanche) la validité du calcul de la date en fonction du type de ce jour (ouvré, chômé ou férié). Format : entre parenthèses. Pour chaque jour est indiqué dans un triplet (ouvré chômé férié) si le calcul est valide, chaque jour est séparé du suivant par des virgules. Par exemple la valeur par défaut est : AUTHO=(ynn,ynn,ynn,ynn,ynn,ynn,ynn).

OFFSET : Report. Format : n (aucun), f (suivant) ou p (précédent). Valeur par défaut : n (aucun). Indique quel report est appliqué si la date calculée n'est pas autorisée dans le masque de la semaine.

AUTHOD : Liste des jours autorisés dans le mois. Par défaut tous les jours sont autorisés. "+" indique que les jours sont comptés à partir du début du mois, "-" indique que les jours sont comptés à partir de la fin du mois. Format : 86 caractères maximum. Entre parenthèses, "1" indique le premier jour à partir du début (+) ou de la fin (-), "2" indique le deuxième jour à partir du début (ou de la fin), etc.

AUTHOM : Liste des mois autorisés dans l'année. Par défaut tous les mois sont autorisés. Format : 28 caractères maximum. Entre parenthèses, chaque mois est représenté par un chiffre : 1 pour janvier, 2 pour février, etc. Un mois absent est un mois pour lequel le calcul de la date est interdit.

TODW : autorisation sur le type jour ouvré : Format 1 caractère :

- A : autorisé
- E : exclu
- S : exclu avec report

SHIFTW : report sur les jours ouvrés, obligatoire si TODW=S

- H : jour férié
- W : jour ouvré
- C : jour chômé

TODC : autorisation sur le type jour chômé : Format 1 caractère :

- A : autorisé
- E : exclu
- S : exclu avec report

SHIFTC : report sur les jours chômés, obligatoire si TODC=S

- H : jour férié
- W : jour ouvré
- C : jour chômé

TODH : autorisation sur le type jour férié : Format 1 caractère :

- A : autorisé
- E : exclu
- S : exclu avec report

SHIFTH : report sur les jours fériés, obligatoire si TODH=S

- H : jour férié
- W : jour ouvré
- C : jour chômé

INF : Option d'affichage. Limite l'affichage au nom des objets. Autre valeur : FULL (par défaut). Affiche toutes les informations disponibles sur la liste des objets.

REPL : Option de remplacement. Indique si la règle extraite (insérée) annule et remplace la règle existante dans le fichier (ou la liste des règles) de destination; par défaut, l'extraction (l'insertion) est refusée pour chaque enregistrement déjà existant (signalé par un message pour chaque occurrence).

OUTPUT : Stockage des résultats. Le nom du fichier est obligatoire pour la commande uxext. Si l'item n'est pas renseigné la sortie se fait à l'écran.

INPUT : Fichier d'entrée pour la commande uxins. Le nom du fichier est obligatoire.

NOVERIF : peut être utilisé avec la commande uxext uniquement. Cette option permet de ne pas vérifier le fichier l'extraction avant d'y écrire de nouveaux enregistrements et d'améliorer ainsi les performances de cette commande. Cet item ne doit être utilisé que pour un nouveau fichier.

3.20.2 Exemples

```
uxadd RUL RUL="DJOMOI" label=\"Dernier jour ouvre du mois\" period=(1m)
pos=(-1wd)
```

Crée la règle "dernier jour ouvré du mois" de période 1 mois, avec une position sur le dernier jour ouvré sans autorisation ni report particulier, dans l'environnement courant (société et nœud).

```
uxadd RUL RUL=05JM_ label=\ "Le 5eme jour du mois ouvre report +\" PERIOD=
(1m) POS=(+5d) OFFSET=f
```

Crée la règle "cinquième jour du mois et report au jour ouvré suivant" de période 1 mois, avec une position sur cinquième jour du mois avec le masque par défaut (jours ouvrés seuls autorisés) et report au suivant, dans l'environnement courant (société et nœud).

```
uxadd RUL RUL=NO_SUN label=\ "Sauf les dimanches\" period=(1d) pos=(+1d)
autho=(YYY,YYY,YYY,YYY,YYY,YYY,NNN)
```

Crée la règle "tous les jours sauf le dimanche" de période 1 jour, avec une position sur le jour même avec un masque d'autorisation qui refuse le dimanche quel que soit le type de ce jour, dans l'environnement courant (société et nœud).

```
uxadd RUL RUL=10DMFY8 LABEL=\ "ev 10d;not 21to25;not in aug\" PERIOD=(10d)
POS=(+1wd) AUTHO=(yyy,yyy,yyy,yyy,yyy,nnn,nnn) AUTHOD=+
(1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,26,27,28,29,30,31)
AUTHOM=(1,2,3,4,5,6,7,9,10,11,12)
```

Crée la règle "tous les 10 jours, sauf du 21 au 25, sauf en août", avec une période de 10 jours, position sur le premier jour calendaire de la période, tous les jours de la semaine sauf samedi et dimanche, sauf du 21 au 25 inclus, sauf au mois d'août.

```
uxlst RUL
```

Liste les règles de l'environnement courant (société et nœud).

```
uxdlt RUL RUL=NO_SUN
```

Supprime la règle NO_SUN de l'environnement courant (société et nœud).

3.21 Les tâches

uxadd TSK : crée une tâche dans la liste des tâches. Cette commande ne crée que l'enveloppe de la Tâche, les autres informations doivent être saisie par l'intermédiaire de la commande uxupd.

Item	Format	Défaut
EXP SIM INT APP	3c	S_ESPEXE
TSK=	64c	TSK(SES)(VSES)(UPR)(VUPR)
SES=	64c	" "
VSES=	3n	
UPR=	64c	
VUPR=	3n	
MU=	64c	
MODEL NOMODEL		NOMODEL

TECHINF : information générales.

Item	Format	Défaut
TYPE=S O P C	1c	
LABEL=	64c	
USER=	64c	

Item	Format	Défaut
PRINTER=	4c	IMPR
PRTY=	3n	100
QUEUE=	31c	SYS_BATCH (SYS\$BATCH sous OpenVMS)
PDATE=(±n,±n,W C)		(0,0,C)
[OFFSET]=±nn[D C H W] or NONE		NONE
PDATE_OFFSET=Y N	1c	N
HOLD NOHOLD		NOHOLD
DISABLE ENABLE		ENABLE
MULT NOMULT		NOMULT
AUTOREL NOAUTOREL		NOAUTOREL
FORCE NOFORCE		NOFORCE
HISTORY NOHISTORY		HISTORY
SERIAL PARALLEL		SERIAL
OPT_W_EVENT OPT_WO_EVENT		OPT_WO_EVENT

Exemple :

```
uxadd TSK EXP SES=D_LOAD_BCK UPR=D_LOAD_FIL MU=HEADQUART TECHINF TYPE=S
USER=admin
```

uxupd TSK : modifie les caractéristiques d'une tâche et définit les informations complémentaires lors d'une création.

Item	Format	Défaut
EXP SIM INT APP	3c	S_ESPEXE
TSK=	64c *	*
SES=		

Famille	Description
CTLDUR	Limite de temps. Permet d'arrêter l'exécution de la Tâche si elle dépasse le temps indiqué.
TIME	Horaires de lancement. Permet la définition des heures de lancement journaliers ou multi journaliers pour des tâches planifiées ou optionnelles, des heures de lancement pour les tâches cycliques ainsi que les heures de lancement ou d'exclusion d'une tâche provoquée.  Pour passer une tâche de multi-journalière à journalière (ou le contraire), il faut tout d'abord changer le type de la tâche (MULT ou NOMULT) dans une première commande uxupd, puis définir les horaires (TIME) dans une seconde commande.
IMPL	Planification implicite. Permet la définition de règles de planification et de leurs dates d'application ainsi que la date de première planification de la tâche.
EXPLI	Dates explicite. Permet la définition de dates et heures explicites de planification de la tâche ainsi que la date de traitement associée.
EXCLU	Dates d'interdiction. Permet la définition de dates d'interdiction visant à interdire le calcul de la planification (à partir des règles définies pour la tâche).
INTERVAL	Fenêtre d'exclusion. Permet la définition de fenêtres d'exclusion visant à interdire le calcul de la planification (à partir des règles définies pour la tâche).
LAUNCHVAR	Variables d'une tâche. Permet la création et la suppression des variables dans la tâche et leur valorisation. Ces variables doivent être choisies dans la liste des variables de l'Uproc identifiant la tâche.

TECHINF: informations générales.

Item	Format	Défaut
TYPE=S O P C	1c	
USER=	64c	
PRINTER=	4c	
PRTY=	3n	
QUEUE=	31c	
PDATE=(±n,±n,W C)		
HOLD NOHOLD		
DISABLE ENABLE		
MULT NOMULT		
AUTOREL NOAUTOREL		
FORCE NOFORCE		
HISTORY NOHISTORY		
SERIAL PARALLEL		
OPT_W_EVENT OPT_WO_EVENT		

Exemple :

```
uxupd TSK EXP SES=D_LOAD_BCK UPR=D_LOAD_FIL MU=HEADQUART TECHINF
USER=univ54e PDATE="( -1, 0, W)"
```

PARENT :

Item	Format	Défaut
PARENT_TSK=TSK_CODE ANY	64c	ANY
PARENT_MU= MU_CODE ANY	64c	ANY

```
uxupd TSK EXP SES=D_LOAD_BCK UPR=D_LOAD_FIL MU=HEADQUART PARENT
```

CTLDUR :

Item	Format	Défaut
HEULIM=(HHMM,B)		

```
uxupd TSK EXP SES=D_LOAD_BCK UPR=D_LOAD_FIL MU=HEADQUART CTLDUR
```

IMPL add : ajoute une règle de planification à la tâche.

Item	Format	Défaut
RULE=((RULE_CODE,U_FMT_DATE)(.....))		
SDATE=	U_FMT_DATE	(obsolète)
START_VALID=	U_FMT_DATE	
END_VALID=	U_FMT_DATE ou NONE	

Exemple :

```
uxupd TSK EXP SES=D_LOAD_BCK UPR=D_LOAD_FIL MU=HEADQUART IMPL add rule="
(("LWDM",20020701)("5DMO+",20020701))"
uxupd TSK EXP SES=D_LOAD_BCK UPR=D_LOAD_FIL MU=HEADQUART IMPL add rule="
(("++FWD",01/01/2010))"
uxupd TSK EXP SES=D_LOAD_BCK UPR=D_LOAD_FIL MU=HEADQUART IMPL add rule="
(("--FWD",01/01/2010))"
```

IMPL dlt : supprime une règle de planification de la tâche.

Item	Format
RULE=	((RULE_CODE,U_FMT_DATE)(.....))

Exemple :

```
uxupd TSK EXP SES=D_LOAD_BCK UPR=D_LOAD_FIL MU=HEADQUART IMPL dlt rule="
("5DMO+",20020701)"
```

EXPLI add : ajoute une date explicite à la tâche.

Item	Format
EDATE=	((U_FMT_DATE,hhmm,hhh,mm,U_FMT_DATE)(.....))

Exemple :

```
uxupd TSK EXP SES=D_LOAD_BCK UPR=D_LOAD_FIL MU=HEADQUART EXPLI add EDATE="
((20021230,2100,005,00,20021231)(20021130,2100,005,00,20021130))"
```

EXPLI dlt : supprime une date explicite de la tâche.

Item	Format
EDATE=	((U_FMT_DATE,hhmm,hhh,mm,U_FMT_DATE)(.....))

Exemple :

```
uxupd TSK EXP SES=D_LOAD_BCK UPR=D_LOAD_FIL MU=HEADQUART EXPLI dlt EDATE="
(20021130,2100,005,00,20021130)"
```

EXCLU add : ajoute une date d'interdiction à la tâche.

Item	Format
XDATE=	((U_FMT_DATE)(...))

Exemple :

```
uxupd TSK EXP SES=D_LOAD_BCK UPR=D_LOAD_FIL MU=HEADQUART EXCLU add XDATE="
((20021231)(20030101))"
```

EXCLU dlt : supprime une date d'interdiction de la tâche.

Item	Format
XDATE=	((U_FMT_DATE)(...))

Exemple :

```
uxupd TSK EXP SES=D_LOAD_BCK UPR=D_LOAD_FIL MU=HEADQUART EXCLU dlt XDATE="
((20021231))"
```

INTERVAL add : ajoute une fenêtre d'exclusion à la tâche.

Item	Format
EXCLUSION=	((U_FMT_DATE,hhmm,U_FMT_DATE,hhmm)(...))

Exemple :

```
uxupd TSK EXP SES=D_LOAD_BCK UPR=D_LOAD_FIL MU=HEADQUART INTERVAL add
EXCLUSION="((20061230,2100,20070331,1630)(20071201,2000,20071231,1000))"
```

INTERVAL dlt : supprime une fenêtre d'exclusion de la tâche.

Item	Format
EXCLUSION=	((U_FMT_DATE,hhmm,U_FMT_DATE,hhmm)(...))

Exemple :

```
uxupd TSK EXP SES=D_LOAD_BCK UPR=D_LOAD_FIL MU=HEADQUART INTERVAL dlt
EXCLUSION="((20061230,2100,20070331,1630))"
```

TIME : horaire de planification journalier.

Item	Format
DALW=	(hhmm,hhh,mm)
MON=	(hhmm,hhh,mm)
TUE=	(hhmm,hhh,mm)
WED=	(hhmm,hhh,mm)
THU=	(hhmm,hhh,mm)
FRI=	(hhmm,hhh,mm)

Item	Format
SAT=	(hhmm,hhh,mm)
SUN=	(hhmm,hhh,mm)

Exemple :

```
uxupd TSK EXP SES=D_LOAD_BCK UPR=D_LOAD_FIL MU=HEADQUART TIME DALW="
(2100,005,00)" WED="(1900,005,00)"
```

TIME : horaire de planification multi-journalier.

Item	Format
MULW=	((hhmm,hhmm,mmm),hhh,mm)

Exemple :

```
uxupd TSK EXP UPR=D_LOAD_FIL MU=HEADQUART TIME MULW="
((2000,2359,030),000,15)"
```

TIME : horaire de planification d'une tâche à relance automatique.

Item	Format
RELW=	((hhmm),hhh,mm,hhh,mm,nnnn,hhh,mm)

Exemple :

```
uxupd TSK EXP UPR=D_LOAD_FIL MU=HEADQUART TIME RELW="
((2000,2359)0010,000,0000)"
```

TIME : horaire de planification d'une tâche provoquée.

Item	Format
PRLW=	(hhmm,hhh,mm,hhmm,hhmm) ou (ONDEMAND,hhh,mm,hhmm,hhmm)

Exemple :

```
uxupd TSK EXP UPR=D_LOAD_FIL MU=HEADQUART TECHINF TYPE=P
uxupd TSK EXP UPR=D_LOAD_FIL MU=HEADQUART TIME PRLW="
(0800,012,00,1300,1400)"
```

TIME : horaire de planification d'une tâche cyclique.

Item	Format
CYCLICAL=	((hhhhh,mm),(U_FMT_DATE),(hhmm),(hhh,mm))

Exemple :

```
uxupd TSK EXP UPR=D_LOAD_FIL MU=HEADQUART TECHINF TYPE=C
uxupd TSK EXP UPR=D_LOAD_FIL MU=HEADQUART TIME CYCLICAL="((00022,30),
(20060823),(1837),(000,10))"
```

LAUNCHVAR add : ajoute la valorisation d'une variable.

Item	Format
VARNAME=	20c

Item	Format
VVALUE=	dépend de la définition de la variable

Exemple :

```
uxupd TSK UPR=D_LOAD_FIL MU=HEADQUART LAUNCHVAR add VARNAME=VARTXT
VVALUE=XY_VARTXT_VALTSK
```

LAUNCHVAR dlt : supprime la valorisation d'une variable.

Item	Format
VARNAME=	20c

Exemple :

```
uxupd TSK UPR=D_LOAD_FIL MU=HEADQUART LAUNCHVAR dlt VARNAME=VARTXT
```

uxdup TSK : duplique une tâche.

Item	Format	Défaut
EXP SIM INT APP	3c	S_ESPEXE
TSK=	64c	
SES=	64c	
VSES=	3n	
UPR=	64c	
VUPR=	3n	
MU=	64c	
TTYPE=S O P C	1c	
MODEL NOMODEL		NOMODEL
TTSK=	64c	
TSES=	64c	
TVSES=	3n	
TUPR=	64c	
TVUPR=	3n	
TMU=	64c	
TMODEL TNOMODEL		TNOMODEL

Exemple :

```
uxdup TSK EXP SES=D_LOAD_BCK UPR=D_LOAD_FIL VUPR=000 MU=HEADQUART
TSES=D_LOAD_BCK TUPR=D_LOAD_FIL TVUPR=000 TMU=HEADQUART TMODEL
```

uxshw TSK : affiche les caractéristiques d'une tâche.

Item	Format	Défaut
EXP SIM INT APP	3c	S_ESPEXE
TSK=	64c *	

Item	Format	Défaut
SES=	64c *	" "
VSES=	3n *	
UPR=	64c *	
VUPR=	3n *	
MU=	64c *	
MODEL NOMODEL		NOMODEL
TTYPE=S O P C *	1c *	
OUTPUT=file_name		

Exemple :

```
uxshw TSK EXP SES=D_LOAD_BCK UPR=D_LOAD_FIL MU=HEADQUART|more
```

uxdlit TSK : supprime une tâche.

Item	Format	Défaut
EXP SIM INT APP	3c	S_ESPEXE
TSK=	64c *	
SES=	64c *	" "
VSES=	3n *	
UPR=	64c *	
VUPR=	3n *	
MU=	64c *	
MODEL NOMODEL		NOMODEL
TTYPE=S O P C *	1c *	

Exemple :

```
uxdlit TSK EXP UPR=D_LOAD_FIL MU=HEADQUART
```

uxlst TSK : liste les tâches.

Item	Format	Défaut
EXP SIM INT APP	3c	S_ESPEXE
TSK=	64c *	*
SES=	64c *	*
VSES=	3n *	*
UPR=	64c *	*
VUPR=	3n *	*
MU=	64c *	*
MODEL NOMODEL		NOMODEL
TTYPE=S O P C *	1c *	
INF FULL		INF
MUTIME ASTIME		

Item	Format	Défaut
OUTPUT=file_name		

Exemple :

```
uxlst TSK FULL OUTPUT=c:\temp\LstTsk.txt
```

uxtra TSK : transfère les tâches dans un autre espace de la même société sur le même nœud.

Item	Format	Défaut
APP INT SIM	3c	S_ESPEXE
TSK=	64c *	
SES=	64c *	""
VSES=	3n *	
UPR=	64c *	
VUPR=	3n *	
MU=	64c *	
MODEL NOMODEL		NOMODEL
TESP=	I S X	

Exemple :

```
uxtra TSK app UPR=D_* VUPR=* SES=D_* VSES=* MU=* TESP=X
```

uxext TSK : extrait la définition complète d'une tâche (modèle ou non).

Item	Format	Défaut
EXP SIM INT APP	3c	S_ESPEXE
TSK=	64c *	
SES=	64c *	*
VSES=	3n *	
UPR=	64c *	
VUPR=	3n *	
MU=	64c *	
MODEL		NOMODEL
REPL		
OUTPUT=file_name		
NOVERIF		

 TSK= ou UPR= est obligatoire dans la commande.

Exemple :

```
uxext TSK EXP MU=H* UPR=D_* OUTPUT=c:\temp\tsk.ext
```

uxins TSK : insère la définition d'une tâche (modèle ou non) dans les fichiers correspondants de Dollar Universe pour le même espace et la même version de Dollar Universe que l'extraction et dans l'état Activé/Désactivé qu'elle possédait au moment de son extraction.

Item	Format	Défaut
EXP SIM INT APP	3c	S_ESPEXE
TSK=	64c *	
SES=	64c *	" "
VSES=	3n *	
UPR=	64c *	
VUPR=	3n *	
MU=	64c *	
MODEL		NOMODEL
CIBLE=(MU1,MU2,...) *		
REPL		
INPUT=file_name		

 TSK= ou UPR= est obligatoire dans la commande.

Exemples :

```
uxins TSK EXP MU=* UPR=* CIBLE=HEADQUART INPUT=c:\temp\tsk.ext
uxins TSK EXP MU=* UPR=* MODEL CIBLE=model INPUT=c:\temp\tsk.ext
```

Si des tâches ont été extraites de l'espace exploitation, elles peuvent être insérées dans les espaces simulation, intégration ou application en définissant au préalable la nouvelle valeur de la version (paramètres nœud S_VUPR_REINSERT et S_VSES_REINSERT, catégorie Réglages de l'interface commande) :

```
S_VUPR_REINSERT=XYZ
S_VSES_REINSERT=STU
```

- XYZ étant la nouvelle valeur de la version de l'Uproc soit au minimum 001 pour insérer la tâche dans les espaces intégration ou application.
- STU étant la nouvelle valeur de la version de la session soit au minimum 001 pour insérer la tâche dans les espaces intégration ou application.

 Toutes les commandes d'insertion suivantes utiliseront ces paramètres pour définir la version des Uprocs insérées (quelque soit l'espace cible).

3.21.1 Description des items

UPR : Code Uproc. Format : 64 caractères alphanumériques. Requiert la saisie d'un code Uproc (valeur générique acceptée sauf pour uxadd) obligatoire si TSK n'est pas indiqué.

TUPR : Code de l'Uproc de la tâche résultat de la duplication. Format : 64 caractères alphanumériques.

VUPR : Version d'Uproc. Format : 3 caractères numériques. Permet la saisie d'une version particulière d'une Uproc. Par défaut, toutes les versions des Uprocs renseignées sont considérées (inutile avec SIM et EXP).

TVUPR : version de l'Uproc de la tâche résultat de la duplication. Format : 3 caractères numériques. Obligatoire pour uxdup tsk.

TSK : Code de la Tâche, obligatoire si UPR n'est pas indiqué. Format : 64 caractères alphanumériques. Pour uxadd, si le code de la Tâche n'est pas renseigné, par défaut il vaut : TSK(SSS)(VVV)(UUU)(VVV) avec :

- (SSS)(VVV) : code et version de la session (si elle existe)
- (UUU)(VVV) : code et version de l'Uproc

TTSK : Code de la Tâche résultat de la duplication. Format : 64 caractères alphanumériques.

SES : Code session. Format : 64 caractères alphanumériques. Requiert la saisie d'un code session (valeur générique acceptée sauf pour uxadd).

TSES : Code de la session de la tâche résultat de la duplication. Format : 64 caractères alphanumériques.

VSES : Version de session. Format : 3 caractères numériques. Permet la saisie d'une version particulière d'une session. Par défaut, toutes les versions des sessions renseignées sont considérées (inutile avec SIM et EXP).

TVSES : version de la session de la tâche résultat de la duplication. Format : 3 caractères numériques.

UG ou **MU** : Code du modèle ou de l'unité de gestion. Format : 64 caractères alphanumériques. Requiert la saisie d'un code unité de gestion (valeur générique acceptée sauf pour uxadd), saisie obligatoire.

TMU : code de l'unité de gestion de la tâche résultat de la duplication. Format : 64 caractères alphanumériques.

TYPE : Type de la tâche lors de sa définition. Format : S (planifiée), O (optionnelle), P (provoquée) ou C (cyclique).

TTYPE : Filtre sur le type de la tâche. Format : S (planifiée), O (optionnelle), P (provoquée) ou C (cyclique).

MODEL : Nature de la tâche. Indique que la tâche est modèle. Une tâche non modèle ne porte pas de mot clé. Dans le cas de la commande uxins, MODEL indique que la tâche doit être insérée comme modèle. Si rien n'est spécifié, la tâche sera insérée en non modèle, quelle que soit sa nature d'origine.

TNOMODEL : Nature de la tâche résultat de la duplication. Indique que la tâche dupliquée doit être non modèle. Pour dupliquer une tâche en modèle, le mot clé TMODEL doit être utilisé.

PDATE : Algorithme de calcul de la date de traitement. Permet de définir automatiquement une date de traitement différente de la date de traitement qui aurait été calculée par l'automate. Valeur par défaut : la date de traitement est calculée par rapport au jour du premier pilotage du lancement. Format : (J,U,T) où :

- J : est le nombre de jours, signé, pour une Uproc avec période fonctionnelle, 0 sinon,
- U : est le nombre d'unités de la période fonctionnelle, si la période est jour, ce nombre vaut 0,
- T est le type de décalage en jours ouvrés : W ou calendaires : C.

Par exemple : PDATE=(-1,0,C) : moins un jour calendaire du même mois.

OFFSET : Décalage appliqué sur la date de planification calculée par le calculateur (d'après les règles). Le décalage comprend un sens (+ ou -) et un nombre de jours. Format : ±nn[D|C|H|W] ou NONE

- ±nn: nombre de jours (positif ou négatif), inférieur ou égal à 99.
- D: décalage en jours calendaires
- C: décalage en jours chômés
- H: décalage en jours fériés
- W: décalage en jours ouvrés
- NONE : pas de décalage

PDATE_OFFSET : Décalage sur la date de traitement.

- Y: la date de traitement est calculée après le calcul de OFFSET (éventuellement)
- N: la date de traitement est calculée d'après la date originale en V5 mais d'après la date du lancement en v6.

Par exemple : si une règle est créée qui spécifie un lancement pour le premier jour ouvré de la semaine. Une tâche utilise cette règle avec un décalage de 1. Si lundi est chômé, le lancement sera le mardi. La date de traitement sera :

En v5 :

- Mercredi si PDATE_OFFSET=Y.
- Lundi si PDATE_OFFSET=N.

En v6 :

- Mercredi si PDATE_OFFSET=Y.
- Mardi si PDATE_OFFSET=N.

PRINTER : Imprimante. Non utilisé sous UNIX and Windows.

PRTY : Priorité. Format : 3 caractères numériques de 001 à 255. Valeur par défaut : 100. Priorité d'entrée de la tâche dans la queue batch.

QUEUE : Queue batch. Format : 31 caractères alphanumériques (" \$" accepté sous OpenVMS). Queue batch de soumission de la tâche. Valeur par défaut : SYS_BATCH (SYS\$BATCH sous OpenVMS).

USER : Compte de soumission. Format : 64 caractères alphanumériques. Compte dans l'environnement duquel l'exécution sera soumise.

HOLD : Tâche simulée. Valeur par défaut : NOHOLD: tâche non simulée. HOLD et DISABLE sont incompatibles.

DISABLE : Tâche désactivée. ENABLE : tâche activée

 HOLD et DISABLE sont incompatibles.

FORCE : Lancement forcé en fin de période. Valeur par défaut : NOFORCE. Indique que l'exécution du lancement sera forcée en fin de plage horaire de la tâche même si toutes les conditions de l'Uproc ne sont pas vérifiées.

HISTORY : mémorisation de l'exécution des Uprocs de la tâche dans l'historique des exécutions. Valeur par défaut : HISTORY. Autre valeur : NOHISTORY : pas de mémorisation.

SERIAL : Type de lancement. Valeur par défaut : SERIAL. Indique que le prochain lancement de la tâche sera calculé à la fin de la plage horaire de la tâche. La valeur PARALLEL indique que des lancements en parallèle sont possibles car le prochain lancement est calculé au début de la fenêtre de lancement.

OPT_W_EVENT : pour une tâche optionnelle, l'événement sera créé (statut Terminé, si l'Uproc est mémorisée) même si la tâche optionnelle ne s'est pas exécutée du fait de sa planification. Valeur par défaut : OPT_WO_EVENT: l'événement ne sera pas créé.

PARENT_TSK : code de la Tâche principale à la quelle la Tâche spécifique (déjà créée) va être liée. PARENT_TSK et PARENT_MU doivent être cohérents. ANY : la prise en compte de la tâche optionnelle par la tâche principale repose sur les codes UG et Session (comportement de Dollar Universe v5)

PARENT_MU : code de l'Unité de Gestion de la Tâche principale à la quelle la Tâche spécifique (déjà créée) va être liée. PARENT_TSK et PARENT_MU doivent être cohérents. ANY : la prise en compte de la tâche optionnelle par la tâche principale repose sur les codes UG et Session (comportement de Dollar Universe v5)

HEULIM : Contrôle l'heure limite d'exécution de la Tâche. Au-delà de cette heure, la tâche sera arrêtée.
Format : (HHMM,B) :

- HHMM : est l'heure limite d'exécution de la Tâche : de toutes les Uprocs s'il s'agit d'une session.
- (HHMM,B) : indique que les jobs en cours de la tâche sont suspendus (sous UNIX et OpenVMS uniquement) au-delà de l'heure indiquée
- (HHMM,) : indique que les jobs sont arrêtés au-delà de l'heure indiquée.

MULT : Lancements multiples. Indique que les horaires deancements sont multi journaliers. La famille TIME doit alors être complété par le mot clé MULW. Valeur par défaut : NOMULT. Dans ce cas, la famille TIME est alors complété par le mot clé DALW pour une tâche planifiée ou PRLW pour une tâche provoquée.

 Pour passer une tâche de multi-journalière à journalière (ou le contraire), il faut tout d'abord changer le type de la tâche (MULT ou NOMULT) dans une première commande, puis définir les horaires (famille **TIME**) dans une seconde commande.

AUTOREL : Relance automatique. Indique que la tâche est à relance automatique. La famille TIME doit alors être complété par le mot clé RELW. Valeur par défaut : NOAUTOREL.

CIBLE : Unités de gestion cibles de l'insertion. Obligatoire. Limité à uxins. Format :

- * : valeur générique acceptée,
- Un code d'Unité de Gestion, si la cible est unique,
- (CODE_UG, CODE_UG, ...) si la cible est multiple, CODE_UG étant un code d'unité de gestion.

Si deux tâches distinctes, par le biais des cibles décrites, donnent lieu à l'insertion d'une même tâche (éventuellement dans la même commande), la deuxième insertion est rejetée, tout comme si la tâche existait déjà (un message d'avertissement est fourni pour chaque rejet). Si l'option REPL est utilisée, la dernière insertion écrase la première.

Si CIBLE=*, les tâches non modèles contenues dans le fichier d'import sont insérées avec leur UG de départ (celle spécifiée lors de l'extraction). L'UG de la tâche contenue dans le fichier d'import est contrôlée dynamiquement et doit exister sur la machine cible. Si elle n'existe pas, la tâche n'est pas insérée. Cette anomalie ne bloque pas la suite du processus d'insertion pour les autres tâches contenue dans le fichier d'import.

Cette fonctionnalité permet de réaliser des opérations de sauvegarde / restauration pour l'ensemble de la planification active, par les 4 commandes ci-dessous (il est recommandé de réaliser une extraction pour les tâches modèles et une autre pour les tâches non modèles) :

```
uxext TSK SES=* UPR=* MU=* OUTPUT=c:\temp\tsk_nomod.ext
uxins TSK SES=* UPR=* MU=* CIBLE=* INPUT=c:\temp\tsk_nomod.ext
uxext TSK SES=* UPR=* MU=* MODEL OUTPUT=c:\temp\tsk_mod.ext
uxins TSK SES=* UPR=* MU=* MODEL CIBLE=* INPUT=c:\temp\tsk_
mod.ext
```

REPL : Option de remplacement. Indique si l'enregistrement extrait (inséré) annule et remplace l'enregistrement déjà existant dans le fichier (ou la tâche) de destination; par défaut, l'extraction (l'insertion) est refusée pour chaque enregistrement déjà existant (signale par un message pour chaque occurrence).

INF : Option d'affichage. Limite l'affichage au nom des objets. Autre valeur : FULL (par défaut). Affiche toutes les informations disponibles sur la liste des objets.

ASTIME ou **MUTIME** : Heure de référence utilisée par la commande. Si MUTIME est utilisé, la colonne MU TIME OFFSET est affichée (uxlst). Si ce flag n'est pas utilisé dans la commande :

- Si le paramètre noeud U_TREF n'est pas défini ou vaut "ASTIME" : l'heure de référence est celle du serveur.
- Si le paramètre noeud U_TREF vaut "MUTIME" : l'heure de référence est celle de l'Unité de Gestion.

DALW : Horaires de lancement journaliers. Doit être associé au mot clé NOMULT. Indique le début de la plage horaire de lancement et sa durée. Une plage standard peut être définie quel que soit le jour de la semaine (mot clé : DALW), des exceptions peuvent être faites pour chaque jour (mots clés : MON, TUE, WED, THU, FRI, SAT, SUN). Format : DALW=(hhmm,hhh,mm) :

- HHMM : heure de début de la plage horaire en heure et minutes,
- HHH : durée de la plage horaire en heures.
- MM : durée de la plage horaire en minutes.

 Le nombre de caractères pour la saisie des heures et des minutes doit être respecté exactement.

MULW : Horaires de lancement multi journaliers. Doit être associé au mot clé MULT. Permet de définir des plages multiples de lancement dans la journée. Format : MULW=((hhmm,hhmm,mmm),hhh,mm). Le module (HHMM,HHMM,MMM) peut être répété autant de fois que nécessaire.

- HHMM,HHMM : heure de début et de fin du calcul en heure et minutes,
- MMM : périodicité du calcul,
- HHH : durée de la plage horaire en heures.
- MM : durée de la plage horaire en minutes.

 Le nombre de caractères pour la saisie des heures et des minutes doit être respecté exactement.

RELW : Horaires de lancement pour une tâche à relance automatique. Doit être associé au mot clé AUTOREL. Permet de définir la fenêtre de validité, le délai avant une relance et le nombre maximum de relances. Format : RELW=((hhmm),hhh,mm,hhh,mm,nnnn,hhh,mm)

- HHMM : heure de début de la fenêtre de validité en heure et minutes
- HHH,MM : durée de la fenêtre de lancement de la tâche en heure et minutes
- HHH,MM : délai à attendre après la fin d'exécution de la tâche pour relancer la tâche, en heures et minutes
- NNNN : nombre maximum de relances pendant la fenêtre de validité. 0000 exprime un nombre illimité
- HHH,MM : durée maximum.

 Le nombre de caractères pour la saisie des heures et des minutes doit être respecté exactement.

PRLW : Horaires de lancement d'une tâche provoquée. Permet de définir une plage horaire de lancement et d'exclusion. Par défaut aucune plage horaire n'est définie, la tâche sera soumise dès sa provocation. Format : PRLW=(hhmm,hhh,mm,hhmm,hhmm) :

- HHMM: heure de début de soumission immédiate, ou mot clé ONDEMAND pour lancer la tâche immédiatement,
- HHH : durée de la plage horaire en heures,
- MM : durée de la plage horaire en minutes,
- HHMM,HHMM : heure de début et de fin de soumission différée.

 Le nombre de caractères pour la saisie des heures et des minutes doit être respecté exactement.

CYCLICAL : Horaires de lancement d'une tâche cyclique. Format : CYCLICAL=((hhhhh,mm),(U_FMT_DATE),(hhmm),(hhh,mm)):

- HHHHH,MM : durée du cycle de planification.
 - U_FMT_DATE : date origine.
-

- HHMM : heure origine.
- HHH,MM : durée de la fenêtre de lancement.

 Le nombre de caractères pour la saisie des heures et des minutes doit être respecté exactement.

RULE : Règle de planification de la tâche. Permet d'associer une ou plusieurs nouvelles règles de planification à la tâche ou d'en supprimer. Format : ((COD_REGLE,A_DATE)(COD_REGLE, A_DATE)...). Le module (COD_REGLE, A_DATE) peut être répété au maximum sept fois dans la même commande.

- COD_REGLE est le code de la règle manipulée. Par défaut, RULE_CODE est une règle implicite. Une règle de planification peut commencer par le caractère "+". Une règle d'exclusion doit commencer par le caractère "-". Si le code de la règle commence par le caractère "+" ou "-", l'utilisateur doit rajouter, au début de RULE_CODE, le caractère "+" pour une règle de planification ou le caractère "-" pour une règle d'exclusion.
- A_DATE est la date d'application de la règle à la tâche au format indiquée par le paramètre U_FMT_DATE.

 Le nombre de caractères A_DATE doit être respecté exactement.

SDATE : Date de première planification. Format : indiqué par la valeur du paramètre U_FMT_DATE. SDATE est maintenu pour compatibilité uniquement, START_VALID et END_VALID doivent être utilisés en remplacement.

START_VALID : début de la plage de validité ou première date de planification. Indique à partir de quelle date la tâche doit être planifiée. Format : indiqué par la valeur du paramètre U_FMT_DATE.

END_VALID : fin de la plage de validité ; aucun lancement ne sera généré après cette date. Format : indiqué par la valeur du paramètre U_FMT_DATE, ou NONE si aucune date de fin n'est prévue.

EDATE : Dates explicites. Permet d'ajouter ou de supprimer une date explicite à la tâche. Format : ((E_DATE,HHMM,HHH,MM,P_DATE)...). Le module (E_DATE,HHMM,HHH,MM,P_DATE) peut être répété autant de fois, dans la même commande, que des dates explicites doivent être ajoutées ou supprimées.

- E_DATE est la date explicite de lancement, au format indiqué par la valeur du paramètre U_FMT_DATE,
- HHMM est l'heure de début de la plage horaire,
- HHH est la durée de la plage horaire en heures,
- MM est la durée de la plage horaire en minutes,
- P_DATE est la date de traitement associée à l'exécution, au format indiqué par la valeur du paramètre U_FMT_DATE.

 Le nombre de caractères doit être respecté exactement.

XDATE : Dates d'interdiction. Permet d'ajouter ou de supprimer une date d'interdiction à la tâche. Format : ((X_DATE)...). où X_DATE : est la date d'interdiction de lancement, au format indiqué par la valeur du paramètre U_FMT_DATE. Le module (X_DATE) peut être répété autant de fois, dans la même commande, que des dates d'interdiction doivent être ajoutées ou supprimées.

 Le nombre de caractères doit être respecté exactement.

EXCLUSION : Fenêtre d'exclusion. Permet d'ajouter ou de supprimer une fenêtre d'exclusion à la tâche. Format : ((U_FMT_DATE,hhmm,U_FMT_DATE,hhmm)(...)). Le module peut être répété autant de fois, dans la même commande, que des fenêtres d'exclusion doivent être ajoutées ou supprimées.

- U_FMT_DATE : est la date de début de la fenêtre d'exclusion.
- HHMM : est l'heure de début de la fenêtre d'exclusion.

- U_FMT_DATE : est la date de fin de la fenêtre d'exclusion.
- HHMM : est l'heure de fin de la fenêtre d'exclusion.

 Le nombre de caractères doit être respecté exactement.

VARNAME : Nom de la variable. Format : 20 caractères alphanumériques maximum. Saisie obligatoire. Ce nom doit être choisi dans la liste des variables de l'Uproc identifiant la tâche.

VVALUE : Valeur de la variable dans la tâche. Le format doit être compatible avec le format déclaré lors de la création de la variable dans l'Uproc, la valeur doit être comprise dans la plage définie pour une variable de type quantité.

- Si la valeur par défaut de la variable de l'Uproc doit être modifiée, la saisie de VVALUE est requise,
- Si la valeur contient des caractères non alphanumériques (espaces, /...) elle doit être encadrée de guillemets.

OUTPUT : Stockage des résultats. Le nom du fichier est obligatoire pour la commande uxext. Si l'item n'est pas renseigné la sortie se fait à l'écran.

INPUT : Fichier d'entrée pour la commande uxins. Le nom du fichier est obligatoire.

TESP : code de l'espace cible du transfert : X pour Exploitation, S pour Simulation ou I pour intégration.

NOVERIF : peut être utilisé avec la commande uxext uniquement. Cette option permet de ne pas vérifier le fichier l'extraction avant d'y écrire de nouveaux enregistrements et d'améliorer ainsi les performances de cette commande. Cet item ne doit être utilisé que pour un nouveau fichier.

3.21.2 Exemples

```
uxadd TSK EXP SES=D_LOAD_BCK UPR=D_LOAD_FIL MU=SIEGE TECHINF type=S  
USER=admin
```

Crée une tâche à partir de la session D_LOAD_BCK, Uproc entête D_LOAD_FIL sur l'unité de gestion SIEGE non modèle dans l'espace d'exploitation. La tâche est planifiée, le compte de soumission est admin. Cette tâche ne peut pas créer de lancement car, dans l'état, aucune date n'a été définie.

```
uxupd TSK EXP SES=D_LOAD_BCK UPR=D_LOAD_FIL MU=SIEGE TECHINF PDATE=( -1, 0, W)
```

Met à jour la tâche créée ci-dessus en indiquant que la calcul de la date de traitement est "moins un jour ouvré" par rapport à la date de premier pilotage du lancement. Les autres paramètres ont conservé leur valeur.

```
uxupd TSK EXP SES=D_LOAD_BCK UPR=D_LOAD_FIL MU=SIEGE TIME DALW=  
(2100, 005, 00) WED=(1900, 005, 00)
```

Met à jour la tâche créée ci-dessus en indiquant des heures de lancement : tous les jours à partir de 21h pendant 5 heures sauf le mercredi à partir de 19h pendant 5 heures.

```
uxupd TSK EXP SES=D_LOAD_BCK UPR=D_LOAD_FIL MU=SIEGE IMPL ADD RULE=  
(( "DJOMOI", 19990501) ("05JM_O", 19990501))
```

Met à jour la tâche créée ci-dessus en indiquant une planification selon les deux règles : "DJOMOI" (dernier jour ouvré du mois) et "05JM_O" (5ème jour du mois report au jour ouvré suivant) avec des dates d'application prévues au 1^{er} mai 1999, dans le cas où le paramètre U_FMT_DATE vaut YYYYMMDD.

```
uxupd TSK EXP SES=D_LOAD_BCK UPR=D_LOAD_FIL MU=HEADQUART IMPL add rule="  
( ("++FWDm", 01/01/2010) )"
```

Met à jour la tâche créée ci-dessus en indiquant une planification utilisant la règle de planification "+FWDM" (premier jour ouvré du mois) avec une date d'application au 1er janvier 2010, U_FMT_DATE ayant pour format JJ/MM/AAAA).

```
uxupd TSK EXP SES=D_LOAD_BCK UPR=D_LOAD_FIL MU=HEADQUART IMPL add rule="
( "-+FWDM", 01/01/2010 ) "
```

Met à jour la tâche créée ci-dessus en indiquant une planification utilisant une règle d'exclusion "-+FWDM" (premier jour ouvré du mois) avec une date d'application au 1er janvier 2010, U_FMT_DATE ayant pour format JJ/MM/AAAA).

```
uxupd TSK EXP SES=D_LOAD_BCK UPR=D_LOAD_FIL MU=SIEGE EXPLI ADD EDATE=
( (19991230, 2100, 005, 00, 19991231) )
```

Met à jour la tâche créée ci-dessus en créant une date explicite pour le 1^{er} janvier 1997 à 21 heures pendant 5 heures avec la date de traitement du 2 janvier 97, dans le cas où le paramètre U_FMT_DATE vaut YYYYMMDD.

```
uxupd TSK EXP SES=D_LOAD_BCK UPR=D_LOAD_FIL MU=SIEGE EXCLU ADD XDATE=
( (19991231) )
```

Met à jour la tâche créée ci-dessus en créant une date d'interdiction pour le 31 décembre 1996, dans le cas où le paramètre U_FMT_DATE vaut YYYYMMDD.

```
uxadd TSK EXP UPR=D_LOAD_FIL MU=SIEGE TECHINF type=P USER=admin
```

Crée une tâche provoquée de l'Uproc D_LOAD_FIL sur l'unité de gestion SIEGE non modèle dans l'espace d'exploitation. Le compte de soumission est admin.

```
uxupd TSK EXP UPR= D_LOAD_FIL MU=HEADQUART TIME PRLW="
(ONDEMAND, 000, 00, 0000, 0000) "
```

Lance la tâche immédiatement.

```
uxlst TSK EXP UPR=D_* FULL
```

Liste les tâches qui viennent d'être créées ainsi que leurs caractéristiques.

```
uxupd TSK UPR=D_LOAD_FIL MU=SIEGE LAUNCHVAR ADD VARNAME=VARTXT VVALUE=XY_
VARTXT_VALTSK
```

Modifie la valeur de la variable VARTXT définie pour D_LOAD_FIL en donnant une valeur par défaut de «XY_VARTXT_VALTSK» pour cette variable dans la tâche.

```
uxupd TSK UPR=D_LOAD_FIL MU=SIEGE LAUNCHVAR ADD VARNAME=VARDATE
VVALUE=2000/01/01
```

Modifie la valeur de la variable VARDATE définie pour D_LOAD_FIL en donnant une valeur par défaut de "2000/01/01" pour cette variable dans la tâche.

```
uxext TSK EXP MU=MOD UPR=D_* MODEL output=c:\temp\DOC.ext
```

Extrait toutes les tâches modèles "MOD" de l'espace exploitation dont l'Uproc commence par D_ et archive les données dans le fichier c:\temp\DOC.ext.

```
uxins TSK EXP MU="*" UPR=* CIBLE=SIEGE input=c:\temp\DOC.ext
```

Insère toutes les tâches (non modèles par défaut) du fichier c:\temp\DOC.ext, comme tâches de l'unité de gestion SIEGE dans l'environnement courant (société, nœud) en espace exploitation sauf si elles

Insère toutes les tâches du fichier c:\temp\DOC.ext, comme tâches modèles dans l'environnement courant (société, nœud) en espace exploitation sauf si elles existent déjà.

```
uxadd upr exp upr=UPR_1 appl=U_ fperiod=d
uxadd ses exp ses=SES_1 header=UPR_1
uxadd ses exp ses=SES_2 header=UPR_1
uxadd MU MU=MU_1 TNODE=%S_NODENAME%
uxadd MU MU=MU_2 TNODE=%S_NODENAME%
uxadd USER USER=USER_1
uxadd tsk exp tsk=parent_task ses=SES_1 upr=UPR_1 MU=MU_1 TECHINF TYPE=o
USER=USER_1
uxadd tsk exp tsk=child_task ses=SES_2 upr=UPR_1 MU=MU_1 TECHINF TYPE=o
USER=USER_1
uxupd tsk exp tsk=child_task MU=MU_1 PARENT PARENT_TSK=parent_task PARENT_
MU=MU_2
```

Cet exemple montre la création de deux tâches spécifiques liées à une tâche principale.

3.22 Les déclencheurs

uxadd TRIG : ajoute un déclencheur dans la liste des déclencheurs avec un événement de type générique. Les événements de type générique ou `_SAPJOB` sont exclusifs et ont des propriétés différentes.

Item	Format	Défaut
TRIG=	64c	
TYPE=	64c	
ENABLE DISABLE		ENABLE
TRANSMITTED NOTTRANSMITTED		TRANSMITTED
LABEL=	64c	
APP INT SIM EXP		S_ESPEXE

Exemple :

```
uxadd TRIG TRIG=TRIG_1 TYPE=WINDOWS_EVENT LABEL=TRIGGER_WINDOWS
```

uxadd TRIG : ajoute un déclencheur dans la liste des déclencheurs avec un événement de type `_SAPJOB`.

Item	Format	Défaut
TRIG=	64c	
TYPE=_SAPJOB		
ENABLE DISABLE		ENABLE
TRANSMITTED NOTTRANSMITTED		TRANSMITTED
LABEL=	64c	
APP INT SIM EXP		S_ESPEXE

Exemple :

```
uxadd TRIG TRIG=SAP_trig TYPE=_SAPJOB ENABLE EXP
```

uxupd TRIG : modifie les caractéristiques d'un déclencheur.

Item	Format	Défaut
TRIG=	64c	
TYPE=	64c	
LABEL=	64c	
ENABLE DISABLE		
TRANSMITTED NOTTRANSMITTED		

Exemple :

```
uxupd TRIG TRIG=TRIG_1 LABEL=TRIGGER_WINDOWS DISABLE
```

Famille	Description
ACTION	Spécifications pour lancer le déclencheur (tâche, UG, durée, modèle ou non modèle)
CONDITIONS	Conditions de filtre des propriétés des événements

ACTION : détermine les spécifications pour lancer la tâche associée au déclencheur

Item	Format	Défaut
ACTION	64c	
TSK	64c	
MU	64c	
MODEL NOMODEL		NOMODEL
DURATION	HHHMM	

Exemple :

```
uxupd TRIG TRIG=TRIG_33 ACTION TSK=TASK_33 MU=calpm001 DURATION=00045
```

CONDITIONS add : ajoute une condition basée sur les propriétés de l'événement. Cette commande ne peut pas être utilisée pour un déclencheur de type _SAPJOB.

Item	Format	Default
NAME=	64c	
VALUE=	64c	

Exemples :

```
uxupd TRIG TRIG=TRIG_001 CONDITIONS add NAME=FILE_RESOURCE VALUE=FILE01
uxupd TRIG TRIG=TRIG_WINDOWS CONDITIONS add NAME=FILE_SIZE VALUE=2KB
```

CONDITIONS upd : modifie une condition d'un événement générique.

Item	Format	Default
NAME=	64c	
VALUE=	64c	

Exemple :

```
uxupd TRIG TRIG=TRIG01 CONDITIONS upd NAME=FILE VALUE=FILE01
```

CONDITIONS upd : modifie une condition d'un événement de type _SAPJOB

Item	Format	Default
NAME=	SAPJOB_NAME SAPJOB_STATUS SAPJOB_GROUP SAP_JOBUSER	*
VALUE=	64c	

Exemple :

```
uxupd TRIG TRIG=trig_2 CONDITIONS upd NAME=SAPJOB_NAME VALUE=2
```

 Lorsqu'un déclencheur de type _SAPJOB est créé, les conditions ne peuvent qu'être modifiées. Elles ne peuvent être ni créées ni supprimées.

CONDITIONS dlt : supprime une condition d'un déclencheur. Cette commande ne peut pas être utilisée pour un déclencheur de type _SAPJOB.

Item	Format	Default
NAME=	64c	
VALUE=	64c	

Exemple :

```
uxupd TRIG TRIG=TRIG22 CONDITIONS dlt NAME=FILE_RESOURCE
```

uxdup TRIG : duplique un déclencheur existant.

Item	Format	Défaut
TRIG=	64c	
TTRIG=	64c	
TTYPE=	64c	
TTSK=	64c	
TLABEL=	64c	
APP INT SIM EXP		S_ESPEXE

Exemple :

```
uxdup TRIG trig=TEST_11 TTRIG=TEST_22 TLABEL=duplicate
```

uxshw TRIG : affiche les caractéristiques d'un déclencheur.

Item	Format	Défaut
TRIG=	64c	
TYPE=	64c	
OUTPUT=filename		
APP INT SIM EXP		S_ESPEXE

Exemple :

```
uxshw TRIG TRIG=TRIG_01
```

uxdlt TRIG : supprime un déclencheur.

Item	Format	Défaut
TRIG=	64c	
EXP SIM INT APP		S_ESPEXE

Exemple :

```
uxdl1 TRIG TRIG=TRIG_01
```

uxlst TRIG: liste les déclencheurs.

Item	Format	Défaut
TRIG=	64c	
TYPE=	64c _SAPJOB	
EXP SIM INT APP		S_ESPEXE
OUTPUT=filename		
INF FULL		INF

Exemple :

```
uxlst TRIG FULL OUTPUT="C:\temp\lsttrigger.ext"
```

uxext TRIG : extrait la définition complète d'un déclencheur.

Item	Format	Défaut
TRIG=	64c	
TYPE=	64c _SAPJOB	
EXP SIM INT APP		S_ESPEXE
OUTPUT=filename		

Exemple :

```
uxext TRIG TRIG=TRIG_01 OUTPUT="C:\temp\trigger.txt"
```

uxins TRIG : insère la définition complète d'un déclencheur dans les fichiers correspondants de Dollar Universe.

Item	Format	Défaut
TRIG=	64c	
TYPE=	64c _SAPJOB	
EXP SIM INT APP		S_ESPEXE
INPUT=filename		

Exemple:

```
uxins TRIG TRIG=TRIG01 INPUT="C:\temp\trigger.txt"
```

3.22.1 Description des items

TRIG : nom du déclencheur. Format : 64 caractères alphanumériques

TTRIG : nom du déclencheur, résultat d'une duplication. Format : 64 caractères alphanumériques.

LABEL : libellé descriptif du déclencheur. Format : 64 caractères alphanumériques.

TLABEL : libellé descriptif du déclencheur résultat de la duplication. Format : 64 caractères alphanumériques.

TYPE : type d'événement. Format : 64 caractères alphanumériques (ex. LOG_EVENT, SYSTEM, SYSLOAD), toujours stocké en majuscules dans la BD. Les types livrés commençant par un caractère "_" sont réservés et ont des propriétés et des conditions prédéfinies. Le type livré actuellement est : _SAPJOB

NAME : nom de la propriété d'un événement. Format 64 caractères alphanumériques. Les propriétés d'un événement de type _SAPJOB sont prédéfinies :

- SAPJOB_NAME : nom du job SAP, format 64 caractères, vide par défaut.
- SAPJOB_STATUS : AFRSP par défaut (Aborted, finished, running, scheduled ou preliminary). Dans le cas de la duplication d'un statut, toutes les occurrences après le premier sont ignorées. Par exemple : "name=SAPJOB_STATUS value=AFRPSF" est équivalent à "name=SAPJOB_STATUS value=AFRSP".

Tableau de correspondance entre les différentes valeurs de statut :

Code	Interface client SAP	Commandes DUAS
A	Canceled	Aborted
F	Finished	Finished
R	Active	Running
S	Scheduled	Preliminary
P	Released	Scheduled
-	Ready	-

- SAPJOB_GROUP : groupe de l'utilisateur, * par défaut
- SAPJOB_USER : utilisateur SAP, * par défaut

VALUE : valeur attendue de la propriété de l'événement

TSK : code de la tâche. Format: 64 caractères alphanumériques. Si le code de la tâche est vide, la valeur par défaut est : TSK(SSS)(VVV)(UUU)(VVV) avec :

- (SSS)(VVV): code et version de la Session
- (UUU)(VVV): code et version de l'Uproc

MU : code de l'Unité de Gestion. Format : 64 caractères alphanumériques.

ENABLE : active le déclencheur. **DISABLE** : désactive le déclencheur

DURATION : durée de la fenêtre de lancement de la tâche. Format : HHHMM

INPUT : nom du fichier pour une commande uxins. Le nom du fichier est obligatoire.

OUTPUT : stockage des résultats. Le nom du fichier est obligatoire pour la commande uxext. Si l'item n'est pas renseigné la sortie se fait à l'écran.

TRANSMITTED | NOTTRANSMITTED : détermine si les conditions des propriétés de l'événement sont appliquées à la tâche du déclencheur avec les valeurs associées.

FULL : affiche le nom de la tâche et de l'Unité de gestion liées au déclencheur. Autre valeur (par défaut) **INF** : affiche uniquement l'entête de la tâche.

 **FULL** et **INF** sont incompatibles entre eux.

NOMODEL : nature de la tâche. Indique que la tâche sélectionnée est de nature non modèle. **MODEL** indique que la tâche sélectionnée est de nature modèle.

3.22.2 Exemples

```
uxadd TRIG TRIG=TRIG_WINDOWS TYPE=WINDOWS_EVENT LABEL=WINDOWS_EVENT_TRIGGER
ENABLE TRANSMITTED
```

Crée un déclencheur basé sur un type d'événement **WINDOWS_EVENT**. Le déclencheur est activé mais n'a pas de tâche associée.

```
uxupd TRIG TRIG=WINDOWS_EVENT ACTION TSK=TASK_1 MU=CALPM002 NOMODEL
DURATION=00330
```

Ajoute une tâche au déclencheur créé ci-dessus sur l'Unité de gestion **CALPM002**, de nature non modèle et de fenêtre de lancement de 3 heures et 30 minutes.

```
uxupd TRIG TRIG=WINDOWS_EVENT DISABLE
```

Modifie le déclencheur créé et le désactive. Si le type d'événement spécifié est reçu, le déclencheur ne lancera pas la tâche associée.

```
uxupd TRIG TRIG=WINDOWS_EVENT CONDITIONS add NAME=FILE_NAME
VALUE=REPORT.ext
```

Modifie le déclencheur créé ci-dessus en ajoutant la condition **FILE_NAME** avec la valeur **REPORT.ext**. Si ce fichier est trouvé, le déclencheur lancera la tâche associée.

```
uxupd TRIG TRIG=WINDOWS_EVENT CONDITIONS upd NAME=FILE_NAME
VALUE=ACCOUNT.ext
```

Modifie le déclencheur créé ci-dessus en modifiant la valeur de la condition **FILE_NAME** à **ACCOUNT.ext**.

```
uxadd TRIG TRIG=TRIG_SAP TYPE=_SAPJOB LABEL=SAP_JOB_TRIGGER TRANSMITTED
```

Crée un déclencheur de type **_SAPJOB** qui transmettra les propriétés de l'événement si la tâche associée existe.

```
uxupd TRIG TRIG=SAPJOB CONDITIONS upd SAPJOB_NAME=NEW_JOB
```

Modifie la condition **SAPJOB_NAME** du déclencheur **SAPJOB** de * (par défaut) à **NEW_JOB**.

```
uxext TRIG EXP TRIG=MAIL* output=c:\temp\user\FILE.ext
```

Extrait tous les déclencheurs de l'Espace Exploitation dont le nom commence par "MAIL" et les archive dans le fichier **c:\temp\user\FILE.ext**.

```
uxins TRIG EXP input=c:\temp\user\FILE.ext
```

Insère tous les déclencheurs du fichier **c:\temp\FILE.ext** dans l'environnement courant de Dollar Universe.

3.23 Les dossiers d'exploitation

uxadd BOOK : crée un nouveau dossier d'exploitation.

Item	Format	Défaut
BOOK=	64c	

Item	Format	Défaut
AREA=A I S X	4c	
DOM=	1c *	
APPL=	64c *	
TSK=	64c *	
MU=	64c *	
UPR=	64c *	
SES=	64c *	
USER=	64c *	
QUEUE=	31c *	
INFO=	50c *	
SEV=	4n *	
TYPE=BK_INT BK_URL BK_FILE_TXT BK_FILE_BIN	12c	BK_INT
FILE=	256c	
URL=	256c	
TEXT=	1024c	“ ”

Exemple :

```
uxadd BOOK book=RUNB00K1 area=X upr=TEST type=BK_INT text=\"Dossier
exploitation\"
```

uxupd BOOK : modifie un dossier d'exploitation.

Item	Format	Défaut
BOOK=	64c	
AREA= A I S X	4c *	
DOM=	1c *	
APPL=	64c *	
TSK=	64c *	
MU=	64c *	
UPR=	64c *	
SES=	64c *	
USER=	64c *	
QUEUE=	31c *	
INFO=	50c *	
SEV=	4n *	
TYPE=BK_INT BK_URL BK_FILE_TXT BK_FILE_BIN	12c	
FILE=	256c	
URL=	256c	
TEXT=	1024c	

Exemple :

```
uxupd book book=RUNBOOK1 type=BK_URL URL="http://www.Automic.com/fr/"
```

uxshw BOOK : affiche les caractéristiques d'un dossier d'exploitation.

Item	Format	Défaut
BOOK=	64c *	
TYPE=BK_INT BK_URL BK_FILE_TXT BK_FILE_BIN	12c	*
AREA= A I S X	4c *	*
DOM=	1c *	*
APPL=	64c *	*
TSK=	64c *	*
MU=	64c *	*
UPR=	64c *	*
SES=	64c *	*
USER=	64c *	*
QUEUE=	31c *	*
INFO=	50c *	*
SEV=	4n *	*
OUTPUT=file_name		

Exemple :

```
uxshw BOOK BOOK=RUN*
```

uxlst BOOK : liste les dossiers d'exploitation.

Item	Format	Défaut
BOOK=	64c	*
TYPE=BK_INT BK_URL BK_FILE_TXT BK_FILE_BIN	12c	*
AREA=	3c	*
DOM=	1c	*
APPL=	64c	*
TSK=	64c	*
MU=	64c	*
UPR=	64c	*
SES=	64c	*
USER=	64c	*
QUEUE=	31c	*
INFO=	50c	*
SEV=	4n	*
FULL INF		FULL
OUTPUT=file_name		

Exemple :

```
uxlst BOOK FULL OUTPUT=c:\temp\LstBook.txt
```

uxdlit BOOK : supprime un dossier d'exploitation.

Item	Format	Défaut
BOOK=	64c	

Exemple :

```
uxdlit BOOK BOOK=RUNBOOK1
```

uxext BOOK : extrait la définition complète d'un dossier d'exploitation

Item	Format	Défaut
BOOK=*	64c *	
REPL		
OUTPUT=file_name		

Exemple :

```
uxext BOOK BOOK=D* OUTPUT="c:\temp\Book.ext"
```

uxins BOOK : insère la définition complète d'un dossier d'exploitation.

Item	Format	Défaut
BOOK=*	64c *	
REPL		
INPUT=file_name		

Exemple :

```
uxins BOOK BOOK=* INPUT="c:\temp\Book.ext"
```

3.23.1 Description des items

BOOK : saisissez le nom du dossier d'exploitation. Format 64 caractères alphanumériques.

Les informations ci-dessous permettent de sélectionner les exécutions qui afficheront le dossier d'exploitation.

AREA : espace(s) dans le(s)quel(s) le dossier d'exploitation doit être utilisé. Saisissez de 1 à 4 caractères parmi A, I, S, X

- A : Espace Application
- I : Espace Intégration
- S : Espace Simulation
- X : Espace Exploitation

DOM : saisissez le code du domaine du job (1 caractère alphanumérique) pour lequel ce dossier d'exploitation doit être utilisé ou *.

APPL : saisissez le code de l'application du job (64 caractères alphanumériques) pour laquelle ce dossier d'exploitation doit être utilisé ou *.

TSK : saisissez le code de la Tâche du job (64 caractères alphanumériques) pour laquelle ce dossier d'exploitation doit être utilisé ou *.

MU : saisissez le code de l'Unité de Gestion du job (64 caractères alphanumériques) pour laquelle ce dossier d'exploitation doit être utilisé ou *.

SES : saisissez le code de la session du job (64 caractères alphanumériques) pour laquelle ce dossier d'exploitation doit être utilisé ou *.

UPR : saisissez le code de l'Uproc du job (64 caractères alphanumériques) pour laquelle ce dossier d'exploitation doit être utilisé ou *.

USER : saisissez le code du compte de soumission du job (64 caractères alphanumériques) pour lequel ce dossier d'exploitation doit être utilisé ou *.

QUEUE : saisissez le code de la queue DQM du job (31 caractères alphanumériques) pour laquelle ce dossier d'exploitation doit être utilisé ou *.

INFO : saisissez la valeur du champ Information du job (50 caractères alphanumériques) pour laquelle ce dossier d'exploitation doit être utilisé ou *.

SEV : saisissez la valeur de la Sévérité du job (4 caractères numériques) pour laquelle ce dossier d'exploitation doit être utilisé ou *.

TYPE : type du dossier d'exploitation. Format 12 caractères alphanumériques). Saisissez l'une des quatre valeurs ci-dessous :

- **BK_INT** : Dossier d'exploitation interne (valeur par défaut pour uxadd). Complétez cette information par l'item TEXT ci-dessous.
TEXT : saisissez les instructions qui apparaîtront lorsque le dossier d'exploitation sera affiché.
- **BK_URL** : URL externe. Complétez obligatoirement cette information par l'item URL ci-dessous.
URL : saisissez le nom de l'URL à ouvrir entre guillemets.
- **BK_FILE_TEXT** : Fichier texte externe. Complétez obligatoirement cette information par l'item FILE ci-dessous.
- **BK_FILE_BIN** : Fichier exécutable externe. Complétez obligatoirement cette information par l'item FILE ci-dessous.
FILE : saisissez le nom du fichier à ouvrir ou à exécuter. La taille maximum d'un fichier attaché est de 10 Mo. Lorsque le fichier est attaché et le dossier d'exploitation enregistré, le fichier est copié dans le sous-répertoire data/runbooks de Dollar Universe.

INF : Option d'affichage. Limite l'affichage au nom des objets. Autre valeur : FULL (par défaut). Affiche toutes les informations disponibles sur la liste des objets.

REPL : option de remplacement. Indique si l'enregistrement extrait (inséré) annule et remplace l'enregistrement déjà existant dans le fichier de destination; par défaut, l'extraction (l'insertion) est refusée pour chaque enregistrement déjà existant (signalé par un message pour chaque occurrence).

INPUT : Fichier d'entrée pour la commande uxins. Le nom du fichier est obligatoire.

OUTPUT : Stockage des résultats. Le nom du fichier est obligatoire pour la commande uxext. Si l'item n'est pas renseigné la sortie se fait à l'écran.

3.23.2 Exemples

```
uxadd BOOK book=TEST appl=COMPTA type=BK_FILE_TEXT FILE=c:\temp\text.txt
```

Crée un dossier d'exploitation (pour tous les espaces par défaut) nommé TEST pour tous les jobs de l'application COMPTA. Ce dossier d'exploitation sera accessible dans un fichier externe : c:\temp\text.txt.

4 Commandes d'exploitation

Ce chapitre présente les commandes utilisées pour contrôler la production.

4.1 Les ressources

La ressource doit avoir fait l'objet d'un pilotage avec des quotas avant de pouvoir exécuter l'une de ces commandes.

uxhld RES : alloue tout ou partie des quotas d'une ressource.

-  Si la ressource est variabilisée par les éléments !SOC!, !ESP!, !UG! (ou !MU!) ou !DTRAIT!, l'allocation des quotas sera réalisée en tenant compte de la valeur de ces éléments dans la commande. Si la ressource n'est pas variabilisée par les éléments !SOC!, !ESP!, !UG! (ou !MU!) ou !DTRAIT!, une quelconque valeur de ces éléments dans la commande ne sera pas prise en compte.

Item	Format	Défaut
RES=	64c *	
LOCK QT1= & QT2=		
QT1=	4n	0
QT2=	4n	0
MU=	64c * " "	*
PDATE=	AAAAMMJJ * " "	*
ESP=	X S I A * " "	*
TSOC=	6c * " "	*

Exemple :

```
uxhld RES RES=D_BASE QT1=4
```

uxrls RES : libère tout ou partie des quotas d'une ressource.

Item	Format	Défaut
RES=	64c *	
CLEAR QT1= & QT2=		
QT1=	4n	0
QT2=	4n	0
MU=	64c * " "	*
PDATE=	AAAMMJJ * " "	*
ESP=	X S I A * " "	*
TSOC=	6c * " "	*

 La commande **uxrls res** étant susceptible de traiter l'ensemble des espaces, les tentatives de connexion à des serveurs d'IO arrêtés peut faire perdre du temps. Le positionnement du paramètre noeud U_RES_AVAIL_AREAS (catégorie Réglages avancés), avec le code des espaces effectivement actifs permet d'optimiser les performances. Reportez-vous au Manuel Utilisateur d'UniViewer.

La commande OS/400 UXRLSRES permet de libérer une ressource à partir de la ligne de commande. F4 permet de saisir les paramètres unitairement. F11 affiche le nom de chaque paramètre.

Exemple :

```
uxrls RES RES=D_BASE CLEAR
```

uxshw RES : affiche la définition d'une ressource et l'allocation de ses quotas.

Item	Format	Défaut
RES=	64c *	
TSOC=	6c * " "	
MU=	64c * " "	
PDATE=	AAAAMMJJ * " "	
ESP=	X S I A * " "	
INF FULL RSV		INF
OUTPUT=file name		

Exemple :

```
uxshw RES RES=D_*
```

uxrsv RES : réserve les quotas d'une ressource afin de définir une priorité d'ordonnancement.

Item	Format	Défaut
RES=	64c *	
EXP SIM INT APP	3c	S_ESPEXE
TSK=	64c	
UPR=	64c	
VUPR=	3n	
SES=	64c	
VSES=	3n	
MU=	64c	
QT1=	4n	
QT2=	4n	
ESP=	X S I A * " "	
TSOC=	6c * " "	
PDATE=	AAAAMMJJ * " "	

Exemple :

```
uxrsv RES RES=D_FIL_PAY ESP=X UPR=TEST_ECA MU=$S_NODENAME QT1=10 QT2=0
PDATE=$S_DATRAIT
```

⚠ Attention : la variable S_DATRAIT est définie en interne dans le script de l'Uproc.

uxcnr RES : supprime la réservation des quotas d'une ressource faite par la commande uxrsv.

Item	Format	Défaut
RES=	64c *	
EXP SIM INT APP	3c	S_ESPEXE
TSK=	64c	
UPR=	64c	
VUPR=	3n	
SES=	64c	
VSES=	3n	
MU=	64c	
TSOC=	6c * " "	
ESP=	X S I A * " "	
PDATE=	AAAMMJ * " "	

Exemple :

```
uxcnr RES RES=D_FIL_PAY ESP=X UPR=TEST_ECA MU=$S_NODENAME PDATE=$S_DATRAIT
```

⚠ Attention : la variable S_DATRAIT est définie en interne dans le script de l'Uproc.

uxpre RES : supprime des événements ressource dont les quotas ont été libérés (la ressource peut être variabilisée avec !UG!, !PDATE!, !SOC! ou !ESP!).

Item	Format	Défaut
RES=	64c *	
EXP SIM INT APP	3c	S_ESPEXE
MU=	64c *	
PDATE=	AAAMMJ * " "	
TSOC=	6c * " "	
ESP=	X S I A * " "	

4.1.1 Description des items

RES : Référence ressource. Format : 64 caractères alphanumériques. Requiert la saisie de la référence de la ressource (valeur générique acceptée pour uxshw avec l'option FULL).

TSOC : Société. Format : 6 caractères alphanumériques. La saisie du caractère "*" permet la sélection de toutes les ressources faisant référence à une société, la saisie du caractère " " permet la sélection de toutes les ressources ne faisant pas référence à une société. Permet la sélection des ressources faisant référence à une société dans le nom du fichier ou sa localisation.

ESP : Espace. Format : 1 caractère : "X", "S", "I", "A", "*" ou " ". La saisie du caractère "*" permet la sélection de toutes les ressources faisant référence à un espace, la saisie du caractère " " permet la sélection de toutes les ressources ne faisant pas référence à un espace. Permet la sélection des ressources faisant référence à un espace dans le nom du fichier ou sa localisation. Pour les commandes

uxrsv et uxcnr, permet de spécifier dans quel espace se trouve la tâche ayant fait la réservation, l'espace décrit par la variable S_ESPEXE est pris par défaut.

TSK : Code de la Tâche qui réserve ou libère la ressource. Format : 64 caractères alphanumériques. Obligatoire si UPR est absent.

UPR : Code Uproc. Commandes uxrsv et uxcnr uniquement. Format : 64 caractères alphanumériques. Requiert la saisie d'un code Uproc, identifiant de la tâche qui réserve ou libère la ressource (valeur générique acceptée sauf pour uxrsv). Obligatoire si TSK est absent.

VUPR : Version d'Uproc. Commandes uxrsv et uxcnr uniquement. Format : 3 caractères numériques. Permet la saisie d'une version particulière d'une Uproc. Par défaut, toutes les versions des Uprocs renseignées sont considérées (inutile avec SIM et EXP).

SES : Code session. Commandes uxrsv et uxcnr uniquement. Format : 64 caractères alphanumériques. Requiert la saisie d'un code session, identifiant de la tâche qui réserve ou libère la ressource (valeur générique acceptée sauf pour uxrsv).

VSES : Version de session. Commandes uxrsv et uxcnr uniquement. Format : 3 caractères numériques. Permet la saisie d'une version particulière d'une session. Par défaut, toutes les versions des sessions renseignées sont considérées (inutile avec SIM et EXP).

MU : Unité de gestion. Format : 64 caractères alphanumériques. Permet la sélection des ressources faisant référence à une unité de gestion dans le nom du fichier ou sa localisation. La saisie du caractère "*" permet la sélection de toutes les ressources faisant référence à une UG, la saisie du caractère " " permet la sélection de toutes les ressources ne faisant pas référence à une UG. Pour les commandes uxrsv et uxcnr, permet d'identifier la tâche qui réserve ou libère la ressource.

PDATE : Date de traitement. Permet la sélection des ressources faisant référence à une date de traitement dans le nom du fichier ou sa localisation. Format : AAAAMMJJ. La saisie du caractère "*" permet la sélection de toutes les ressources faisant référence à une date de traitement, la saisie du caractère " " permet la sélection de toutes les ressources ne faisant pas référence à une date de traitement. Pour les commandes uxrsv et uxcnr, permet d'identifier la tâche qui réserve ou libère la ressource.

QT1 ou **QT2** : Quotas d'allocation de niveau 1 et 2 respectivement. Format : de 0 à 9999.

- Dans la commande uxhld, ces quotas permettent d'allouer les valeurs indiquées, lors de l'exécution de la commande, dans la mesure où ils sont disponibles. L'utilisation de QT1 ou QT2 est incompatible avec l'utilisation de LOCK.
- Dans la commande uxrsv, ces quotas permettent de réserver, lors de l'exécution de la commande, pour la tâche spécifiée, les valeurs indiquées dans la mesure où ces quotas sont disponibles.
- Dans la commande uxrls res (ou UXRLSRES sous OS/400), ces quotas permettent de libérer les valeurs de quotas indiquées lors de l'exécution de la commande. L'utilisation de QT1 ou QT2 est incompatible avec l'utilisation de CLEAR.

LOCK : Allocation. Permet de réserver en totalité la ressource sélectionnée. Utilisé dans uxhld, la ressource est totalement réservée. L'usage de LOCK est incompatible avec QT1 ou QT2.

CLEAR : Libération. Permet de libérer la ressource sélectionnée. Utilisé dans uxrls res (ou UXRLSRES sous OS/400), la ressource est totalement libérée sans distinction de l'origine de l'allocation par l'exécution de la commande. L'usage de CLEAR est incompatible avec QT1 ou QT2.

FULL : Permet l'affichage de la totalité des informations sur l'allocation d'une ressource : Uproc origine de l'allocation, quotas alloués par une Uproc, quotas alloués par les commandes, quotas disponibles.

- Autre valeur RSV : permet l'affichage des réservations de la ressource.
- Autre valeur INF : se reporter au chapitre "Les commandes de paramétrage" paragraphe "[Les ressources](#)".

 Affichage de l'allocation. INF, FULL et RSV sont incompatibles entre eux.

OUTPUT : Stockage des résultats. Si l'item n'est pas renseigné la sortie se fait à l'écran. La description de cet item est fournie dans le dictionnaire d'items.

4.1.2 Exemples

```
uxhld RES RES=D_BASE QT1=4
```

Alloue un quota de niveau 1 pour une valeur de 4 sur la ressource "D_BASE" dans l'environnement courant (société, nœud).

```
uxshw RES RES=D_BASE FULL
```

Affiche les allocations de la ressource D_BASE dans l'environnement courant (société, nœud). Soit par exemple :

Dde Ress: D_BASE * *							
Utilisation des quotas de la ressource							
Ressource: D_BASE				Ress: Exec:			
quotas maxi	Alloc.	Auto.	Alloc.	Manu.	Residuel		
0010	0010	0000	0000	0004	0000	0006	0010

- quotas maxi : affiche les valeurs maximum des quotas 1 et 2,
- Alloc. Auto. : affiche les valeurs des quotas 1 et 2 alloués par des Uprocs en cours d'exécution,
- Alloc. Manu. : affiche les valeurs des quotas 1 et 2 alloués par des Uprocs terminées ou par des commandes uxhld RES,
- Résiduel : affiche les valeurs des quotas 1 et 2 restantes pour des allocations futures.

Si la même commande est exécutée pendant l'exécution de l'Uproc D_LOAD_FIL (qui a une condition de ressource sur la totalité de D_BASE) sur l'unité de gestion et l'espace d'exécution, on obtient alors le résultat suivant :

Dde Ress: D_BASE * *							
Utilisation des quotas de la ressource							
Ressource: D_BASE				Ress: Exec:			
quotas maxi	Alloc.	Auto.	Alloc.	Manu.	Residuel		
0010	0010	0000	0000	0004	0000	0006	0010

Utilisation des quotas de la ressource							
Ressource: D_BASE X SIEGE				Ress: Exec: EXCL			
session	Uproc	Es UG	Date Trt	Quo1	Quo2	X Numproc	
D_LOAD_BCK	D_LOAD_FIL	X SIEGE		0010	0010	0	0000259
quotas maxi	Alloc.	Auto.	Alloc.	Manu.	Residuel		
0010	0010	0010	0010	0000	0000	0000	0000

La ressource D_BASE est décomposée en fait en 2 ressources (symbolisé dans l'exemple par les deux cadres) en fonction de son environnement d'exécution. La première allocation résulte de l'exécution de la commande uxhld, la seconde allocation résulte de l'exécution de l'Uproc D_LOAD_FIL.

- Ressource : indique la référence de la ressource suivi éventuellement de l'espace, la date de traitement, l'unité de gestion ou la société si les mots clés correspondants sont intégrés au nom de la ressource. C'est le cas ici pour D_BASE dont le nom est défini par : !UG!!ESP!.
- Exec : EXCL indique que la ressource a été attribuée en exclusive à l'exécution de l'Uproc.
- Session Uproc Es UG Date Trt Quo1 Quo2 X Numproc : indiquent le contexte dans lequel la ressource a été attribuée. Es pour espace, Date Trt pour Date de traitement, Quo pour quota.

```
uxrls RES RES=D_BASE ESP=* MU=* CLEAR
```

Libère tous les quotas de la ressource D_BASE allouée au titre de toutes les unités de gestion et espaces.

```
uxrsv RES RES=PRIO_ORDO ESP=X UPR=IU_DT1 MU=S01 QT1=90 QT2=0 PDATE=${S_
DATRAIT}
```

Réserve, pour la tâche définie par l'Uproc IU_DT1, l'unité de gestion S01 dans l'espace exploitation, pour la date de traitement courante, la ressource de référence PRIO_ORDO avec un quota QT1 de 90.

Si la ressource dispose d'un quota QT1 maximum de 100, 10 restent libres à l'issue de l'exécution de la commande. Une autre tâche, n'ayant pas effectué de réservation sur cette ressource, pourra s'exécuter si sa condition sur cette ressource demande un quota inférieur à 10.

```
uxrsv RES RES=PRIO_ORDO ESP=X UPR=IU_DT2 MU=S01 QT1=10 QT2=0 PDATE=$D_
DATRAIT
```

Ces deux dernières commandes définissent une priorité de pilotage par Dollar Universe, c'est la tâche qui a réservé le plus de quotas qui s'exécutera en premier, dans cet exemple : celle définie sur l'Uproc IU_DT1.

```
uxcnr RES RES=PRIO_ORDO ESP=X UPR=IU_DT2 MU=S01 PDATE=$D_DATRAIT
```

Libère la réservation faite sur la ressource PRIO_ORDO par la tâche définie par l'Uproc IU_DT2, l'unité de gestion S01 dans l'espace exploitation.

4.2 Les tâches

uxhld TSK : suspend une tâche.

uxrls TSK : réactive une tâche.

Item	Format	Défaut
EXP SIM INT APP	3c	S_ESPEXE
TSK=	64c *	
SES=	64c *	" "
VSES=	3n *	
UPR=	64c *	
VUPR=	3n *	
MU=	64c *	
TTYPE=S O P C *	1c *	*
MODEL NOMODEL		NOMODEL

Exemple :

```
uxhld TSK EXP SES=D_LOAD_BCK UPR=D_LOAD_FIL MU=HEADQUART
uxrls TSK EXP SES=D_LOAD_BCK UPR=D_LOAD_FIL MU=HEADQUART
```

uxlst TSK : liste les tâches.

Item	Format	Défaut
EXP SIM INT APP	3c	S_ESPEXE
TSK=	64c *	*
SES=	64c *	*
VSES=	3n *	*
UPR=	64c *	*
VUPR=	3n *	*
MU=	64c *	*
MODEL NOMODEL		NOMODEL
TTYPE=S O P C *	1c *	*
FULL		INF
OUTPUT=file_name		

Exemple :

```
uxlst TSK FULL OUTPUT=c:\temp\LstTsk.txt
```

4.2.1 Description des items

TSK : Code de la Tâche. Format : 64 caractères alphanumériques. Incompatible avec UPR, SES et USER. Si le code de la Tâche n'est pas renseigné, par défaut il vaut : TSK(SSS)(VVV)(UUU)(VVV) avec :

- (SSS)(VVV) : code et version de la session
- (UUU)(VVV) : code et version de l'Uproc

UPR : Code Uproc. Format : 64 caractères alphanumériques. Requiert la saisie d'un code Uproc (valeur générique acceptée sauf pour uxadd) obligatoire.

VUPR : Version d'Uproc. Format : 3 caractères numériques. Permet la saisie d'une version particulière d'une Uproc. Par défaut, toutes les versions des Uprocs renseignées sont considérées (inutile avec SIM et EXP).

SES : Code session. Format : 64 caractères alphanumériques. Requiert la saisie d'un code session (valeur générique acceptée sauf pour uxadd).

VSES : Version de session. Format : 3 caractères numériques. Permet la saisie d'une version particulière d'une session. Par défaut, toutes les versions des sessions renseignées sont considérées (inutile avec SIM et EXP).

MU : Code du modèle ou de l'unité de gestion. Format : 64 caractères alphanumériques. Requiert la saisie d'un code unité de gestion (valeur générique acceptée sauf pour uxadd), saisie obligatoire.

TTYPE : Filtre sur le type de la tâche : S (planifiée), O (optionnelle), P (provoquée), C (cyclique) ou *.

NOMODEL : nature de la tâche. Indique que la tâche sélectionnée est de nature non modèle. MODEL indique que la tâche sélectionnée est de nature modèle.

FULL : Option d'affichage. Affiche toutes les informations disponibles sur la liste des tâches. Autre valeur (par défaut) INF : affiche uniquement l'entête de la tâche.

 FULL et INF sont incompatibles entre eux.

OUTPUT : Stockage des résultats. Si l'item n'est pas renseigné la sortie se fait à l'écran. La description de cet item est fournie dans le dictionnaire d'items.

4.2.2 Exemples

```
uxhld TSK EXP UPR=D_LOAD_FIL SES=D_LOAD_BCK MU=SIEGE NOMODEL
```

Désactive la tâche non modèle de la session D_LOAD_BCK dont l'Uproc d'en-tête est D_LOAD_FIL sur l'unité de gestion SIEGE dans l'espace exploitation.

```
uxlst TSK EXP UPR=D_LOAD_FIL SES=D_LOAD_BCK MU=SIEGE FULL
```

Liste l'ensemble des caractéristiques de la tâche qui vient d'être désactivée.

```
uxrls TSK EXP UPR=D_LOAD_FIL SES=D_LOAD_BCK MU=SIEGE
```

Réactive la tâche de l'Uproc IU_TEST sur l'unité de gestion SIEGE dans l'espace exploitation.

4.3 Les lancements prévus

uxadd FLA : crée un lancement prévu dans la liste des lancements.

La commande affiche dans la sortie standard le numéro de lancement associé avec le format suivant : "numlanc nnnnnnn".

Item	Format	Défaut
EXP SIM INT APP	3c	S_ESPEXE
TSK=	64c	
SES=	64c	" "
VSES=	3n	
UPR=	64c	
VUPR=	3n	
MU=	64c	
PRINTER=	4c	
PRTY=	3n	100
QUEUE=	31c	SYS_BATCH (SYS\$BATCH sous OpenVMS)
USER=	64c	
LSTART=	(U_FMT_DATE,hhmm)	System time
LEND=	(U_FMT_DATE,hhmm)	System time + 5 minutes
EXLW=	(hhmm,hhmm)	(0000,0000)
PDATE=	U_FMT_DATE	
LAUNCHVAR=	((var,val),...)	
FORCE NOFORCE		NOFORCE
BYPASS NOBYPASS		NOBYPASS
MUTIME ASTIME		

Item	Format	Défaut
VARTSK NOVARTSK		
INFO	50c	
SEV	4n ou " "	

La commande OS/400 UXADDFLA permet de créer un lancement à partir de la ligne de commande. F4 permet de saisir les paramètres unitairement. F11 affiche le nom de chaque paramètre.

Exemple :

```
uxadd FLA UPR=D_LOAD_FIL MU=HEADQUART USER=sup500e LSTART="(20020901,2000)"
LEND="(20020901,2200)" PDATE=20020901 LAUNCHVAR="( (VARDATE,2010/01/01) )"
```

uxupd FLA : modifie un lancement prévu.

Item	Format	Défaut
EXP SIM INT APP	3c	S_ESPEXE
TSK=	64c *	
SES=	64c *	
VSES=	3n *	
UPR=	64c *	
VUPR=	3n *	
MU=	64c *	
NSEQ=	7n *	
SINCE=	(U_FMT_DATE,hhmm)	
BEFORE=	(U_FMT_DATE,hhmm)	
STATUS= L H W O R	1c	
PRINTER=	4c	
PRTY=	3n	
QUEUE=	31c	
USER=	64c	
LSTART=	(U_FMT_DATE,hhmm)	
LEND=	(U_FMT_DATE,hhmm)	
EXLW=	(hhmm,hhmm)	
NOW		
PDATE=	U_FMT_DATE	
LAUNCHVAR=	((var,val),...)	
FORCE NOFORCE		
BYPASS NOBYPASS		
MUTIME ASTIME		
INFO	50c	
SEV	4n or " "	

La commande OS/400 UXUPDFLA permet de modifier un lancement à partir de la ligne de commande. F4 permet de saisir les paramètres unitairement. F11 affiche le nom de chaque paramètre.

Exemple :

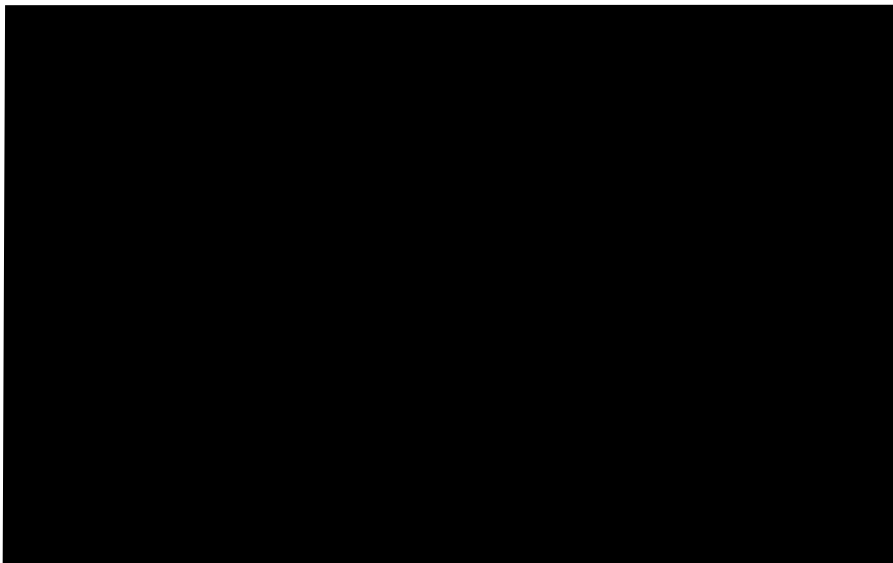
```
uxupd FLA UPR=D_LOAD_FIL MU=HEADQUART LAUNCHVAR="( ( VARNUM, 693 ) )"
NSEQ=0000175
```

uxdlf FLA : supprime un lancement prévu.

uxhld FLA : suspend un lancement prévu.

uxrls FLA : réactive un lancement prévu.

Pour ces trois commandes, la syntaxe est la suivante N



Item	Format	Défaut
NSEQ=	7n *	
SINCE=	(U_FMT_DATE,hhmm)	
BEFORE=	(U_FMT_DATE,hhmm)	
STATUS=	L H W O R	
OUTPUT=file_name		

Exemple :

```
uxshw FLA EXP SES=D_LOAD_BCK UPR=D_LOAD_FIL MU=HEADQUART NSEQ=* |more
```

uxlst FLA : liste les lancements prévus.

Item	Format	Défaut
EXP SIM INT APP	3c	S_ESPEXE
TSK=	64c *	
SES=	64c *	" "
VSES=	3n *	*
UPR=	64c *	*
VUPR=	3n *	*
MU=	64c *	*
NUPR=	7n *	
NSESS=	7n *	
NSEQ=	7n *	
SINCE=	(U_FMT_DATE,hhmm)	
BEFORE=	(U_FMT_DATE,hhmm)	
STATUS=	L H W O R	
FULL INF		INF
ASTIME MUTIME		
OUTPUT=file_name		

Exemple :

```
uxlst FLA FULL OUTPUT=c:\temp\LstFla.txt
```

4.3.1 Description des items

TSK : Code de la Tâche. Format : 64 caractères alphanumériques. Incompatible avec UPR, SES et USER.

UPR : Code Uproc. Format : 64 caractères alphanumériques. Requiert la saisie d'un code Uproc (valeur générique acceptée sauf pour uxadd).

VUPR : Version d'Uproc. Format : 3 caractères numériques. Permet la saisie d'une version particulière d'une Uproc. Par défaut, toutes les versions des Uprocs renseignées sont considérées (inutile avec SIM et EXP).

SES : Code session. Format : 64 caractères alphanumériques. Requiert la saisie d'un code session (valeur générique acceptée sauf pour uxadd).

VSES : Version de session. Format : 3 caractères numériques. Permet la saisie d'une version particulière d'une session. Par défaut, toutes les versions des sessions renseignées sont considérées (inutile avec SIM et EXP).

MU : Code de l'unité de gestion. Format : 64 caractères alphanumériques. Requiert la saisie d'un code unité de gestion (valeur générique acceptée sauf pour uxadd).

NUPR, NSESS : numéro d'exécution d'Uproc / de Session. Format : 7 caractères numériques ou *.

NSEQ : Numéro de lancement. Format : 7 caractères numériques, obligatoire pour uxupd. Indique quel numéro de lancement doit être affecté par la commande (pour le cas où il existe plusieurs lancements portant sur la même Uproc session et unité de gestion).

SINCE : Début de la plage horaire d'observation. Permet de déterminer, en fonction de leur plage horaire prévue, les lancements à observer (sont extraits tous les lancements dont la plage horaire présente une intersection non vide avec la plage d'observation).

- Sous Windows/UNIX/OpenVMS, le format est : (U_FMT_DATE,HHMM) le format de la date est indiqué par la valeur du paramètre U_FMT_DATE, heure et minutes sur 4 caractères. Le nombre de caractères doit être respecté exactement.
- Sous OS/400, la date doit être saisie au format utilisé par le système. Le format de l'heure est HHMMSS.

BEFORE : Fin de la plage horaire d'observation.

- Sous Windows/UNIX/OpenVMS, le format est : (U_FMT_DATE,HHMM) le format de la date est indiqué par la valeur du paramètre U_FMT_DATE, heure et minutes sur 4 caractères. Le nombre de caractères doit être respecté exactement.
- Sous OS/400, la date doit être saisie au format utilisé par le système. Le format de l'heure est HHMMSS.

STATUS : Etat d'un lancement. Par défaut tous les lancements sont considérés. Permet de sélectionner les lancements en fonction de leur état par la saisie d'une ou plusieurs des valeurs suivantes (séparées par une virgule) :

- L : attente de premier lancement,
- H : suspendu (désactivé),
- W : attente d'événement,
- O : horaire dépassé,
- R : refusé au pilotage.

PRINTER : Imprimante. Non utilisé sous UNIX et Windows.

PRTY : Priorité. Format : 3 caractères numériques de 001 à 255. Valeur par défaut : 100. Priorité d'entrée de la tâche dans la queue batch.

QUEUE : Queue batch. Format : 32 caractères alphanumériques (" \$" accepté sous OpenVMS). Valeur par défaut : SYS_BATCH (SYS\$BATCH sous OpenVMS). Queue batch de soumission de la tâche.

USER : Compte de soumission. Format : 64 caractères alphanumériques. Compte dans l'environnement duquel l'exécution sera soumise.

FORCE : Lancement forcé en fin de période. Valeur par défaut : NOFORCE. Indique que l'exécution du lancement sera forcée en fin de plage horaire de la tâche même si toutes les conditions de l'Uproc ne sont pas vérifiées.

CENTRAL : Obsolète, présent pour compatibilité uniquement.

RESTART : Obsolète, présent pour compatibilité uniquement.

BYPASS : Lancement sans pilotage. Indique que les conditions liées à l'Uproc ne seront pas examinées. L'Uproc pourra donc s'exécuter même si ses conditions ne sont pas vérifiées. Dans le cas d'un lancement portant sur une session, seule la première Uproc de la session bénéficie d'un lancement sans pilotage, les Uprocs suivantes verront leurs conditions normalement vérifiées.

- Sous Windows/UNIX/OpenVMS : Valeur par défaut : NOBYPASS.
- Sous OS/400 : Y (yes) pour un lancement sans pilotage; N (No) sinon, valeur par défaut.

LSTART : Date et heure de début de plage horaire du lancement, par défaut le lancement est immédiat.

- Sous Windows/UNIX/OpenVMS, le format est (U_FMT_DATE,HHMM) où U_FMT_DATE est la date de début du lancement au format indiqué par la valeur du paramètre U_FMT_DATE, HHMM est l'heure de début du lancement en heures et minutes. Le nombre de caractères doit être respecté exactement. Obligatoire pour uxadd. Incompatible avec NOW.
- Sous OS/400, la date doit être saisie au format utilisé par le système. Le format de l'heure est HHMMSS.

LEND : Date et heure de fin de plage horaire du lancement, par défaut la durée de la plage horaire de lancement est de 5 minutes.

- Sous Windows/UNIX/OpenVMS, le format est (U_FMT_DATE,HHMM) où U_FMT_DATE est la date de fin du lancement au format indiqué par la valeur du paramètre U_FMT_DATE, HHMM est l'heure de fin du lancement en heures et minutes. Le nombre de caractères doit être respecté exactement. Obligatoire pour uxadd. Incompatible avec NOW.
- Sous OS/400, la date doit être saisie au format utilisé par le système. Le format de l'heure est HHMMSS.

EXLW : Plage horaire d'exclusion du lancement.

- Sous Windows/UNIX/OpenVMS, le format est (HHMM,HHMM) où (HHMM est l'heure de début de la plage horaire d'exclusion du lancement en heures et minutes, HHMM) est l'heure de fin de la plage horaire d'exclusion du lancement en heures et minutes. Le nombre de caractères doit être respecté exactement. Incompatible avec NOW.
- Sous OS/400, la date doit être saisie au format utilisé par le système. Le format de l'heure est HHMMSS.

NOW : Début de la fenêtre de lancement : heure courante du nœud de l'environnement courant et durée de 5 minutes. Peut être utilisé avec la commande uxupd FLA sous Windows/UNIX/OpenVMS uniquement. Incompatible avec les items LSTART, LEND et EXLW.

PDATE : Date de traitement associée au lancement. La valeur dépend de la période fonctionnelle de l'Uproc.

- Sous Windows/UNIX/OpenVMS, le format est indiqué par le paramètre U_FMT_DATE. Le nombre de caractères doit être respecté exactement.
- Sous OS/400, la date doit être saisie au format utilisé par le système.

LAUNCHVAR : Liste complète des variables du lancement et de leurs valeurs. Format : ((var,val), (var),...). La liste est encadrée de parenthèses et de guillemets, chaque couple variable – valeur est encadré de parenthèses, chaque couple variable – valeur est séparé du suivant par une virgule (sans espace).

- Le nom de la variable doit être choisi dans la liste des variables de l'Uproc identifiant le lancement.
 - La valeur de la variable, si elle est indiquée remplace la valeur de la variable de la tâche ou de l'Uproc, si la valeur n'est pas indiquée (la virgule ne doit pas être présente) c'est la valeur par défaut de la variable de la tâche ou de l'Uproc qui est prise en compte. Si la valeur contient des caractères
-

non alphanumériques (espaces, /...) elle doit être encadrée de guillemets. Le format de la valeur doit être compatible avec le format déclaré lors de la création de la variable dans l'Uproc.

 La sortie de la commande ne contient pas la valeur des variables de type mot de passe.

INF : Option d'affichage des informations. Valeur par défaut. Permet l'affichage du nom des lancements uniquement. Autre valeur : **FULL** : Permet d'accéder à l'ensemble des informations des lancements.

 **INF** et **FULL** sont incompatibles entre eux.

ASTIME ou **MUTIME** : Heure de référence utilisée par la commande. Si **MUTIME** est utilisé, la colonne **MU TIME OFFSET** est affichée (uxlst). Si ce paramètre n'est pas utilisé dans la commande :

- Si le paramètre noeud **U_TREF** n'est pas défini ou vaut "**ASTIME**" : l'heure de référence est celle du serveur.
- Si le paramètre noeud **U_TREF** vaut "**MUTIME**" : l'heure de référence est celle de l'Unité de Gestion.

 Sous OS/400, le paramètre **TYPETIME** peut prendre les valeurs **MUTIME** ou **ASTIME**; valeur par défaut **ASTIME**.

VARTSK : Indique que le lancement est créé en utilisant les valeurs des variables de la tâche correspondante (lorsqu'elle existe). Sinon, le lancement est créé en n'utilisant pas les valeurs des variables de la tâche correspondante (lorsqu'elle existe) mais les valeurs de variables de l'Uproc définissant le lancement.

- Sous Windows/UNIX/OpenVMS : Autre valeur : **NOVARTSK**
- Sous OS/400 : **Y** (yes) pour utiliser les valeurs des variables de la tâche (valeur par défaut); **N** (No) pour utiliser les valeurs des variables de l'Uproc.

INFO: champ Information du lancement.

- Si l'argument **INFO** est présent et valorisé à autre chose que " " (caractère espace), le champ Information du lancement contient la valeur de l'argument **INFO**.
- Si l'argument **INFO** n'est pas présent ou si il est valorisé à " " (caractère espace), le contenu du champ Information du lancement respecte les règles suivantes :
 - Par défaut le champ Information du lancement contient la valeur du champ Information de l'Uproc (s'il est défini dans l'Uproc).
 - Si le champ Information de l'Uproc n'est pas défini et si le paramètre noeud **U_FMT_INFO** a été défini (catégorie Réglages avancés), le champ Information du lancement contient les libellés de l'Uproc et de la Session du lancement (pour l'utilisation du paramètre noeud **U_FMT_INFO**, reportez-vous au Manuel Utilisateur d'UniViewer).

SEV ou **SEVERITY** (sous OS/400) : champ Sévérité du lancement.

- Si l'argument **SEV/SEVERITY** est présent et valorisé à autre chose que " " (caractère espace), le champ Sévérité du lancement contient valeur de l'argument **SEV**.
- Si l'argument **SEV/SEVERITY** n'est pas présent ou si il est valorisé à " " (caractère espace), le champ Sévérité du lancement contient la valeur du champ sévérité de l'Uproc du lancement.

OUTPUT : Stockage des résultats. Si l'item n'est pas renseigné la sortie se fait à l'écran.

4.3.2 Exemples

```
uxadd FLA EXP SES=D_LOAD_BCK UPR=D_LOAD_FIL VUPR=000 MU=SIEGE USER=admin
LSTART=(19990507,1255) LEND=(19990507,1700) PDATE=19990605 FORCE
```

Crée un lancement à partir de la session D_LOAD_BCK, Uproc entête D_LOAD_FIL sur l'unité de gestion SIEGE dans l'espace d'exploitation pour le compte utilisateur admin. Le lancement est prévu le 7 mai 1999 de 12h55 à 17h00 avec l'option lancement forcé en fin de période dans le cas où le paramètre U_FMT_DATE vaut YYYYMMDD.

```
uxupd FLA EXP SES=D_LOAD_BCK UPR=D_LOAD_FIL VUPR=000 MU=SIEGE NOFORCE
NSEQ=0000002
```

Met à jour le lancement créé ci-dessus en indiquant que le lancement ne sera plus forcé en fin de période s'il était nécessaire. Le numéro de séquence NSEQ est donné par une commande uxlst fla.

```
uxhld FLA EXP SES=D_LOAD_BCK UPR=D_LOAD_FIL VUPR=000 MU=SIEGE NSEQ=0000002
```

Suspend le lancement créé ci-dessus.

```
uxadd FLA UPR=IU_LPR mu=X01 USER=root PDATE=00/00/0000 LSTART="
(08/01/1998,1000)" LEND="(09/12/1998,1000)" LAUNCHVAR="( (VARTXT,TXT_VAL_
FLA), (VARDATE,2010/01/01))"
```

Crée un lancement de l'Uproc IU_LPR sur l'unité de gestion X01 avec de nouvelles valeurs pour les deux variables de la tâche.

```
uxupd FLA UPR=IU_LPR MU=X01 NSEQ="*" LAUNCHVAR=( (VARNUM,693))
```

Modifie la valeur de la variable VARNUM, les valeurs de VARTXT et VARDATE sont conservées.

4.4 Les événements

uxadd EVT : ajoute un événement dans la liste des événements.

Item	Format	Défaut
EXP SIM INT APP	3c	S_ESPEXE
SES=	64c	" "
VSES=	3n	
NSESS=	7n	
UPR=	64c	
VUPR=	3n	
MU=	64c	
USER=	64c	
STEP=	2n	00
PDATE=	U_FMT_DATE	
NSTAT=S P E T A W	1c	T

Exemple :

```
uxadd EVT EXP SES=D_LOAD_BCK NSESS=0000010 UPR=D_LOAD_FIL MU=HEADQUART
USER=univ54e PDATE=20021030 NSTAT=E
```

uxupd EVT : modifie un événement.

Item	Format	Défaut
EXP SIM INT APP	3c	S_ESPEXE

Item	Format	Défaut
SES=	64c *	" "
VSES=	3n *	
NSESS=	7n	
UPR=	64c *	
VUPR=	3n *	
NUPR=	7n	
MU=	64c *	
USER=	64c	
STEP=	2n	
PDATE=	U_FMT_DATE	
NSTAT=S P E T A W	1c	T
STATUS= S P E T A *		AT

Exemple :

```
uxupd EVT EXP SES="\ " UPR=D_LOAD_FIL MU=HEADQUART NUPR=0000118 NSTAT=A
```

uxshw EVT : affiche les caractéristiques d'un événement.

Item	Format	Défaut
EXP SIM INT APP	3c	S_ESPEXE
SES=	64c *	" "
VSES=	3n *	
NSESS=	7n	
UPR=	64c *	
VUPR=	3n *	
NUPR=	7n	
MU=	64c *	
PDATE=	U_FMT_DATE	
SINCE=	(U_FMT_DATE,hhmm)	
BEFORE=	(U_FMT_DATE,hhmm)	
STATUS=S P E T A *		*
OUTPUT=file_name		

Exemple :

```
uxshw EVT EXP SES=* UPR=* MU=HEADQUART
```

uxdlit EVT : supprime un événement.

Item	Format	Défaut
EXP SIM INT APP	3c	S_ESPEXE
SES=	64c *	" "
VSES=	3n *	

Item	Format	Défaut
NSESS=	7n	
UPR=	64c *	
VUPR=	3n *	
NUPR=	7n	
MU=	64c *	
PDATE=	U_FMT_DATE	
STATUS=S P E T A *		AT
SINCE=	(U_FMT_DATE,hhmm)	
BEFORE=	(U_FMT_DATE,hhmm)	

Exemple :

```
uxdlst EVT EXP SES=\" \" UPR=D_LOAD_FIL MU=HEADQUART PDATE=20020831
USER=admin
```

uxlst EVT : liste les événements.

Item	Format	Défaut
EXP SIM INT APP	3c	S_ESPEXE
SES=	64c *	
VSES=	3n *	
NSESS=	7n	
UPR=	64c *	
VUPR=	3n *	
NUPR=	7n	
PDATE=	U_FMT_DATE	
MU=	64c *	
SINCE=	(U_FMT_DATE,hhmm)	
BEFORE=	(U_FMT_DATE,hhmm)	
STATUS=S P E T A *		*
FULL INF		INF
OUTPUT=file_name		

Exemple :

```
uxlst EVT FULL OUTPUT=c:\temp\LstEvt.txt
```

4.4.1 Description des items

TSK : Code de la Tâche. Format : 64 caractères alphanumériques.

UPR : Code Uproc. Format : 64 caractères alphanumériques. Requiert la saisie d'un code Uproc (valeur générique acceptée sauf pour uxadd).

VUPR : Version de l'Uproc. Format : 3 caractères numériques. Permet la saisie d'une version particulière d'une Uproc. Par défaut, toutes les versions des Uprocs renseignées sont considérées (inutile avec SIM et EXP).

NUPR : Numéro d'exécution de l'Uproc. Format : 7 caractères numériques. Permet la saisie du numéro d'exécution de l'Uproc. Ce numéro permet de différencier deux événements portant sur la même Uproc au sein de la même session sur la même unité de gestion pour la même date de traitement.

SES : Code session. Format : 64 caractères alphanumériques. Requiert la saisie d'un code session (valeur générique acceptée sauf pour uxadd).

VSES : Version de session. Format : 3 caractères numériques. Permet la saisie d'une version particulière d'une session. Par défaut, toutes les versions des sessions renseignées sont considérées (inutile avec SIM et EXP).

NSESS : Numéro d'exécution de la session. Format : 7 caractères numériques. Permet la saisie du numéro d'exécution de la session. Ce numéro permet de différencier deux événements portant sur la même session sur la même unité de gestion pour la même date de traitement.

MU : Code de l'unité de gestion. Format : 64 caractères alphanumériques. Requiert la saisie d'un code unité de gestion (valeur générique acceptée sauf pour uxadd) obligatoire.

SINCE : Début de la plage horaire d'observation. Défaut : (000000,0000). Permet de déterminer, en fonction de leur date et heure d'exécution, les événements à observer. Format : (U_FMT_DATE,HHMM) la date au format indiqué par la valeur du paramètre U_FMT_DATE, heure et minutes sur 4 caractères. Le nombre de caractères doit être respecté exactement.

BEFORE : Fin de la plage horaire d'observation. Défaut : (9991231,2359). Format : (U_FMT_DATE,HHMM) la date au format indiqué par la valeur du paramètre U_FMT_DATE, heure et minutes sur 4 caractères. Le nombre de caractères doit être respecté exactement.

STATUS : Etat d'un événement. Permet de sélectionner les événements en fonction de leur état par la saisie d'une ou plusieurs des valeurs suivantes (séparées par une virgule ou pas) :

- S : pilotage en cours (Started),
- P : attente exécution (Pending),
- E : exécution en cours,
- T : terminé,
- A : incidenté (Aborted).

USER : Compte de soumission. Format : 64 caractères alphanumériques. Compte de soumission pour lequel l'événement est utilisé.

NSTAT : Etat de l'événement. Indique l'état dans lequel l'événement doit être défini. Format : 1 caractère alphabétique parmi :

- S : pilotage en cours (Started),
- P ou W : attente exécution (Pending),
- E : exécution en cours,
- T : terminé,
- A : incidenté (Aborted).

STEP : numéro de jalon de l'événement. Format : 2 caractères numériques. Indique le numéro de jalon associé à l'événement.

PDATE : Date de traitement associée à l'événement. Le nombre de caractères doit être respecté exactement. Format indiqué par la valeur du paramètre U_FMT_DATE. La valeur dépend de la période fonctionnelle de l'Uproc.

INF : Option d'affichage des informations. Valeur par défaut. Permet l'affichage du nom des événements uniquement. Autre valeur : **FULL** : Permet d'accéder à l'ensemble des informations des événements.

 **INF** et **FULL** sont incompatibles entre eux.

OUTPUT : Stockage des résultats. Si l'item n'est pas renseigné la sortie se fait à l'écran. La description de cet item est fournie dans le dictionnaire d'items.

4.4.2 Exemples

```
uxadd EVT EXP UPR=D_LOAD_FIL MU=SIEGE USER=admin PDATE=19990507 NSTAT=T
```

Crée un événement à partir de l'Uproc D_LOAD_FIL

Exemple :

```
uxpur CTL EXP SES=* UPR=* MU=* SINCE=(20020101,0000) BEFORE=(20020701,0000)
```

uxend CTL : arrête une exécution en cours (état "E") et par conséquent tous les processus qui y sont rattachés. L'état est alors positionné à Incidenté ("A"). Si l'exécution de l'Uproc est protégée par une commande uxsetabort (voir paragraphe "[Interdire l'arrêt de l'exécution d'une Uproc](#)"), un message informe l'utilisateur de la suite donnée à la commande uxend ctl: repoussée à la fin de la zone protégée ou abandonnée.

Item	Format	Défaut
EXP SIM INT APP	3c	S_ESPEXE
TSK=	64c *	
SES=	64c *	" "
VSES=	3n *	
NSESS=	7n *	
UPR=	64c *	
VUPR=	3n *	
NUPR=	7n *	
MU=	64c *	
SINCE=	(U_FMT_DATE,hhmm)	
BEFORE=	(U_FMT_DATE,hhmm)	
STATUS=	H O R S W P E F T A *	E
OUTPUT=file_name		

Exemple :

```
uxend CTL EXP SES=D_LOAD_BCK UPR=D_LOAD_FIL MU=HEADQUART
```

uxrst CTL : reprend une exécution.

Item	Format	Défaut
EXP SIM INT APP	3c	S_ESPEXE
TSK=	64c *	
SES=	64c *	" "
VSES=	3n *	
NSESS=	7n *	
UPR=	64c *	
VUPR=	3n *	
NUPR=	7n *	
MU=	64c *	
SINCE=	(U_FMT_DATE,hhmm)	
BEFORE=	(U_FMT_DATE,hhmm)	
STATUS=	T A	
LW=	(hhmm,hhmm)	(0000,0000)

Item	Format	Défaut
STEP=	2n	00
LAUNCHVAR=	((var,val),...)	
BYPASS NOBYPASS		NOBYPASS
PCHAIN_RELAUNCH=	FROMABORTED ONLYABORTED	
OUTPUT=file_name		

Exemple :

```
uxrst CTL EXP SES=D_LOAD_BCK UPR=D_LOAD_FIL MU=HEADQUART LW=(1600,1700)
BYPASS LAUNCHVAR=( (VARTXT,0), (VARNUM,200) )
```

uxlst CTL : liste les exécutions de façon plus ou moins complète en fonction des critères de sélection et des informations demandées.

Item	Format	Défaut
EXP SIM INT APP	3c	S_ESPEXE
TSK=	64c *	
SES=	64c *	" "
VSES=	3n *	
NSESS=	7n *	
UPR=	64c *	
VUPR=	3n *	
NUPR=	7n *	
MU=	64c *	
NSEQ	7n *	
SINCE=	(U_FMT_DATE,hhmm)	
BEFORE=	(U_FMT_DATE,hhmm)	
NOTIFY		
STATUS=	H O R S W P E F T A *	*
RECOVER		
INF HST LOG FULL FULL INFO MGT VAR		INF
ASTIME MUTIME		
OUTPUT=file_name		

Exemple :

```
uxlst CTL EXP FULL OUTPUT=c:\temp\LstCtl.txt
```

4.5.1 Description des items

TSK : Code de la Tâche. Format : 64 caractères alphanumériques.

UPR : Code Uproc. Format : 64 caractères alphanumériques. Requiert la saisie d'un code Uproc (valeur générique acceptée sauf pour uxrst).

VUPR : Version d'Uproc. Format : 3 caractères numériques. Permet la saisie d'une version particulière d'une Uproc. Par défaut, toutes les versions des Uprocs renseignées sont considérées (inutile avec SIM et EXP).

NUPR : Numéro d'exécution de l'Uproc. Format : 7 caractères numériques. Permet la saisie du numéro d'exécution de l'Uproc. Ce numéro permet de différencier deux événements portant sur la même Uproc au sein de la même session sur la même unité de gestion pour la même date de traitement.

SES : Code session. Format : 64 caractères alphanumériques. Requiert la saisie d'un code session (valeur générique acceptée sauf pour uxrst).

VSES : Version de session. Format : 3 caractères numériques. Permet la saisie d'une version particulière d'une session. Par défaut, toutes les versions des sessions renseignées sont considérées (inutile avec SIM et EXP).

NSESS : Numéro d'exécution de la session. Format : 7 caractères numériques. Permet la saisie du numéro d'exécution de la session. Ce numéro permet de différencier deux événements portant sur la même session sur la même unité de gestion pour la même date de traitement.

MU : Code de l'unité de gestion. Format : 64 caractères alphanumériques. Requiert la saisie d'un code unité de gestion (valeur générique acceptée sauf pour uxrst).

NSEQ : Numéro de lancement. Format : 7 caractères numériques Indique le numéro de lancement (pour le cas où il existe plusieurs lancements identiques).

SINCE : Début de la plage horaire d'observation. Défaut : (000000,0000). Permet de déterminer, en fonction de leur date de début d'exécution, les exécutions à observer. Format : (U_FMT_DATE,HHMM) la date est au format indiqué par la valeur du paramètre U_FMT_DATE, heure et minutes sur 4 caractères. Le nombre de caractères doit être respecté exactement.

BEFORE : Fin de la plage horaire d'observation. Défaut : (19991231,2359). Permet de déterminer, en fonction de leur date de fin d'exécution, les exécutions à observer. Format : (U_FMT_DATE,HHMM) la date est au format indiqué par la valeur du paramètre U_FMT_DATE, heure et minutes sur 4 caractères. Le nombre de caractères doit être respecté exactement.

NOTIFY : option de sélection sur la date. Valeur par défaut " ". Associé à SINCE et / ou BEFORE, permet de filtrer les exécutions selon leur date de dernière mise à jour par l'automate (par exemple passage de l'état "exécution en cours" à "terminé"). Si le mot clé n'est pas cité, les enregistrements sont sélectionnés selon leur date de début et de fin d'exécution.

STATUS : Etat d'une exécution. Par défaut toutes les exécutions sont considérées. Permet de sélectionner les exécutions en fonction de leur état par la saisie d'une ou plusieurs des valeurs suivantes (séparées par une virgule) :

- H : suspendu (désactivé),
- O : horaire dépassé,
- R : refusé au pilotage.
- S : pilotage en cours (Scheduling),
- W : attente d'événement (Waiting),
- P : attente exécution (Pending),
- E : exécution en cours,
- F : consignes en cours,
- T : terminé,
- A : incidenté (Aborted).
- * : affiche la dernière exécution d'un job, quelque soit son statut. Si un job a été repris, seule sa dernière exécution sera affichée.

RECOVER : Affiche toutes les exécutions des jobs (item STATUS=*) à l'heure à laquelle la commande uxlst CTL a été exécutée. Si un job a été repris, toutes ses exécutions sont listées.

USER : Compte de soumission d'une exécution. Format : 64 caractères alphanumériques. Permet de sélectionner les exécutions en fonction du compte de soumission de celles ci. Par défaut tous les comptes de soumission sont considérées.

LW : Plage horaire de la reprise. Format : (HHMM,HHMM) où :

- (HHMM est l'heure de début de la plage horaire de la reprise en heures et minutes.
- HHMM) est l'heure de fin de la plage horaire de la reprise en heures et minutes.

 Le nombre de caractères doit être respecté exactement.

STEP : Numéro de jalon de la reprise. Format : 2 caractères numériques. Indique le numéro de jalon à partir duquel la reprise sera effectuée.

LAUNCHVAR : Liste complète des variables de l'exécution et de leurs valeurs. Format : ((var,val), (var)...). La liste est encadrée de parenthèses et de guillemets, chaque couple variable – valeur est encadré de parenthèses, chaque couple variable – valeur est séparé du suivant par une virgule (sans espace).

- Le nom de la variable doit être choisi dans la liste des variables du lancement.
- La valeur indiquée remplace la valeur de la variable dans le lancement, si la valeur n'est pas indiquée (la virgule ne doit pas être présente) c'est la valeur par défaut de la variable du lancement, de la tâche ou de l'Uproc qui est prise en compte. Si la valeur contient des caractères non alphanumériques (espaces, /...) elle doit être encadrée de guillemets.

 Le format de la valeur doit être compatible avec le format déclaré lors de la création de la variable dans l'Uproc.

BYPASS : Lancement sans pilotage. Valeur par défaut : NOBYPASS. Indique que les conditions liées à l'Uproc ne seront pas examinées. L'Uproc pourra donc s'exécuter même si ses conditions ne sont pas vérifiées. Dans le cas d'un lancement portant sur une session, seule la première Uproc de la session bénéficie d'un lancement sans pilotage, les Uprocs suivantes verront leurs conditions normalement vérifiées.

INF : Option d'affichage. Valeur par défaut. Permet l'affichage du nom et de l'état uniquement des exécutions du suivi de l'exploitation.

- Autre valeur : HST : permet l'affichage des traces automates de toutes les exécutions sélectionnées.
- Autre valeur : LOG : permet l'affichage du fichier log de toutes les exécutions sélectionnées.
- Autre valeur : FULL : permet l'affichage de l'ensemble des informations des exécutions,
- Autre valeur: FULL INFO; affiche les exécutions ayant les valeurs des paramètres Information et Sévérité.
- Autre valeur : VAR : permet l'affichage de la valeur des variables de l'exécution.

 INF, FULL, LOG, HST et VAR sont incompatibles entre eux. LOG est incompatible avec OUTPUT.

ASTIME ou **MUTIME** : Heure de référence utilisée par la commande. Si MUTIME est utilisé, la colonne MU TIME OFFSET est affichée (uxlst). Si ce flag n'est pas utilisé dans la commande :

- Si le paramètre noeud U_TREF n'est pas défini ou vaut "ASTIME" : l'heure de référence est celle du serveur.
- Si le paramètre noeud U_TREF vaut "MUTIME" : l'heure de référence est celle de l'Unité de Gestion.

PCHAIN_RELAUNCH : utilisation par Dollar Universe Manager for SAP Solutions.

- FROMABORTED : les jobs incidentés (et leurs successeurs éventuellement) sont relancés.
- ONLYABORTED : seuls les jobs incidentés sont relancés.

OUTPUT : Stockage des résultats. Si l'item n'est pas renseigné la sortie se fait à l'écran. OUTPUT est incompatible avec LOG.

4.5.2 Exemples

```
uxrst CTL EXP SES=D_LOAD_BCK UPR=D_LOAD_FIL MU=SIEGE LW=(1600,1700) BYPASS
```

Reprend l'exécution, dans l'espace exploitation, de la session D_LOAD_BCK à partir de l'Uproc D_LOAD_FIL sur l'unité de gestion SIEGE sans examiner les conditions liées à cette Uproc, entre 16 heures et 17 heures. Cette reprise est possible car l'événement correspondant existe dans l'état Incidenté.

```
uxlst CTL EXP SES=D_LOAD_BCK MU=SIEGE INF
```

Affiche le statut de toutes les exécutions de la session D_LOAD_BCK sur l'unité de gestion SIEGE du contrôle de l'exploitation de l'espace exploitation.

```
uxlst CTL UPR=IU* SINCE=(2001/06/12,1200) BEFORE=(2001/06/13,1200) NOTIFY
```

Extrait toutes les Exécutions mises à jour (ayant changé d'état) entre le 2001/06/12 à midi et le 2001/06/13 à midi - pour une date au format YYYY/MM/DD.

```
uxpur CTL EXP UPR=D_* MU=SIEGE SINCE=(19990501,1200) BEFORE=(19990508,1200)
```

Purge le contrôle de l'exploitation de toutes les exécutions de l'Uproc IU_TEST version 001 sur l'unité de gestion SIEGE entre le 1er mai 1999 à 12 heures et le 8 mai 1999 à 12 heures pour une date au format YYYYMMDD.

```
uxrst CTL UPR=IU_LPR MU=X01 NUPR=%NUMPROC% LAUNCHVAR="( (VARTXT,0),  
(VARNUM,200) )"
```

Modifie deux des valeurs des variables du lancement lors de la commande de reprise de l'exécution. les autres variables restent telles qu'elles ont été définies pour le lancement.

4.6 Les notes

uxadd NOTE : ajoute une note à un job (lancement ou exécution).

Item	Format	Défaut
EXP SIM INT APP	3c	S_ESPEXE
NUPR= NSEQ=	7n	
NOTE=	1024c	

Exemple :

```
uxadd note note="texte" nupr=2
```

uxupd NOTE : met à jour la note associée à un job.

Item	Format	Défaut
EXP SIM INT APP	3c	S_ESPEXE
NUPR= NSEQ=	7n	
NOTE=	1024c	

Exemple :

```
uxupd note note="texte" nupr=2
```

uxshw NOTE: affiche les caractéristiques de la note.

Item	Format	Défaut
EXP SIM INT APP	3c	S_ESPEXE
MU=	64c *	*
UPR=	64c *	*
TSK=	64c *	*
NUPR= NSEQ=	7n *	
SES=	64c *	*
NSESS	7n *	*
OUTPUT=file_name		

Exemple :

```
uxshw NOTE NSEQ=*
```

uxdlit NOTE: supprime une note.

Item	Format	Défaut
EXP SIM INT APP	3c	S_ESPEXE
NUPR= NSEQ=	7n *	

Exemple :

```
uxdlit note nupr=2
```

uxlst NOTE: liste toutes les notes.

Item	Format	Défaut
EXP SIM INT APP	3c	S_ESPEXE
MU=	64c *	*
UPR=	64c *	*
TSK=	64c *	*
NUPR= NSEQ=	7n *	
SES=	64c *	*
NSESS	7n *	*
INF FULL		INF
OUTPUT=file_name		

Exemple :

```
uxlst note OUTPUT=c:\temp\LstNote.txt
```

4.6.1 Description des items

NOTE : Texte de la note qui doit être associé au job. Format : 1024 caractères alphanumériques. Obligatoire avec uxadd et uxupd.

NUPR : Numéro d'exécution de l'Uproc. Format : 7 caractères numériques. Obligatoire pour uxadd, uxupd, uxshw et uxdlit si NSEQ est absent.

NSEQ : Numéro de lancement. Format : 7 caractères numériques. Obligatoire pour uxadd, uxupd, uxshw et uxdlit si NUPR est absent, si NUPR est nul ou si les lancements associés ont le statut "Horaire dépassé".

MU : filtre sur l'Unité de Gestion du job. Format : 64 caractères alphanumériques ou *.

UPR : filtre sur l'Uproc. Format : 64 caractères alphanumériques ou *.

SES : filtre sur la session. Format : 64 caractères alphanumériques ou *.

NSESS : filtre sur le numéro d'exécution de la session. Format : 7 caractères numériques ou *.

TSK : filtre sur la Tâche. Format : 64 caractères alphanumériques ou *.

INF : Option d'affichage des informations. Valeur par défaut. Autre valeur : FULL : Permet d'accéder à l'ensemble des informations.

 INF et FULL sont incompatibles entre eux.

OUTPUT : Fichier de stockage des résultats. Si l'item n'est pas renseigné la sortie se fait à l'écran.

4.6.2 Exemples

```
uxadd note nupr=12345 note=\"texte de la note\"
```

Crée une note pour l'exécution dont le numéro d'Uproc est 12345.

```
uxadd note nseq=123456 note=\"lancement en attente\"
```

Crée une note pour le lancement dont le numéro de lancement est 123456.

4.7 Les plages de maintenance

uxadd OUT : crée une plage de maintenance dans la liste des plages de maintenance.

Item	Format	Défaut
EXP SIM INT APP	3c	S_ESPEXE
TYPE=G T M	1c	
MU=	64c	
MUT=	1c	
LABEL=	64c	" "
OSTART=	(U_FMT_DATE,hhmm)	
OEND=	(U_FMT_DATE,hhmm)	

Exemple :

```
uxadd OUT EXP TYPE=M MU=HEADQUART LABEL=\"2009-05-18:12-14h\" OSTART=(20090518,1200) OEND=(20090518,1400)
```

uxshw OUT: affiche les caractéristiques de toutes les pages de maintenance situées entre les items SINCE et BEFORE (si spécifiés).

Item	Format	Défaut
EXP SIM INT APP	3c	S_ESPEXE
TYPE=G T M *	1c *	
MU=	64c *	*
MUT=	1c *	*
SINCE=	(U_FMT_DATE,hhmm)	
BEFORE=	(U_FMT_DATE,hhmm)	
OUTPUT=file_name		

Exemple :

```
uxshw OUT EXP TYPE=M MU=HEADQUART SINCE=(20090101,0000) BEFORE=
(20090630,2359)
```

uxdlt OUT: supprime toutes les pages de maintenance situées entre les items SINCE et BEFORE (si spécifiés).

Item	Format	Défaut
EXP SIM INT APP	3c	S_ESPEXE
TYPE=G T M *	1c *	
MU=	64c *	*
MUT=	1c *	*
SINCE=	(U_FMT_DATE,hhmm)	
BEFORE=	(U_FMT_DATE,hhmm)	

Exemple :

```
uxdlt OUT EXP TYPE=M MU=HEADQUART SINCE=(20090101,0000) BEFORE=
(20090630,2359)
```

uxlst OUT: liste toutes les pages de maintenance situées entre les items SINCE et BEFORE (si spécifiés).

Item	Format	Défaut
EXP SIM INT APP	3c	S_ESPEXE
TYPE=G T M *	1c *	*
MU=	64c *	*
MUT=	1c *	*
INF FULL		FULL
SINCE=	(U_FMT_DATE,hhmm)	
BEFORE=	(U_FMT_DATE,hhmm)	
OUTPUT=file_name		

Exemple :

```
uxlst OUT OUTPUT=c:\temp\LstOutage.txt
```

4.7.1 Description des items

TYPE : Choix de l'Unité de Gestion cible de la plage de maintenance. Format: G, M, T ou * (* non autorisé pour uxadd).

- G: Toutes les Unités de gestion du nœud cible de la commande. Les items MU et MUT ne doivent pas être utilisés.
- M: Unité de Gestion spécifiée dans l'item MU. MUT ne doit pas être utilisé.
- T : type d'Unité de Gestion spécifié dans l'item MUT. MU ne doit pas être utilisé.

MU : Code de l'unité de gestion. Format : 64 caractères alphanumériques. Saisie d'un code unité de gestion (valeur générique acceptée sauf pour uxadd).

MUT : Code du type de l'unité de gestion. Format : 1 caractère alphanumérique. Saisie d'un code type (valeur générique acceptée sauf pour uxadd).

LABEL : Libellé descriptif de la plage de maintenance. Format : 64 caractères alphanumériques.

OSTART : Début de la fenêtre de la plage de maintenance pour la commande uxadd. Défaut : (000000,0000).

OEND : Fin de la fenêtre de la plage de maintenance pour la commande uxadd. Défaut : (99991231,2359)

SINCE : Début de la fenêtre d'observation. Permet de déterminer les plages de maintenance à observer (sont extraites toutes les plages de maintenance dont la fenêtre présente une intersection non vide avec la plage d'observation). Format : (U_FMT_DATE,HHMM) le format de la date est indiqué par la valeur du paramètre U_FMT_DATE, heure et minutes sur 4 caractères. Le nombre de caractères doit être respecté exactement.

BEFORE : Fin de la fenêtre d'observation. Format : (U_FMT_DATE,HHMM) le format de la date est indiqué par la valeur du paramètre U_FMT_DATE, heure et minutes sur 4 caractères. Le nombre de caractères doit être respecté exactement.

INF : Option d'affichage. Limite l'affichage au nom et à la description des objets. Autre valeur : FULL (par défaut), affiche toutes les informations disponibles sur la liste des objets: type, UG, type d'UG, label, fenêtre. Les items INF et FULL sont incompatibles entre eux.

OUTPUT : Fichier de stockage des résultats. Si l'item n'est pas renseigné la sortie se fait à l'écran.

4.7.2 Exemples

```
uxadd OUT EXP TYPE=M MU=HEADQUART LABEL=\" HEADQUART 2009-05-18:12-14h\"
OSTART=(20090518,1200) OEND=(20090518,1400)
```

Crée une plage de maintenance entre le 18 mai 2009 à 12h et le 18 mai 2009 à 14h pour l'unité de gestion HEADQUART dans l'espace d'exploitation.

```
uxadd OUT EXP TYPE=G LABEL=\"GENERAL 2009-05-18:14-16h\" OSTART=
(20090518,1400) OEND=(20090518,1600)
```

Crée une plage de maintenance générale dans l'espace d'exploitation entre le 18 mai 2009 à 14h et le 18 mai 2009 à 16h.

Si ces deux plages de maintenance sont créées, le 18 mai 2009 en espace exploitation,

- Les lancements de l'UG HEADQUART qui doivent être pilotés entre 12h et 14h seront reportés à 16h (cumul avec plage de maintenance générale de 14h à 16h). Certains lancements qui devaient être pilotés avant 12h pourront être reportés (selon leur durée moyenne d'exécution) après 16h.

- Les lancements des autres UG pourront être pilotés jusqu'à 14h (selon leur durée moyenne d'exécution). Les autres seront reportés à 16h.

4.8 Les exceptions

uxadd OEX : ajoute une exception dans la liste des exceptions.

Item	Format	Défaut
EXP SIM INT APP	3c	S_ESPEXE
TYPE=SKIP BYPASS HOLD JUMP		
MU= MUT=	64c 1c	
UPR=	64c	
VUPR=	3n	
TSK=	64c	
TASK_MU=	64c	
SES=	64c	" "
VSES=	3n	
NUMLANC=	7n	
NUMSESS=	7n	
NUMORDART_SE=	7n	
EVT NOEVT		NOEVT
LABEL=	64c	
PDATE=	U_FMT_DATE	
ESTART=	(U_FMT_DATE,hhmm)	NOW
EEND=	(U_FMT_DATE,hhmm)	NOW + 1 jour
MAXCOUNT=	6n	1

Exemple :

```
uxadd OEX type=SKIP MU=MY_MU1 UPR=UPRTEST
```

uxupd OEX : modifie une exception.

Item	Format	Défaut
EXP SIM INT APP	3c	S_ESPEXE
TYPE=SKIP BYPASS HOLD JUMP		
MU= MUT=	64c 1c	
UPR=	64c	
VUPR=	3n	
TSK=	64c	
SES=	64c	
VSES=	3n	

Item	Format	Défaut
NUMLANC=	7n	
NUMSESS=	7n	
NUMORDART_SE=	7n	
EVT NOEVT		
LABEL=	64c	
PDATE=	U_FMT_DATE	
ESTART=	(U_FMT_DATE,hhmm)	
EEND=	(U_FMT_DATE,hhmm)	
LSTART=	(U_FMT_DATE,hhmm)	
LEND=	(U_FMT_DATE,hhmm)	
MAXCOUNT=	6n	

Exemple :

```
uxupd oex UPR=UPRTEST MU=MY_MU1 Label=Exception
```

uxdlit OEX : supprime une exception.

Item	Format	Défaut
EXP SIM INT APP	3c	S_ESPEXE
TYPE=SKIP BYPASS HOLD JUMP		
MU= MUT=	64c 1c	
UPR=	64c	
VUPR=	3n	*
TSK=	64c	*
SES=	64c	*
VSES=	3n	*
NUMLANC=	7n	*
NUMSESS=	7n	*
NUMORDART_SE=	7n	*
PDATE=	U_FMT_DATE	*
ESTART=	(U_FMT_DATE,hhmm)	*
EEND=	(U_FMT_DATE,hhmm)	*

Exemple :

```
uxdlit oex MU=MY_MU1 upr=UPRTEST type=SKIP
```

uxlst OEX : liste les exceptions.

Item	Format	Défaut
EXP SIM INT APP	3c	S_ESPEXE
TYPE=SKIP BYPASS HOLD JUMP		*
MU= MUT=	64c 1c	*

Item	Format	Défaut
UPR=	64c	*
VUPR=	3n	*
TSK=	64c	*
SES=	64c	*
VSES=	3n	*
NUMLANC=	7n	*
NUMSESS=	7n	*
NUMORDART_SE=	7n	*
PDATE=	U_FMT_DATE	*
ESTART=	(U_FMT_DATE,hhmm)	*
EEND=	(U_FMT_DATE,hhmm)	*
FULL INF		INF
OUTPUT=file_name		

Exemple :

```
uxlst oex full output=c:\temp\Lst0ex.txt
```

4.8.1 Description des items

TYPE : type de l'exception. Obligatoire pour uxadd. Saisir l'un des quatre mot clé ci-dessous :

- SKIP : lancement avec examen des conditions mais sans exécution du script
- BYPASS : lancement sans pilotage (sans examen des conditions)
- HOLD : lancement suspendu
- JUMP : pas de lancement : pas d'examen des conditions, pas d'exécution du script.

MU : Code de l'unité de gestion. Format : 64 caractères alphanumériques. Obligatoire pour uxadd (valeur générique acceptée sauf pour uxadd), incompatible avec MUT.

MUT : Code du type de l'unité de gestion. Format : 1 caractère alphanumérique. Obligatoire pour uxadd (valeur générique acceptée sauf pour uxadd), incompatible avec MU.

UPR : Code de l'Uproc. Format : 64 caractères alphanumériques. Obligatoire pour uxadd (valeur générique acceptée sauf pour uxadd).

VUPR : Version de l'Uproc. Format : 3 caractères numériques. Permet la saisie d'une version particulière d'une Uproc (inutile avec SIM et EXP).

TSK : Code de la Tâche. Format : 64 caractères alphanumériques.

TASK_MU : Code de l'Unité de Gestion de la tâche. Format : 64 caractères alphanumériques. L'Unité de Gestion de la tâche et celle de l'exception peuvent être différentes si une Unité de Gestion spécifique ou un TIH est défini dans la Session (ou dans le cas d'une tâche provoquée ou optionnelle). Si le code de la tâche n'est pas spécifié, ce champ n'est pas utilisé. Si le code de la tâche est spécifié et que ce champ est vide, le code de l'Unité de Gestion de la tâche est automatiquement utilisé comme code de l'Unité de Gestion de l'exception.

SES : Code session. Format : 64 caractères alphanumériques. Obligatoire si NUMSESS est spécifié.

VSES : Version de session. Format : 3 caractères numériques (inutile avec SIM et EXP).

NUMLANC : Numéro de lancement. Format : 7 caractères numériques. Permet la saisie d'un numéro de lancement. Ce numéro permet de différencier deux lancement portant sur la même Tâche pour la même date de traitement. Si NUMLANC est renseigné avec une valeur >0, VUPR (si différent de la valeur par défaut), SES (si nécessaire) et VSES (idem) doivent également être saisis.

NUMSESS : Numéro d'exécution de la session. Format : 7 caractères numériques. Permet la saisie du numéro d'exécution de la session. Ce numéro permet de différencier deux Lancements portant sur la même session sur la même unité de gestion pour la même date de traitement.

NUMORDART_SE : Numéro de l'Uproc dans la session. Format : 7 caractères numériques. Obligatoire si NUMSESS est spécifié et pour les exceptions de type JUMP. Ce numéro permet de différencier deux Uprocs dans la même session.

EVT : pour les exceptions de type JUMP, l'utilisateur peut choisir de créer un événement (EVT) ou pas (NOEVT), même si l'exécution n'a pas eu lieu. Par défaut aucun événement n'est créé.

LABEL : description de l'exception. Format : 64 caractères alphanumériques.

PDATE : Date de traitement associée au lancement. Le nombre de caractères doit être respecté exactement. Format indiqué par la valeur du paramètre U_FMT_DATE. La valeur dépend de la période fonctionnelle de l'Uproc.

 Si PDATE est valorisé par des zéros (00/00/0000,0000) pour les commandes uxadd et uxupd oex, l'exception sera définie sans date de traitement et sera appliquée seulement à des exécutions sans date de traitement.

ESTART :

- pour uxadd : date et heure de début de la plage d'exception. Format : (U_FMT_DATE,hhmm).
- Pour les autres commandes, filtre sur cette plage.

EEND :

- pour uxadd : date et heure de fin de la plage d'exception. Format : (U_FMT_DATE,hhmm).
- Pour les autres commandes, filtre sur cette plage.

 Si ESTART et EEND sont valorisés par des zéros (00/00/0000,0000) l'exception sera définie sans plage horaire.

LSTART : nouvelle date et heure de début de la plage d'exception. Format : (U_FMT_DATE,hhmm).

LEND : nouvelle date et heure de fin de la plage d'exception. Format : (U_FMT_DATE,hhmm).

MAXCOUNT : saisissez le nombre de lancement pour lesquels l'exception sera appliquée. Au-delà, l'exception ne sera plus appliquée même si la plage horaire est valide ou même si la date de traitement est valide.

INF : Option d'affichage des informations. Valeur par défaut. Autre valeur : FULL : Permet d'accéder à l'ensemble des informations.

 INF et FULL sont incompatibles entre eux.

OUTPUT : Stockage des résultats. Si l'item n'est pas renseigné la sortie se fait à l'écran. La description de cet item est fournie dans le dictionnaire d'items.

4.8.2 Exemples

```
uxadd OEX EXP TYPE=BYPASS UPR=D_LOAD_FIL MU=SIEGE PDATE=20110714
```

Crée une exception de type "lancement sans pilotage" pour le lancement de l'Uproc D_LOAD_FIL sur l'unité de gestion SIEGE dans l'espace d'exploitation pour une date au format AAAMMMJJ.

```
uxupd OEX EXP TYPE=JUMP UPR=D_LOAD_FIL MU=SIEGE
```

Modifie cette exception pour en changer le type à JUMP.

```
uxdlt OEX EXP TYPE=JUMP UPR=D_LOAD_FIL MU=SIEGE PDATE=20110714
```

Supprime l'exception créée et modifiée précédemment.

4.9 Les automates

Ces commandes sont présentes pour compatibilité uniquement. Elles sont sans effet. Les commandes **ux* fnc**, décrites dans le Manuel d'Installation de Dollar Universe, doivent être utilisées à la place en V6.

uxend ATM : arrête les automates.

Item	Format	Défaut
EXP SIM INT APP	3c	S_ESPEXE
CAL LAN ECH SURV ALL		ALL

Exemple :

```
uxend ATM EXP
```

uxstr ATM : démarre les automates et affecte le cycle de l'échangeur.

uxwak ATM : réveille les automates. Cette commande peut être utilisée pour réveiller un automate d'une machine distante.

Item	Format	Défaut
EXP SIM INT APP	3c	S_ESPEXE
CAL LAN ECH SURV ALL		ALL
CYCLE=	6n	120

Exemple :

```
uxstr ATM EXP LAN
uxwak ATM NODE=saturn EXP LAN
```

uxlst ATM : affiche l'état des automates et les informations liées aux dates et heures d'arrêt, de démarrage et de réveil.

Item	Format	Défaut
EXP SIM INT APP	3c	S_ESPEXE
CAL LAN ECH SURV ALL		ALL
OUTPUT=file_name		

Exemple :

```
uxlst ATM EXP OUTPUT=c:\temp\LstAtmExp.txt
```

 La commande **uxlst fnc** qui liste l'état de l'ensemble des services de Dollar Universe (automates et processus techniques) est décrite dans le Manuel d'Installation de Dollar Universe au paragraphe "Lister les services".

4.9.1 Description des items

CAL : Permet la sélection de l'automate calculateur.

LAN : Permet la sélection de l'automate lanceur.

ECH : Permet la sélection de l'automate échangeur.

SURV : Permet la sélection de l'automate surveillant.

ALL : Permet la sélection de tous les automates.

 ALL, CAL, LAN, ECH et SURV sont mutuellement incompatibles.

CYCLE : Cycle de l'échangeur. Format : 6 caractères numériques. 120 secondes par défaut. Indique, en secondes, la période à laquelle l'automate échangeur va tester le rétablissement de la communication avec le nœud pour lequel un échange est resté sans réponse. La modification de la valeur de ce cycle n'est possible que par la commande uxstr.

OUTPUT : Stockage des résultats. Si l'item n'est pas renseigné la sortie se fait à l'écran. La description de cet item est fournie dans le dictionnaire d'items.

4.9.2 Exemples

```
uxend ATM EXP ALL
```

Arrête tous les automates de l'espace exploitation.

```
uxstr ATM EXP CAL
```

Démarre l'automate calculateur de l'espace exploitation.

```
uxlst ATM EXP
```

Affiche les informations de tous les automates de l'espace exploitation.

```
uxwak ATM NODE=Asterix EXP LAN
```

Réveille l'automate lanceur sur le nœud Asterix dans l'espace exploitation.

4.10 L'historique des exécutions

uxshw HCX : affiche une exécution.

Item	Format	Défaut
EXP SIM INT APP	3c	S_ESPEXE
TSK=	64c *	*
SES=	64c *	" "
NSESS=	7n *	" "
UPR=	64c *	

Item	Format	Défaut
NUPR=	7n *	*
MU=	64c *	
STATUS=S P E F T A H O R W L *	1c *	*
USER=	64c *	*
PDATE=	U_FMT_DATE	
SINCE=	(U_FMT_DATE,hhmm)	
BEFORE=	(U_FMT_DATE,hhmm)	
HIST TECH PLPR VAR PARM		

Exemple :

```
uxshw HCX EXP UPR=IU_TEST NUPR=* MU=* TECH
```

uxlst HCX : liste les exécutions de façon plus ou moins complète en fonction des critères de sélection et des informations demandées.

Item	Format	Défaut
EXP SIM INT APP	3c	S_ESPEXE
TSK=	64c *	*
SES=	64c *	*
NSESS=	7n *	*
UPR=	64c *	*
NUPR=	7n *	*
MU=	64c *	*
STATUS=S P E F T A H O R W L *	1c *	*
USER=	64c *	*
PDATE=	U_FMT_DATE	
SINCE=	(U_FMT_DATE,hhmm)	
BEFORE=	(U_FMT_DATE,hhmm)	
MUTIME ASTIME		
OUTPUT=file_name		

Exemple :

```
uxlst HCX EXP STATUS=A OUTPUT=c:\temp\LstCt1.txt
```

4.10.1 Description des items

TSK : Code de la Tâche. Format : 64 caractères alphanumériques.

UPR : Code Uproc. Format : 64 caractères alphanumériques. Requiert la saisie d'un code Uproc (valeur générique acceptée). Obligatoire pour uxshw.

NUPR : Numéro d'exécution de l'Uproc. Format : 7 caractères numériques. Permet la saisie du numéro d'exécution de l'Uproc. Ce numéro permet de différencier deux exécutions portant sur la même Uproc au sein de la même session sur la même unité de gestion pour la même date de traitement.

SES : Code session. Format : 64 caractères alphanumériques. Requiert la saisie d'un code session (valeur générique acceptée). Valeur par défaut " ".

NSESS : Numéro d'exécution de la session. Format : 7 caractères numériques. Permet la saisie du numéro d'exécution de la session. Ce numéro permet de différencier deux exécutions portant sur la même session sur la même unité de gestion pour la même date de traitement.

MU : Code de l'unité de gestion. Format : 64 caractères alphanumériques. Requiert la saisie d'un code unité de gestion (valeur générique acceptée). Obligatoire pour uxshw.

STATUS : Etat d'une exécution. Par défaut toutes les exécutions sont considérées. Permet de sélectionner les exécutions en fonction de leur état par la saisie d'une ou plusieurs des valeurs suivantes (séparées par une virgule) :

- S : pilotage en cours
- P : attente exécution
- E : exécution en cours
- F : consignes en cours
- T : terminé
- A : incidenté
- H : suspendu (désactivé)
- O : horaire dépassé
- R : refusé au pilotage.
- W : attente d'événement
- L : attente lancement
- * : tous les états précédents y compris les reprises.

USER : Compte de soumission d'une exécution. Format : 64 caractères alphanumériques. Permet de sélectionner les exécutions en fonction du compte de soumission de celles ci. Par défaut tous les comptes de soumission sont considérées.

PDATE : Date de traitement associée à l'exécution. Le nombre de caractères doit être respecté exactement. Format indiqué par la valeur du paramètre U_FMT_DATE. La valeur dépend de la période fonctionnelle de l'Uproc.

SINCE : Début de la plage horaire d'observation. Défaut : (000000,0000). Permet de déterminer, en fonction de leur date de début d'exécution, les exécutions à observer. Format : (U_FMT_DATE,HHMM) la date est au format indiqué par la valeur du paramètre U_FMT_DATE, heure et minutes sur 4 caractères. Le nombre de caractères doit être respecté exactement.

BEFORE : Fin de la plage horaire d'observation. Défaut : (19991231,2359). Permet de déterminer, en fonction de leur date de fin d'exécution, les exécutions à observer. Format : (U_FMT_DATE,HHMM) la date est au format indiqué par la valeur du paramètre U_FMT_DATE, heure et minutes sur 4 caractères. Le nombre de caractères doit être respecté exactement.

HIST | TECH | PLPR | VAR : Option d'affichage. Obligatoire.

- HIST : permet l'affichage des traces automates de toutes les exécutions sélectionnées.
- Autre valeur : TECH : permet l'affichage des informations techniques de l'exécution.
- Autre valeur : PLPR : permet l'affichage des informations de lancement,
- Autre valeur : VAR : permet l'affichage de la valeur des variables de l'exécution.
- Autre valeur : PARM : affiche les paramètres de l'exécution.

 HIST, TECH, PLPR, PARM et VAR sont incompatibles entre eux.

ASTIME ou **MUTIME** : Heure de référence utilisée par la commande. Si **MUTIME** est utilisé, la colonne **MU TIME OFFSET** est affichée (uxlst). Si ce flag n'est pas utilisé dans la commande :

- Si le paramètre noeud **U_TREF** n'est pas défini ou vaut "**ASTIME**" : l'heure de référence est celle du serveur.
- Si le paramètre noeud **U_TREF** vaut "**MUTIME**" : l'heure de référence est celle de l'Unité de Gestion.

OUTPUT : Stockage des résultats. Si l'item n'est pas renseigné la sortie se fait à l'écran.

4.10.2 Exemples

```
uxshw HCX EXP UPR=D_LOAD_FIL NUPR=* SES=D_LOAD_BCK NSESS=* MU=SIEGE VAR
```

Affiche les variables de l'Uproc **D_LOAD_FIL** s'exécutant pour la session **D_LOAD_BCK** et l'unité de gestion **SIEGE**.

```
uxlst HCX SINCE="(06/01/2004,0000)" BEFORE="(06/30/2004,0000)"
```

Liste l'historique des exécutions de l'espace par défaut entre le 1er et le 30 juin 2004.

4.11 Le planning prévisionnel

uxlst PRV : calcule le planning prévisionnel sur la période indiquée.

Item	Format	Défaut
EXP SIM INT APP	3c	S_ESPEXE
TSK=	64c *	
SES=	64c *	
UPR=	64c *	
MU=	64c *	
DSINCE=	U_FMT_DATE	
DBEFORE=	U_FMT_DATE	
TSINCE=	hhmmss	
TBEFORE=	hhmmss	
MUTIME ASTIME		
OUTPUT=file_name		

Exemple :

```
uxlst PRV EXP SES=D* UPR=* MU=HEADQUART DSINCE=20020701 TSINCE=000000
DBEFORE=20021231 TBEFORE=000000
```

4.11.1 Description des items

TSK : Code de la Tâche (valeur générique acceptée). Format : 64 caractères alphanumériques.

UPR : Code Uproc. Requiert la saisie d'un code Uproc (valeur générique acceptée). Format : 64 caractères alphanumériques.

l'espace application entre le 1er janvier 1997 et le 31 décembre 1998.

```
session VER
```

Item	Format	Défaut
STAUPR STASES BOTH		BOTH
INF FULL		INF
OUTPUT=file_name		

Exemple :

```
uxshw STA EXP SES=D_LOAD_BCK UPR=D_LOAD_FIL MU=HEADQUART FULL
```

uxdlt STA : supprime une tâche de la liste des statistiques.

Item	Format	Défaut
EXP SIM INT APP	3c	S_ESPEXE
SES=	64c *	*
UPR=	64c *	
MU=	64c *	
TSK=	64c *	
STAUPR STASES BOTH		BOTH

Exemple :

```
uxdlt STA SES=D_LOAD_BCK UPR=D_LOAD_FIL MU=HEADQUART
```

uxlst STA : liste les statistiques.

Item	Format	Défaut
EXP SIM INT APP	3c	S_ESPEXE
SES=	64c *	*
UPR=	64c *	*
MU=	64c *	*
TSK=	64c *	*
FULL INF		FULL
STAUPR STASES BOTH		BOTH
OUTPUT=file_name		

Exemple :

```
uxlst STA FULL OUTPUT=c:\temp\LstSta.txt
```

Familles

Une seule des familles décrites ci-dessous peut être utilisée dans une commande uxupd (obsolète).

- **STATIST add** : ajoute les statistiques d'une exécution à une tâche : temps CPU, elapsed ou page faults.
- **STATIST upd** : modifie les statistiques d'une exécution : temps CPU, elapsed ou page faults.
- **STATIST dlt** : supprime une exécution des statistiques: temps CPU, elapsed ou page faults.

4.12.1 Description des items

UPR : Code Uproc. Format : 64 caractères alphanumériques. Requiert la saisie d'un code Uproc (valeur générique acceptée sauf pour uxadd) obligatoire.

SES : Code session. Format : 64 caractères alphanumériques. Requiert la saisie d'un code session (valeur générique acceptée sauf pour uxadd).

MU : Code de l'unité de gestion. Format : 64 caractères alphanumériques. Requiert la saisie d'un code unité de gestion (valeur générique acceptée sauf pour uxadd) obligatoire.

TSK : Code de la Tâche. Format : 64 caractères alphanumériques.

NBEXEC : numéro de l'exécution. Sur 2 caractères numériques, il doit être compris entre 0 et 99. Pour l'exécution de la commande "uxupd ... STATIST add ...", ce numéro doit être immédiatement supérieur au dernier numéro d'exécution enregistré pour cette tâche, sinon l'exécution de la commande est refusé. Les numéros d'exécution existant des statistiques d'une tâche peuvent être affichés par une commande "uxshw STA ... FULL".

CPU : temps CPU d'une exécution exprimé en secondes. Sur 9 caractères numériques.

ELAPS : temps ELAPS d'une exécution exprimé en secondes. Sur 9 caractères numériques.

PGF : Non utilisé sous Windows et Unix. Page faults d'une exécution (sur OpenVMS uniquement). Sur 9 caractères numériques.

INF : Option d'affichage des informations. Valeur par défaut de la commande uxshw.

- Utilisé avec la commande uxlst permet l'affichage de l'identifiant du job et du nombre d'exécutions pour les statuts Exécution en cours (running), Terminé (completed) et incidenté (aborted).
- Utilisé avec la commande uxshw permet l'affichage de l'identifiant du job et du nombre d'exécutions pour les statuts Attente d'événement (event wait); attente d'exécution (pending), Exécution en cours (running), Incidenté (aborted), Terminé (completed), horaire dépassé (overrun) et Refusé (refused) ainsi que les données temps CPU et page faults.

FULL : Option d'affichage des informations. Valeur par défaut de la commande uxlst.

- Utilisé avec la commande uxlst complète l'affichage avec les états Horaire dépassé (overrun), Refusé (refused), le temps CPU et les page faults.
- Utilisé avec la commande uxshw complète l'affichage avec le temps total et le nombre d'exécutions.

 INF et FULL sont incompatibles entre eux.

STAUPR : filtre les statistiques d'Uprocs, incompatible avec STATSES et BOTH

STASES : filtre les statistiques de Sessions, incompatible avec STATUPR et BOTH

BOTH : valeur par défaut, extrait les statistiques d'Uprocs et de Sessions, incompatible avec STATUPR et STATSES

OUTPUT : Stockage des résultats. Si l'item n'est pas renseigné la sortie se fait à l'écran. La description de cet item est fournie dans le dictionnaire d'items.

4.12.2 Exemples

```
uxlst STA EXP UPR=D_LOAD_FIL MU=SIEGE
```

Crée une entrée dans la liste des statistiques pour l'Uproc D_LOAD_FIL sur l'unité de gestion SIEGE dans l'espace d'exploitation.

5 Gestion de batch

Ce chapitre présente les commandes utilisées pour contrôler les traitements batch.

Le déclenchement de travaux batchs par Dollar Universe peut être effectué par différents moyens :

- Une tâche planifiée sera pilotée par l'ordonnanceur lorsque sa date et son heure de planification coïncideront avec la date et l'heure système,
- Une tâche provoquée ou planifiée pourra être déclenchée par une intervention de l'opérateur qui demande la création ou la modification d'un lancement,
- Une tâche provoquée ou planifiée pourra être déclenchée par l'exécution de la commande `uxordre`.

Dans tous les cas, le déclenchement de la tâche débute (lorsque la plage horaire d'exécution de la tâche est valide) par l'examen de ses conditions d'exécution lors de la phase de pilotage. Si ces conditions sont satisfaites, la tâche sera effectivement exécutée.

La durée d'exécution du batch pourra être surveillée et interrompue si elle excède le temps maximum indiqué. Cette fonction est obtenue par la commande `uxsurjob`.

5.1 Déclenchement de batch `uxordre`

Dollar Universe fournit une commande permettant de déclencher des traitements depuis l'environnement CL ou un script.

La commande **`uxordre`** (alias `uxord` sous OpenVMS) permet de déclencher une tâche existante dans le paramétrage de Dollar Universe et définie comme étant une tâche provoquée ou planifiée. La tâche pourra être déclenchée avec ses paramètres de définition, ou en y substituant explicitement certains paramètres décrits ci-après.

Rappel : Si la tâche est :

- Activée : la commande **`uxordre`** crée un lancement actif.
- Simulée : la commande **`uxordre`** crée un lancement désactivé.
- Désactivée : la commande **`uxordre`** ne crée pas de lancement.

La commande **`uxordre`** permet également le passage de paramètres applicatifs (jusqu'à trente paramètres) :

- à la procédure, si la tâche est une Uproc simple,
- à l'Uproc entête de la session, si la tâche porte sur une session.

La commande affiche dans la sortie standard le numéro de lancement associé avec le format suivant : "numlanc nnnnnnn".

Commande	Item	Format	Défaut
<code>uxordre</code>	EXP SIM INT APP	3c	S_ESPEXE
	TSK=	64c	TSK(SES)(VSES)(UPR) (VUPR)
	SES=	64c	" "
	VSES=	3n	
	UPR=	64c	

Commande	Item	Format	Défaut
	VUPR=	3n	
	UG= MU=	64c	
	AFTER=	"U_FMT_DATE-hhmm"	date et heure système
	ELAPS=	"(hhmm)"	99959
	EXCLU=	"(hhmm,hhmm)"	(0000,0000)
	QUEUE=	31c	Tâche
	PRINTER=	4c	Tâche
	DTRAIT= PDATE=	U_FMT_DATE	Tâche
	FORCE NOFORCE		NOFORCE
	BYPASS		
	PARAM=P1,P2,...	255c	
	LAUNCHVAR="var=val,..."	255c	

La commande OS/400 UXORDRE permet de déclencher une tâche existante à partir de la ligne de commande. F4 permet de saisir les paramètres unitairement. F11 donne le nom de chaque paramètre.

Exemples :

```
uxordre EXP UPR=1ST_JOB MU=SATURN PARAM=34,/tmp,type:2
uxordre UPR=0K2 MU=FSOLAPTOP LAUNCHVAR="test=one,test2=two"
```

Description des items

TSK : Code de la Tâche. Format : 64 caractères alphanumériques. Les tâches issues d'une mise à jour de v5 à v6 ont pour identifiant par défaut : TSK(SES)(VSES)(UPR)(VUPR)

UPR : Code Uproc. Format : 64 caractères alphanumériques. Requiert la saisie d'un code Uproc obligatoire.

VUPR : Version d'Uproc. Format : 3 caractères numériques. Permet la saisie d'une version particulière d'une Uproc. Par défaut, toutes les versions des Uprocs renseignées sont considérées (inutile avec les espaces S et X).

SES : Code session. Format : 64 caractères alphanumériques. Permet la saisie d'un code session.

VSES : Version de session. Format : 3 caractères numériques. Permet la saisie d'une version particulière d'une session. Par défaut, toutes les versions des sessions renseignées sont considérées (inutile dans les espaces S et X).

MU (ou **UG**) : Code unité de gestion. Format : 64 caractères alphanumériques. Requiert la saisie d'un code unité de gestion (valeur générique acceptée) obligatoire.

AFTER : Date et heure de début de plage de lancement. Permet de déclencher un traitement en différé. Par défaut : pour une tâche provoquée : les informations correspondantes dans la tâche; pour une tâche planifiée : soumission immédiate.

Format : U_FMT_DATE-HHMM, le format de la date est indiqué par la valeur du paramètre U_FMT_DATE, l'heure est sur 4 caractères. Sous OS/400, la date doit être saisie au format utilisé par le système. Le format de l'heure est HHMMSS.

ELAPS : Durée de la plage de lancement. Format : (HHHMM).

- Sous Windows/UNIX/OpenVMS: valeur par défaut : 99959 (999h 59m soit 41,6 jours).
- Sous OS/400 : valeur par défaut : l'information correspondante dans la tâche.

EXCLU : Plage horaire d'exclusion (sous Windows/UNIX). Format : (HHMM,HHMM). Défaut : vide. Permet d'exclure un intervalle horaire de la plage de lancement.

EXLW : Plage horaire d'exclusion (sous OS/400). Format : (HHMMSS HHMMSS). Permet d'exclure un intervalle horaire de la plage de lancement. Défaut : l'information correspondante dans la tâche.

QUEUE : Queue batch d'exécution du lancement. Format : 32 caractères alphanumériques. Défaut : l'information correspondante dans la tâche. Doit exister au niveau du système d'exploitation.

PRINTER : Imprimante attachée à la tâche. Format : nom logique à quatre caractères. Défaut : l'information correspondante dans la tâche.

CENTRAL : Obsolète.

FORCE : Indicateur de "lancement forcé en fin de période".

- Sous Windows/UNIX/OpenVMS : valeur par défaut : NOFORCE.
- Sous OS/400 : valeur par défaut : l'information correspondante dans la tâche.

BYPASS : Permet d'effectuer un lancement sans vérifier les conditions de l'Uproc.

DTRAIT ou **PDATE** : Date de traitement. Défaut : l'information correspondante dans la tâche.

- Sous Windows/UNIX/OpenVMS : format indiqué par la valeur du paramètre U_FMT_DATE.
- Sous OS/400, la date doit être saisie au format utilisé par le système.

PARAM : Paramètres applicatifs. Permet le passage de 1 à 30 paramètres à la première Uproc de la tâche. Ces paramètres peuvent être récupérés :

- Dans le CL de l'Uproc, sous la forme \$1 à \$30, ou sous la désignation Dollar Universe S_P1 à S_P30,
- Dans les scripts lancés par la coquille batch : U_ANTE_UPROC et U_POST_UPROC, sous la désignation Dollar Universe.

Le nombre maximum de paramètres transmis est de 30. La longueur de chaque paramètre ne doit pas excéder 255 caractères et la longueur totale (somme des longueurs + 2 fois le nombre de paramètres) doit être inférieure à 2074 caractères.

Format : les paramètres saisis sont séparés par des virgules, chaque paramètre peut avoir un format quelconque sauf s'il comporte un espace, il faut alors l'indiquer entre guillemets.

LAUNCHVAR : Liste complète des variables du lancement et de leurs valeurs. Format :

"var=val,var=val,...", 255 caractères maximum. La liste est encadrée de guillemets, chaque couple variable = valeur est séparé du suivant par une virgule (sans espace).

- Le nom de la variable doit être choisi dans la liste des variables de l'Uproc identifiant le lancement.
 - La valeur de la variable, si elle est indiquée, remplace la valeur de la variable de la tâche ou de l'Uproc. Si la valeur n'est pas indiquée (la virgule ne doit pas être présente) c'est la valeur par défaut de la variable de la tâche ou de l'Uproc qui est prise en compte. Si la valeur contient des caractères non alphanumériques (espaces, /...) elle doit être encadrée de guillemets. Le format de la valeur doit être compatible avec le format déclaré lors
-

Cette commande déclenche la tâche provoquée "IU_DTS S01", existante sur le nœud ds5200 dans l'espace d'Exploitation avec ses paramètres de définition et passe à la procédure IU_DTS les paramètres suivants : 34, /tmp, type:2.

5.2 Surveillance de la durée d'exécution

 Ces commandes sont obsolètes et maintenues uniquement pour compatibilité.

La durée d'exécution d'un batch (temps elapsed) peut être surveillée par les commandes `uxsurjob`, `uxspvjob` ou `uxalrjob`.

- **uxsurjob** permet d'interrompre l'exécution dès lors que le temps dépasse celui indiqué en paramètre à la commande.
- **uxspvjob** permet d'exécuter un script utilisateur au-delà du temps indiqué (précis à la minute).
- **uxalrjob** permet d'exécuter un script utilisateur au-delà du temps indiqué (précis à la seconde).

5.2.1 uxsurjob

Lorsque le temps indiqué est écoulé, Dollar Universe tue le batch (kill) et tous les sous processus qui en sont issus dans la mesure où ceux ci y sont rattachés.

Cette commande peut être introduite dans le prétraitement U_ANTE_UPROC (U_ANTE_UPROC.bat sous Windows), du répertoire UXMGR, si l'on souhaite qu'aucune exécution de l'environnement de la société et du nœud ne dépasse un temps imparti, ou bien dans le script de l'Uproc si la durée maximum d'exécution est spécifique pour chaque Uproc ou si la surveillance ne porte que sur certaines exécutions.

Commande	Item
uxsurjob	NB_SEC \$S_NUMPROC %S_NUMPROC%

Elle doit être lancée en arrière plan en utilisant le signe & sous UNIX.

NB_SEC : Temps maximum d'exécution. Indique (en secondes) quel est le temps elapsed maximum que l'exécution ne devra pas dépasser.

S_NUMPROC : est le numéro d'exécution de l'Uproc en cours d'exécution, cette variable est automatiquement renseignée par Dollar Universe.

- Sous Windows : %S_NUMPROC%
- Sous UNIX : \$S_NUMPROC
- Sous OpenVMS: 'S_NUMPROC'

Exemple

```
uxsurjob 3600 %S_NUMPROC%(sous Windows)
uxsurjob 3600 $S_NUMPROC &(sous UNIX)
$ uxsurjob 3600 'S_NUMPROC'(sous OpenVMS)
```

Si cette commande est exécutée dans U_ANTE_UPROC, cela aura pour conséquence que toutes les exécutions dépassant une heure d'exécution seront tuées.

Si elle est introduite dans le script d'une Uproc, elle aura pour conséquence de tuer l'exécution de l'Uproc si celle-ci dépasse une heure de temps elapsed.

5.2.2 uxspvjob

Lorsque le temps indiqué est écoulé, Dollar Universe lance le script utilisateur U_SPV_UPROC situé dans le répertoire UXMGR de la société.

Cette commande peut être introduite dans le prétraitement U_ANTE_UPROC si l'on souhaite appliquer le même traitement à toutes les Uprocs qui dépassent un temps imparti, ou bien dans le script de l'Uproc si la durée surveillée est spécifique pour chaque Uproc ou si la surveillance ne porte que sur certaines exécutions.

Sous Windows, la commande %UNI_DIR_EXEC%\U_SPV_UPROC doit avoir une extension .bat ou .cmd et sous OpenVMS le script @UXMGR:U_SPV_UPROC doit avoir une extension .com..

Commande	Item
uxspvjob	NB_SEC

Elle doit être lancée en arrière plan en utilisant le signe & sous UNIX.

NB_SEC : Temps maximum. Indique en secondes quel est le temps elapsed maximum au delà duquel le script devra être lancé. Le temps indiqué est converti en minutes et arrondi au nombre de minutes supérieur. Par exemple :

- 5s1 min
- 58s1 min
- 67s2 min

Exemple

```
uxspvjob 550(sous Windows)
uxspvjob 550(sous UNIX)
$ SPAWN/NOWAIT uxspvjob 550 (sous OpenVMS)
```

Au-delà de 10 minutes d'exécution de l'Uproc, le script U_SPV_UPROC sera exécuté.

5.2.3 uxalrjob

Cette commande peut être introduite dans le prétraitement U_ANTE_UPROC si l'on souhaite appliquer le même traitement à toutes les Uprocs qui dépassent un temps imparti, ou bien dans le script de l'Uproc si la durée surveillée est spécifique pour chaque Uproc ou si la surveillance ne porte que sur certaines exécutions.

Lorsque le temps indiqué est écoulé, Dollar Universe lance par défaut le script utilisateur U_ALR_JOB (U_ALR_JOB.bat ou U_ALR_JOB.cmd sous Windows) situé dans le sous-répertoire data/nodefiles (UNI_DIR_NODEFILE) de la Société.

Le nom du script peut néanmoins être nommé autrement en valorisant la variable U_ALR_JOB.

Huit paramètres sont passés au script U_ALR_JOB lors de l'exécution de la commande uxalrjob :

- Nom de la session ou " ".
- Numéro d'exécution de la session.
- Nom de l'Uproc.
- Numéro d'exécution de l'Uproc.
- Unité de gestion.
- Numéro de lancement.

- Date de traitement.
- Nombre de secondes de l'alarme.

Ces paramètres peuvent être utilisés par le script sous la forme usuelle: \$1, \$2 ... sous UNIX; %1, %2... sous Windows, P1, P2... sous OpenVMS.

Commande	Item
uxalrjob	NB_SEC

```
uxalrjob 10      sur Windows
uxalrjob 10 &    sur UNIX
$ uxalrjob 10     sur OpenVMS
```

NB_SEC : Durée maximum. Indique en secondes quel est le temps elapsed maximum au-delà duquel le script U_ALR_JOB devra être exécuté.

Exemple

Sous Windows :

Script interne de l'Uproc (au delà de 20 secondes d'exécution de l'Uproc, le script %UXMGR%\U_ALR_JOB.bat sera exécuté):

```
%UNI_DIR_EXEC%\uxalrjob 20
set RESEXE=0
```

Script %UXMGR%\U_ALR_JOB.bat (affiche les paramètres dans le fichier %UXMGR%\alr.log):

```
echo %1 >>%UXMGR%\alr.log
echo %2 >>%UXMGR%\alr.log
echo %3 >>%UXMGR%\alr.log
echo %4 >>%UXMGR%\alr.log
echo %5 >>%UXMGR%\alr.log
echo %6 >>%UXMGR%\alr.log
echo %7 >>%UXMGR%\alr.log
echo %8 >>%UXMGR%\alr.log
exit 0
```

Trace automate d'exécution de l'Uproc :

```
07/07/2006 11:13:53 Start Condition Check
                      Modified directly in the Engine by
                      operator : eca
                      Processing Date 07/07/2006
07/07/2006 11:13:53 Submitted in BATCH Q: SYS_BATCH
                      under ENTRY N° : 0144

BATCH starting
Maximum allowed elapsed time reached : 00000020 seconds . U_ALR_
JOB submitted
Start Completion Instructions
07/07/2006 11:14:25 *** NORMAL TASK COMPLETION ***
```

Fichier %UXMGR%\alr.log

```
"          "
"00000000"
"UPR_ALRJOB"
"0000144"
```

```
"SATURN      "  
"0000120"  
"20060707"  
"20"
```

6 Commandes FTP

Ce chapitre présente les commandes utilisées pour contrôler les transferts de fichiers via FTP.

 Non disponible sous OS/400

La commande `uxstr FTP` permet de déclencher un transfert FTP en utilisant les mécanismes de Dollar Universe Manager for File Transfer.

Selon que le transfert est une récupération ou un envoi de fichier les items suivants doivent être utilisés :

- `uxstr FTP GET` : pour récupérer un fichier par FTP.
- `uxstr FTP PUT` : pour envoyer un fichier par FTP.

Ces commandes ne peuvent être utilisées que dans le script d'une Uproc ou à partir d'une Uproc de type CMD.

6.1 `uxstr FTP GET`

Cette commande permet de récupérer un ou plusieurs fichiers sur un serveur FTP, en utilisant les fonctionnalités proposées par Dollar Universe Manager for File Transfer.

Commande	Item	Format	Valeur par défaut
uxstr FTP GET	PARTNER=	16c	
	ORIGIN_FILENAME=	256c	""
	ORIGIN_PATHNAME=	256c	
	TARGET_FILENAME=	256c	
	TARGET_PATHNAME=	256c	
	LOGON=	32c	anonymous
	PASSWORD=	32c	
	MODE=ASCII BINARY		ASCII
	ACTION=REPLACE ABORT NONE		ABORT
	CRLF NOCRLF		NOCRLF
	DELETE		NODELETE
	NEWER		NONEWER
	RECURSIVE		NORECURSIVE
	BUFFER_SIZE=	2n	Défaut FTP
	RETRY=	1c	
	UMASK=	3c	

Description des items

PARTNER: Nom physique de la machine cible ou adresse IP.

ORIGIN_FILENAME : Nom du fichier distant à récupérer. Par défaut tous les fichiers du répertoire.

ORIGIN_PATHNAME : Répertoire distant. Par défaut, répertoire de connexion de l'utilisateur FTP. Caractère * interdit.

 Notez que les chemins doivent être écrits comme des chemins de commande FTP, par exemple le chemin "C:\Temp\Transfer" doit être écrit: "C:/Temp/Transfer".

TARGET_FILENAME : Nom du fichier local. Inchangé par défaut.

TARGET_PATHNAME : Nom du répertoire local de réception des fichiers. Par défaut nom du répertoire de connexion du compte de soumission de l'Uproc. Caractère * interdit.

LOGON : Utilisateur de connexion FTP.

PASSWORD : Mot de passe de l'utilisateur de connexion FTP. Le mot de passe est invisible dans l'interface graphique.

MODE : Type de transfert : ASCII ou BINARY

ACTION : Comportement si le fichier cible existe déjà : REPLACE (remplacer), ABORT (annuler erreur) ou NONE (ne rien faire).

CRLF: Tous les caractères CR sont convertis en LF durant le transfert ASCII. Par défaut pas de conversion sauf pour les soumissions.

DELETE : Supprime le fichier origine si le transfert est réussi. Par défaut pas de suppression.

NEWER : Transfère le fichier si sa date de modification est plus récente que celle du fichier cible.

RECURSIVE : Transfère tous les fichiers et sous-répertoires contenus dans le chemin ORIGIN_PATHNAME (les paramètres ORIGIN_FILENAME et TARGET_FILENAME ne doivent plus être saisis). Si cette option est utilisée, le nombre de niveaux de répertoires ne doit pas dépasser 150. NORECURSIVE par défaut.

L'option RECURSIVE utilise en interne la commande FTP LIST pour obtenir la liste des fichiers et des répertoires du serveur distant. La réponse attendue du serveur doit être formatée de telle sorte que chaque ligne doit commencer par les droits des fichiers au format UNIX. Cette réponse variant d'un serveur à un autre, la fonction ne pourra pas être utilisée si la réponse a un format différent. Dans ce cas, un fonctionnement dégradé peut être utilisé : chaque fichier du serveur distant est testé à l'aide d'une commande FTP CWD pour vérifier l'existence d'un répertoire. Ce fonctionnement dégradé est moins performant mais il peut être activé à l'aide du paramètre noeud U_FT_GET_DIR_FAILOVER dans la catégorie "Paramètres noeud – Paramétrage FTP".

 Lors de la récupération d'un répertoire en mode RECURSIVE à l'aide d'une Uproc FTP_GET, le mode Append n'est pas supporté, l'action Abort est utilisée par défaut.

BUFFER_SIZE : Définit la taille du buffer de communication en blocs de 512 octets. Si la valeur est définie, le format est un nombre entier, valeur minimum 1, valeur maximum 64.

RETRY : Nombre d'essais en cas d'échec de transmission, sur 1 caractère.

UMASK : Masque appliqué aux permissions du fichier local. Format : 3 caractères numériques. Si UMASK n'est pas défini ou est défini à 000, les fichiers sont téléchargés par défaut avec les permissions 666 (pour un utilisateur root) ou 640 (pour un utilisateur non-root). Sinon, les permissions du fichier sont obtenues en soustrayant UMASK de la valeur par défaut (666 ou 640).

Exemple

```
uxstr FTP GET PARTNER="WinXP" ORIGIN_PATHNAME="/tmp/" CRLF MODE=BINARY
UMASK=700 ACTION=CANCEL RETRY=3
```

6.2 uxstr FTP PUT

Cette commande permet d'envoyer un ou plusieurs fichiers sur un serveur FTP, en utilisant les fonctionnalités proposées par Dollar Universe Manager for File Transfer.

- OS Linux et assimilés : les fichiers sont envoyés avec leurs permissions d'origine.
- Windows : les fichiers sont envoyés avec une permission de 755 (la permission en octal n'existant pas sous Windows).

 L'option CRLF n'est pas disponible pour FTP PUT.

Commande	Item	Format	Valeur par défaut
uxstr FTP PUT	PARTNER=	16c	
	ORIGIN_FILENAME=	256c	""
	ORIGIN_PATHNAME=	256c	
	TARGET_FILENAME=	256c	
	TARGET_PATHNAME=	256c	
	LOGON=	32c	anonymous
	PASSWORD=	32c	
	MODE=ASCII BINARY		ASCII
	ACTION=REPLACE ABORT APPEND NONE		ABORT
	DELETE		NODELETE
	NEWER		NONEWER
	RECURSIVE		NORECURSIVE
	BUFFER_SIZE=	2n	Défaut FTP
	RETRY=	1c	

Description des items

PARTNER : Nom physique de la machine cible ou adresse IP.

ORIGIN_FILENAME : Nom du fichier local à envoyer sur le serveur FTP. Par défaut tous les fichiers du répertoire.

ORIGIN_PATHNAME : Nom du répertoire local à partir duquel les fichiers sont envoyés. Par défaut répertoire de connexion de l'utilisateur FTP. Caractère * interdit.

 Notez que les chemins doivent être écrits comme des chemins de comande FTP, par exemple le chemin "C:\Temp\Transfer" doit être écrit: "C:/Temp/Transfer".

TARGET_FILENAME : Nom du fichier sur le serveur FTP, uniquement si un seul fichier doit être envoyé. Inchangé par défaut.

TARGET_PATHNAME : Nom du répertoire de destination sur le serveur FTP dans lequel les fichiers doivent être envoyés. Par défaut, répertoire de connexion du compte de soumission de l'Uproc. Caractère * interdit.

LOGON : Utilisateur de connexion FTP.

PASSWORD : Mot de passe de l'utilisateur de connexion FTP. Le mot de passe est invisible dans l'interface graphique.

MODE : Type de transfert : ASCII ou BINARY

ACTION : Comportement si le fichier cible existe déjà : REPLACE (remplacer), ABORT (annuler erreur), APPEND (concaténer) ou NONE (ne rien faire).

DELETE : Supprime le fichier origine si le transfert est réussi. Par défaut pas de suppression.

NEWER : Transfère le fichier si sa date de modification est plus récente que celle du fichier cible.

RECURSIVE : Transfère tous les fichiers et sous-répertoires contenus dans le chemin ORIGIN_PATHNAME (les paramètres ORIGIN_FILENAME et TARGET_FILENAME ne doivent plus être saisis). Si cette option est utilisée, le nombre de niveaux de répertoires ne doit pas dépasser 150. NORECURSIVE par défaut.

L'option RECURSIVE utilise en interne la commande FTP LIST pour obtenir la liste des fichiers et des répertoires du serveur distant. La réponse attendue du serveur doit être formatée de telle sorte que chaque ligne doit commencer par les droits des fichiers au format UNIX. Cette réponse variant d'un serveur à un autre, la fonction ne pourra pas être utilisée si la réponse a un format différent. Dans ce cas, un fonctionnement dégradé peut être utilisé : chaque fichier du serveur distant est testé à l'aide d'une commande FTP CWD pour vérifier l'existence d'un répertoire. Ce fonctionnement dégradé est moins performant mais il peut être activé à l'aide du paramètre noeud U_FT_GET_DIR_FAILOVER dans la catégorie "Paramètres noeud – Paramétrage FTP".

BUFFER_SIZE : Définit la taille du buffer de communication en blocs de 512 octets. Si la valeur est définie, le format est un nombre entier, valeur minimum 1, valeur maximum 64.

RETRY : Nombre d'essais en cas d'échec de transmission sur 1 caractère.

Exemple

```
uxstr FTP PUT PARTNER="hp10201" ORIGIN_FILENAME="data.txt" TARGET_
PATHNAME="/tmp/" ACTION=ABORT
```

7 Commandes DQM

Ce chapitre présente les commandes utilisées pour contrôler la soumission des traitements batch.

7.1 Créer une queue batch

La commande **uxaddque** permet de définir une nouvelle queue batch physique ou une queue logique. La queue batch est créée inactive, il conviendra donc de la démarrer avant de pouvoir l'utiliser.

 Lorsqu'on associe les queues physiques à une queue logique, il n'y a pas de vérification que la queue physique existe réellement car elle peut avoir été définie sur un nœud distant. Il faut donc s'assurer préalablement que les queues physiques citées existent réellement.

La syntaxe de cette commande est la suivante :

Commande	Item	Format
uxaddque	SOC=	6 caractères.
	QUEUE=	31 caractères*
	GENE	
	LSTQUE=	"noeud1:queue1,noeud2:queue2,..."
	NODE=	64 caractères max. Défaut : S_NOEUD ou S_NODENAME
	JOBLIM=	[0>999] défaut : 0
	MAXSEC=	n
	PERSEC=	n
	BASPRI=	[0->255] défaut : 0
	ADDPRI=	[0->255] défaut : 0
	MAXPRI=	[0-255] défaut : 0
	EXEPRI=	[0-255] défaut : 0
	[NO]SECPRI	
	LSTMU=	"UG1:NbJobs, UG2:NbJobs,..."

Description des items

SOC : Environnement d'exécution de la commande.

QUEUE : Nom de la queue batch.

GENE : Indicateur de création d'une queue batch logique. Incompatible avec **NODE**.

LSTQUE : Liste des queues physiques associées à la queue logique. Chaque queue physique doit être précédée de son nœud de résidence. Format : "noeud1:queue1,noeud2:queue2,...". L'ordre de cette liste est identique à celui utilisé par le serveur DQM pour dispatcher les jobs.

NODE : Nom du nœud sur lequel s'exécutent les jobs gérés par la queue batch physique. Par défaut, c'est le nœud courant. Incompatible avec **GENE** et **LSTQUE**.

JOBLIM : Nombre maximum de jobs pouvant s'exécuter en même temps dans la queue batch ou dans le cas d'une queue logique pour toutes les queues physiques associées. Lorsque cette limite est atteinte, les jobs entrent dans la queue dans l'état "Attente exécution".

PERSEC : Queue physique uniquement. Valeur en seconde du cycle de révision des priorités des jobs au sein de la queue batch. Toutes les persec secondes, la priorité des jobs est augmentée de addpri, à partir de la priorité de soumission de base jusqu'à au plus maxpri.

BASPRI : Queue physique uniquement. Priorité par défaut de soumission d'un job placée en queue batch. Par défaut, baspri=0.

ADDPRI : Queue physique uniquement. Valeur d'incrément de la priorité d'un job en queue batch dans le cas de la gestion dynamique des priorités.

MAXPRI : Queue physique uniquement. Valeur maximale de la priorité des jobs dans la queue batch. Si joblim le permet, les jobs possédant cette priorité sont mis en exécution.

EXEPRI : Queue physique uniquement. Priorité par défaut d'exécution des jobs dans la queue batch.

SECPRI : Active la gestion des priorités secondaires pour la queue batch. L'item NOSECPRI désactive cette gestion. La gestion des priorités secondaire signifie que des jobs ayant la même priorité de soumission seront examinés selon leur numéro de session et non leur numéro d'entrée dans la queue. Dans ce cas; les jobs d'une première session pourront se terminer avant que la seconde Session ne soit exécutée.

LSTMU : liste du nombre maximum de jobs par Unité de Gestion pour la queue. Format : "UG:NbJobs,UG:NbJobs,...". Les Unités de Gestion dans la liste seront garanties d'un nombre d'exécutions spécifié, même si des jobs sont en attente d'exécution dans la queue.

Exemples

```
$UNI_DIR_EXEC/uxaddque QUEUE="${S_CODUG}_${S_DATRAIT}" NODE="${S_NODENAME}"
JOBLIM=1 MAXSEC=3600 PERSEC=3600 BASPRI=1 ADDPRI=1 MAXPRI=255 EXEPRI=20
```

Exécutée dans le script d'une Uproc, cette commande crée une queue batch physique dont le nom est issu du code de l'unité de gestion et de la date de traitement interprétées dynamiquement lors de l'exécution de l'Uproc. Le nœud (local) est indiqué par la variable S_NODENAME, ce qui permet de distribuer ce script sans avoir à réaliser d'adaptation.

```
%UNI_DIR_EXEC%\uxaddque QUEUE=LOG_QUEUE GENE LSTQUE="saturn:SYS_
BATCH, saturn:PHYS1, uranus:PHYS2"
```

Cette commande crée dans l'environnement courant, la queue batch logique LOG_QUEUE associée aux queues batch physiques SYS_BATCH et PHYS1 sur le nœud saturn et à la queue batch physique PHYS2 sur le nœud uranus.

```
%UNI_DIR_EXEC%\uxaddque QUEUE=PHY_QUEUE LSTMU="C_UK:2, C_FRANCE:2, C_SPAIN:2"
JOBLIM=10
```

Cette commande crée une queue physique supportant 10 jobs en parallèle pour toutes les Unités de Gestion. Les UG C_UK, C_FRANCE et C_SPAIN sont garanties de pouvoir exécuter 2 jobs en parallèle si la queue est chargée. Toutes les Unités de Gestion (incluant celles listées) pourront exécuter des jobs dans la capacité restante (4 jobs).

7.2 Modifier les paramètres d'une queue batch

La commande **uxupdque** permet de modifier la définition d'une queue batch ou d'associer des queues physiques à une queue logique. Pour des raisons de cohérence, la queue batch doit être vide.

La syntaxe est la suivante :

Commande	Item	Format
uxupdque	SOC=	6 caractères.
	QUEUE=	31 caractères*
	LSTQUE=	"noeud1:queue1,noeud2:queue2,..."
	ADDQUE=	"noeud1:queue1,noeud2:queue2,..."
	DLTQUE=	"noeud1:queue1,noeud2:queue2,..."
	NODE=	64 caractères max.
	JOBLIM=	[0-999]
	MAXSEC=	n
	PERSEC=	n
	BASPRI=	[0-255]
	ADDPRI=	[0-255]
	MAXPRI=	[0-255]
	EXEPRI=	[0-255]
	[NO]SECPRI	
	ADDMU=	"UG1:NbJobs1,UG2:NbJobs2,..."
	DLTMU=	"UG1:NbJobs1,UG2:NbJobs2,..."

Description des items

SOC : Environnement d'exécution de la commande. Voir paragraphe "[Environnement](#)".

QUEUE : Nom de la queue batch à modifier (physique ou logique).

LSTQUE : Permet de remplacer toutes les relations définies pour la queue logique par les nouvelles relations définies. Format : "noeud1:queue1,noeud2:queue2,...". Incompatible avec **NODE**. L'ordre de cette liste est identique à celui utilisé par le serveur DQM pour dispatcher jobs.

ADDQUE : Permet d'associer une ou plusieurs queues physiques en plus de celles déjà associées à la queue logique. Format : "noeud1:queue1,noeud2:queue2,...". Incompatible avec **NODE**.

DLTQUE : Permet de supprimer la relation d'une ou plusieurs queues physiques à la queue logique sans supprimer toutes les relations. Format : "noeud1:queue1,noeud2:queue2,...". Incompatible avec **NODE**.

NODE : Nom du nœud sur lequel s'exécutent les jobs gérés par la queue batch physique. Par défaut, c'est le nœud courant. Incompatible avec **LSTQUE**, **ADDQUE** et **DLTQUE**.

JOBLIM : Nombre maximum de jobs pouvant s'exécuter en même temps dans la queue batch physique ou dans le cas d'une queue logique dans toutes les queues physiques associées.

MAXSEC : Queue physique uniquement. Nombre de secondes donnant le temps maximum pendant lequel le job reste en queue batch avant de prendre la priorité maximale et devenir ainsi candidat à

l'exécution. Ce temps correspond au temps maximum de rétention. Le job ne passera en exécution que si la valeur de joblim le permet.

PERSEC : Queue physique uniquement. Valeur en seconde du cycle de révision des priorités des jobs au sein de la queue batch. Toutes les persec secondes, la priorité des jobs est augmentée de addpri, à partir de la priorité de soumission de base jusqu'à au plus maxpri.

BASPRI : Queue physique uniquement. Priorité par défaut de soumission d'un job placée en queue batch. Par défaut, baspri=0.

ADDPRI : Queue physique uniquement. Valeur d'incrément de la priorité d'un job en queue batch dans le cas de la gestion dynamique des priorités.

MAXPRI : Queue physique uniquement. Valeur maximale de la priorité des jobs dans la queue batch. Si joblim le permet, les jobs possédant cette priorité sont mis en exécution.

EXEPRI : Queue physique uniquement. Priorité par défaut d'exécution des jobs dans la queue batch.

SECPRI : Active la gestion des priorités secondaires pour la queue batch. L'item NOSECPRI désactive cette gestion. La gestion des priorités secondaire signifie que des jobs ayant la même priorité de soumission seront examinés selon leur numéro de session et non leur numéro d'entrée dans la queue. Dans ce cas; les jobs d'une première session pourront se terminer avant que la seconde Session ne soit exécutée.

ADDMU : ajoute un nombre maximum de jobs par Unité de Gestion à la liste du nombre maximum de jobs par Unité de Gestion pour la queue. Format : "UG:NbJobs,UG:NbJobs,...".

DLTMU : supprime des Unités de Gestion de la liste du nombre maximum de jobs par Unité de Gestion pour la queue. Format : "UG1:NbJobs1,UG2:NbJobs2,UG3..."

Exemples

```
$UNI_DIR_EXEC/uxupdque queue="$${S_CODUG}_${S_DATRAIT}" joblim=5
```

Cette commande modifie le nombre maximum de jobs qui peuvent être exécutés en parallèle dans la queue batch créée précédemment.

```
%UNI_DIR_EXEC%\uxupdque QUEUE=LOG_QUEUE ADDQUE="uranus:PHYS3"
```

Cette commande modifie la queue batch logique LOG_QUEUE en lui ajoutant la queue batch physique PHYS3 sur le nœud uranus.

7.3 Supprimer une queue batch

La commande **uxdlitque** permet de détruire une queue batch. Cette commande ne fonctionne que si aucun job n'est présent dans la queue batch.

La syntaxe de cette commande est la suivante :

Commande	Item	Format
Commande	Item	31 caractères max.

Description des items

QUEUE : Nom de la queue batch.

Exemple

```
$UNI_DIR_EXEC/uxdl1tque QUEUE="${S_CODUG}_${S_DATRAIT}"
```

Cette commande supprime la queue batch indiquée, si celle-ci est arrêtée et qu'aucun job n'est présent.

7.4 Initialiser une queue batch

La commande **uxresetqueue** supprime tous les jobs (attente exécution, suspendus ou en cours d'exécution) d'une queue (physique ou logique). Initialiser une queue logique n'a pas d'effet sur le contenu des queues physiques associées.

La syntaxe de cette commande est la suivante :

Commande	Item	Format
uxresetque	QUEUE=	31 caractères max.

Description des items

QUEUE : Nom de la queue batch.

Exemple

```
$UNI_DIR_EXEC/uxresetque QUEUE="${S_CODUG}_${S_DATRAIT}"
```

Cette commande initialise la queue batch indiquée, tous les jobs de cette queue seront perdus.

7.5 Démarrer une queue batch

La commande **uxstrque** permet de démarrer une queue batch. Cette commande doit être utilisée si la queue batch est dans l'état inactive et que l'on souhaite exécuter des jobs sur cette queue batch.

 Le démarrage d'une queue batch logique ne permettra l'exécution d'un job que si au moins une queue batch physique associée est démarrée.

La syntaxe de cette commande est la suivante :

Commande	Item	Format
uxstrque	QUEUE=	31 caractères max.

Description des items

QUEUE : Nom de la queue batch. La valeur all permet de démarrer toutes les queues batch.

Exemple

```
$UNI_DIR_EXEC/uxstrque QUEUE=QB_PHYS_1
```

Cette commande démarre la queue batch indiquée. L'état de celle ci passe donc d'inactive à active, elle permet ainsi l'exécution de jobs sur cette queue batch.

7.6 Arrêter une queue batch

La commande **uxstpque** permet de stopper une queue batch. Cette commande doit être utilisée si la queue batch est dans l'état Démarrée et que l'on ne souhaite plus soumettre de nouveaux jobs dans cette queue batch. Les jobs de la queue batch qui étaient dans l'état "en attente" ou "suspendu" ne seront pas exécutés. Les jobs qui étaient dans l'état "en cours" ne seront pas arrêtés. Tous les jobs présents dans la queue batch au moment de l'arrêt resteront dans la queue batch.

La syntaxe de cette commande est la suivante

Commande	Item	Format
uxstpque	QUEUE=	31 caractères max.
	NEXT	

Description des items

QUEUE : Nom de la queue batch. La valeur **all** permet de stopper toutes les queues batch.

NEXT : (Sous UNIX uniquement) Indicateur spécifiant que la queue batch ne sera stoppée qu'une fois les jobs en cours d'exécution présents dans la queue batch seront terminés. Les jobs en attentes ne seront quant à eux pas lancés. Si l'option "NEXT" n'est pas spécifiée, tous les jobs présents dans la queue batch sont suspendus jusqu'au redémarrage de la queue batch.

Exemple

```
$UNI_DIR_EXEC/uxstpque QUEUE=QB_PHYS_1 NEXT
```

Cette commande arrêtera la queue batch lorsque tous les jobs qui étaient dans l'état en cours au moment de la commande seront terminés. Les autres jobs ne seront pas lancés.

7.7 Afficher la liste des queues batch

La commande **uxlstque** affiche la liste des queues batch.

La syntaxe de cette commande est la suivante :

Commande	Item	Format
uxlstque	QUEUE=	* 31 caractères max.

Description des items

QUEUE : Nom de la queue batch. L'item Queue peut contenir des caractères génériques.

Pour chaque queue batch, les caractéristiques suivantes sont affichées :

- QUEUE Name: Le nom de la queue batch
 - Type : PHYS (queue physique) ou LOGI (queue logique)
-

- STA: le statut de la queue batch ON (démarrée) ou OFF (arrêtée)
- JOBLIM : nombre maximum de jobs dans la queue batch
- JOBQUE : nombre de jobs dans la queue
- JOBEXE : nombre de jobs en cours d'exécution
- JOBHLD : nombre de jobs suspendus
- JOBPEND : nombre de jobs en attente

Exemple

```
D:\UNIV56\exec>uxshwque queue=SYS_BATCH
```

Cette commande affiche les informations ci dessous :

QUEUE NAME	TYPE	STA	JOBLIM	JOBQUE	JOBEXE	JOBHLD
JOBPEND						

BATCH1	PHYS	OFF	5	0	0	0
0						
LOG_QUEUE	LOGI	ON	999999	0	0	0
0						
SYS_BATCH	PHYS	ON	99999	1	1	0
0						

La queue SYS_BATCH a un job en cours d'exécution.

7.8 Afficher l'état d'une queue batch

La commande **uxshwque** permet de visualiser l'état d'une queue batch, ainsi qu'une liste contenant le nom et l'état des jobs placés dans cette queue batch. La syntaxe de cette commande est la suivante :

Commande	Item	Format
uxshwque	QUEUE=	31 caractères max.

Description des items

QUEUE : Nom de la queue batch. L'item Queue peut contenir des caractères génériques. La valeur all permet de visualiser l'ensemble des queues batch.

Exemple

```
D:\UNIV56\exec>uxshwque queue=SYS_BATCH
```

Cette commande affiche les informations ci dessous :

```
Queue SYS_BATCH
  JobLim 10
  Priorite de base 1
  Priorite d'execution 10
```

```
Priorite maximale 255
JobQue 1 , JobExe 1 , JobHld 0 , JobPend 0
Jobname      Entry   Node  User      Prio SecPrio  stat      Date
Mu
XSATURNIU_TEST000  000002          univ56a  100  0000000  execution 20080930
11:41 SATURN
Completed successfully
```

Les caractéristiques de la queue batch sont de 10 jobs maximum en cours (JobLim), 1 job est présent dans la queue (JobQue) en cours d'exécution (JobExe), il n'y a pas de job suspendu (JobHld) ni attente dans la queue (JobPend).

Les jobs présents dans la queue sont listés en mentionnant le nom du job (Jobname), le numéro d'entrée dans la queue (Entry), l'utilisateur de soumission (User), la priorité d'exécution (Prio), la priorité secondaire (SecPrio), l'état du job (Stat), la date d'exécution et l'Unité de Gestion (Mu).

7.9 Soumettre un job

Sous UNIX / Linux uniquement

la commande **uxsubjob** permet de soumettre un job en queue batch. Le paramètre nœud DQM_ENABLE_SUBJOB doit être défini à Oui pour que la commande soit prise en compte.

La syntaxe de cette commande est la suivante :

Commande	Item	Format
uxsubjob	QUEUE=	31 caractères max
	CMD=	255 caractères max
	LOG=	255 caractères max
	USER=	12 caractères max
	NAME=	
	PRI=	[0-255]
	PARAM=	"parm1,parm2,...parm30"

Le code retour est 0 si le job a bien été soumis dans la queue batch.

 Si le paramètre nœud DQM_ENABLE_SUBJOB est défini à Non, la commande ne fait rien et retourne 0.

Description des items

QUEUE: Nom de la queue batch dans laquelle sera soumis le job.

CMD: Chemin et nom du fichier de commande (selon les standards du système d'exploitation) à exécuter en queue batch.

LOG: Chemin et nom du fichier log (selon les standards du système d'exploitation) d'exécution du job.

USER: Nom de l'utilisateur pour le compte duquel doit être exécuté le job. Par défaut, prend le nom de l'utilisateur qui exécute la commande uxsubjob.

NAME : Nom que prend le processus au sein de la queue batch. Par défaut, prend le nom du fichier de commande précisé dans cmd, sans le répertoire.

PRI : Priorité de soumission de base qu'aura le job dans la queue batch. Cette priorité sera ensuite gérée au sein de la queue batch en fonction des paramètres de cette queue batch et en particulier de la gestion dynamique des priorités. Par défaut, prend la priorité de base de la queue. Si la priorité est supérieure à 255, elle est ramenée à 255.

PARAM : Liste des paramètres qui doivent être transmis au fichier de commande cmd. Le nombre maximum de paramètres transmis est de 30. La longueur de chaque paramètre ne doit pas excéder 255 caractères et la longueur totale (somme des longueurs + 2 fois le nombre de paramètres) doit être inférieure à 2074.

 Cette commande ne fonctionne que sur le nœud local.

Exemple

```
$UNI_DIR_EXEC/uxsubjob QUEUE=SYS_BATCH CMD=$UXPEX/bck_jour.sh
LOG=$UXLOG/bck_jour.log
```

L'exécution de cette commande soumet le job *bck_jour.sh* dans la queue batch.

7.10 Suspendre un job

La commande **uxhldjob** permet de suspendre un job en queue batch. Le job échappe à la mécanique de gestion des priorités. Il pourra être libéré par l'interface ou par la commande **uxrlsjob**.

La syntaxe de cette commande est la suivante :

Commande	Item	Format
uxhldjob	ENTRY=	[0-999999]

Description des items

ENTRY : Numéro de séquence attribué à un job par DQM. Ce numéro est visualisé par la commande **uxshwque**.

 Cette commande ne fonctionne que sur le nœud local.

Exemple

```
$UNI_DIR_EXEC/uxhldjob ENTRY=$NUMJOB
```

La variable NUMJOB a été précédemment alimentée par le numéro d'entry correspondant au job qui doit être suspendu.

7.11 Libérer un job suspendu

La commande **uxrlsjob** permet de libérer un job retenu en queue batch. Cette commande permet d'annuler l'effet de la commande **uxhldjob**. La syntaxe de cette commande est la suivante :

Commande	Item	Format
uxrlsjob	ENTRY=	[0-999999]

Description des items

ENTRY : Numéro de séquence attribué à un job par DQM. Ce numéro est visualisé par la commande **uxshwque**.

 Cette commande ne fonctionne que sur le nœud local.

Exemple

```
$UNI_DIR_EXEC/uxrlsjob ENTRY=$NUMJOB
```

La variable NUMJOB a été précédemment alimentée par le numéro d'entry correspondant au job qui doit être relâché.

L'exécution de la commande a pour effet l'exécution du job dans la queue batch au plus tôt, en fonction de la priorité du job.

7.12 Afficher l'état d'un job

La commande **uxshwjob** permet de visualiser l'état d'un job placé en queue batch. La syntaxe de cette commande est la suivante :

Commande	Item	Format
uxshwjob	ENTRY=	[0-999999]

Description des items

ENTRY : Numéro de séquence attribué à un job par DQM. Ce numéro est visualisé par la commande **uxshwque**.

 Cette commande ne fonctionne que sur le nœud local.

Exemple

```
$UNI_DIR_EXEC/uxshwjob ENTRY=$NUMJOB
```

La variable NUMJOB a été précédemment alimentée par le numéro d'entry correspondant au job qui doit être affiché.

L'exécution de la commande a pour effet l'affichage de l'état du job dans la queue batch et de ses caractéristiques :

```
Xagencebck_dev000 root 0058 0100 exécution 960514 15:57
```

Où :

- Xagencebck_dev000 est le nom du job.
 - Root est le nom du user d'exécution.
-

- 0058 est le numéro d'entry du job.
- 0100 est la priorité avec laquelle le job est entré dans la queue batch.
- Exécution est l'état du job.
- 960514 15:57 est la date (aammjj) et l'heure de début d'exécution du job.

7.13 Arrêter un job

La commande **uxstpjob** permet d'arrêter un job placé en queue batch quel que soit l'état de ce job. La syntaxe de cette commande est la suivante :

Commande	Item	Format
uxstpjob	ENTRY=	[0-999999]

Description des items

ENTRY : Numéro de séquence attribué à un job par DQM. Ce numéro est visualisé par la commande **uxshwque**.

 Cette commande ne fonctionne que sur le nœud local.

Exemple

```
$UNI_DIR_EXEC/uxstpjob ENTRY=$NUMJOB
```

La variable NUMJOB a été précédemment alimentée par le numéro d'entry correspondant au job qui doit être arrêté.

L'exécution de la commande a pour effet l'arrêt immédiat de l'exécution du job dans la queue batch.

7.14 Modifier les priorités d'un job

La commande **uxsetpri** permet de modifier les priorités d'ordonnancement ou d'exécution d'un job placé en queue batch. Cette modification ne peut intervenir que si le job n'est pas déjà dans l'état en cours. La syntaxe de cette commande est la suivante :

Commande	Item	Format
uxsetpri	ENTRY=	[0-999999]
	PRI=	[0-999]
	EXEPRI=	[0-255]

Description des items

ENTRY : Numéro de séquence attribué au job par DQM. Ce numéro est visualisé par la commande **uxshwque**.

PRI : Nouvelle priorité de soumission dans la queue batch.

EXEPRI : Nouvelle priorité d'exécution dans la queue batch.

 Cette commande ne fonctionne que sur le nœud local.

Exemple

```
$UNI_DIR_EXEC/uxsetpri ENTRY=$NUMJOB PRI=200 EXEPRI=100
```

La variable NUMJOB a été précédemment alimentée par le numéro d'entry correspondant au job qui doit être arrêté.

L'exécution de la commande a pour effet la modification des priorités associées au job dans la queue batch.

8 Gestion de date

Ce chapitre présente les commandes utilisées pour gérer les formats et décalages de dates et heure.

 Non disponible sous OS/400

Dollar Universe utilise, pour le calcul de la planification, des algorithmes perfectionnés de gestion des dates et des heures, que ce soit sur le format de la date ou sur le calcul de celle-ci, ou pour prendre en compte les différents types de jours : calendaires, ouvrés, chômés ou fériés.

8.1 Formatage et décalage de dates

La commande **uxdat** permet de modifier le format d'une date, de réaliser un décalage en jour et d'obtenir le type ou le rang d'un jour dans le calendrier.

La plus petite unité de temps traitée est le jour et la plus grande est l'année.

Les sept résultats sont fournis à l'écran (Unix, Windows) ou sous la forme de symboles (OpenVMS). Seul, l'un des 7 résultats est renseigné selon le type demandé.

Seule la cohérence syntaxique est vérifiée.

La langue par défaut est internationale (Anglais). Le choix de la langue du format est déterminé par le format lui-même: j, a, s, ... pour Français et d, y, w, ... pour International.

Le résultat est en minuscules si le format l'est, il est en majuscules si au moins un caractère du format de sortie spécifié est saisi en majuscules.

Les accès aux calendriers de Dollar Universe ne se font que lors des décalages en jours ouvrés et des demandes de "Statut de la date source". Toutes les autres opérations sont indépendantes de l'environnement de Dollar Universe. L'item optionnel MU=<nom de l'UG> indique le calendrier ciblé.

Deux syntaxes sont possibles selon le résultat que l'on veut obtenir.

Formatages et décalages

Les décalages sont traités séquentiellement de gauche à droite.

Le format de la commande est le suivant :

Commande	Item
uxdat	Format date source Date source Format date cible Décalage

```
uxdat "jj/mm/aaaa" "01/01/1999" "jj-mm-aa" "-1jo/1m"
```

- Résultat 1 : U_DATE_RES : Date au format cible avec le décalage demandé

Type de jour

Le format de la commande est le suivant :

Commande	Item
uxdat	Format date source Date source Type

```
uxdat "jj/mm/aaaa" "01/01/1999" "type" [MU=<nom de l'UG>]
```

Le résultat ou symboles globaux (OpenVMS) dépend de la valeur prise par "type". uxdat valorise les symboles globaux suivants en fonction des arguments.

- Résultat 2 : U_TYPE_JOUR : Statut de la date source (Ouvré, Chômé, Férié).
- Résultat 3 : U_RGO_DEB_MOIS : Rang d'un jour ouvré dans le mois (à partir du début).
- Résultat 4 : U_RGO_FIN_MOIS : Rang d'un jour ouvré dans le mois (à partir de la fin).
- Résultat 5 : U_RGO_DEB_ANNEE : Rang d'un jour ouvré dans l'année (à partir du début).
- Résultat 6 : U_RGO_FIN_ANNEE : Rang d'un jour ouvré dans l'année (à partir de la fin).
- Résultat 7 : U_NBJOURS: Nombre de jours entre deux dates.

8.1.1 Description des items

Format des dates	Format date français	Format date international
Mois sous la forme 1-12	m	m
Mois sous la forme 01-12	mm	mm
Mois sous la forme de jan à déc (jan to dec)	mmm	mmm
Mois sous la forme de janvier à décembre (january to december)	mmmm	mmmm
Jours sous la forme 1-31	j	d
Jours sous la forme 01-31	jj	dd
Jours sous la forme 1 à 7	p	p
Jours sous la forme lun à dim (mon to sun)	jjj	ddd
Jours sous la forme lundi à dimanche (monday to sunday)	jjjj	dddd
Semaine sous la forme 1-52	s	w
Semaine sous la forme 01-52	ss	ww
Quantième sous la forme 1-365	q	q
Quantième sous la forme 001-365	qqq	qqq
Années sous la forme 00 à 99	aa	yy
Années sous la forme 0000 à 9999	aaaa	yyyy

Lorsque le format d'entrée comporte une année sur 2 caractères, l'année est interprétée comme suit :

- de 00 à 40, l'année est située entre 2000 et 2040,
- de 70 à 99, l'année est située entre 1970 et 1999.

 La commande uxdat n'accepte toujours que des dates comprises entre 1970 et 2040.

Décalages	Français	International
Séparateur	/	/

Décalages	Français	International
Position	n	n
Décalage négatif	-n	-n
Décalage positif	+n	+n
Décalage positif de n jour calendaire	+nj	+nd
Décalage négatif de n jour calendaire	-nj	-nd
Décalage positif de n jour ouvré	+njo	+ndo
Décalage négatif de n jour ouvré	-njo	-ndo
Position au nième jour calendaire de la semaine	ns	nw
Position au nième jour ouvré de la semaine	nso	nwo
Décalage positif de n semaine	+ns	+nw
Décalage négatif de n semaine	-ns	-nw
Position au nième jour calendaire du mois	nm	nm
Position au nième jour ouvré du mois	nmo	nmo
Position au nième jour (type p) calendaire du mois	nmp	nmp
Position au nième jour (type p) ouvré du mois	nmop	nmop
Décalage positif de n mois	+nm	+nm
Décalage négatif de n mois	-nm	-nm
Position au nième jour calendaire de l'année	na	ny
Position au nième jour ouvré de l'année	nao	nyo
Position au nième jour (type p) de l'année	nap	nyp
Position au nième jour (type p) ouvré de l'année	naop	nyop
Position au nième mois de l'année	nma	nmy
Décalage positif de n année	+na	+ny
Décalage négatif de n année	-na	-ny

"n" est un entier

Type	Type Français	Type International
Rang d'un jour ouvré dans le mois (début)	odm	obm
Rang d'un jour ouvré dans le mois (fin)	ofm	oem
Rang d'un jour ouvré dans l'année (début)	oda	oby
Rang d'un jour ouvré dans l'année (fin)	ofa	oey

8.1.2 Exemples

Exemples de format pour le 01/02/1999

"j"	1
"jj"	01
"jjj"	lun
"jjjj"	lundi
"jjjj jj mmmm aaaa"	lundi 01 février 1999
"dd mmmm yyyy"	01 february 1999
"q"	32 (quantième : position dans l'année)
"qqq"	032 (quantième : position dans l'année)
"s"	5 (semaine)
"ss"	05 (semaine)
"jj-mmm-aaaa"	01-fev-1999
"dd-mmm-yyyy"	01-feb-1999

Exemples de décalage pour le 01/02/1999

" +1m/1m/- 1jo"	dernier jour ouvré du mois (décalage +1 mois, 1 ^{er} jour du mois, décalage -1 jour ouvré)
" +1a/1a/- 1s/7s"	dernier dimanche de l'année (décalage +1 an, 1 ^{er} jour de l'année, décalage -1 semaine, 7 ^{ème} jour)

8.2 Formatage de date et décalage d'heure

La commande **uxtim** permet de convertir une date connue ou non dans un format donné et/ou d'y ajouter un décalage horaire.

Pour une date connue, le format de la commande est le suivant :

Commande	Item			
uxtim	"Format date heure entrée"	"Date heure entrée"	"Format date heure sortie"	"Décalage horaire"

Par exemple :

```
uxtim "hhMM ap" "0600 PM" "HH:MM" "-0h10"
```

Pour une date correspondant à la date et heure d'exécution de la commande, le format est le suivant :

Commande	Item		
uxtim	NOW	"Format date heure sortie"	"Décalage horaire"

Par exemple :

```
uxtim NOW "aaaa, mmmm, jj HH:MM AP" "+2h50"
```

Sous OpenVMS, uxtim valorise le symbole global UXTIM_VALUE.

8.2.1 Descriptions des items

Format date heure entrée – Format date heure sortie : respectivement le format de la date à convertir et le format souhaité.

Le format de l'heure utilise les symboles HH pour les heures, MM pour les minutes et AP pour indiquer le format horaire (en format 12h).

Par exemple : "15012003, 1506" correspond a "jjmmaaaa, HHMM"

	Format date français	Format date international
Mois sous la forme 01-12	mm	mm
Mois sous la forme de jan à déc (jan to dec)	mmm	mmm
Mois sous la forme de janvier à décembre (january to december)	mmmm	mmmm
Jours sous la forme 01-31	jj	dd
Jours sous la forme 1 à 7	p	p
Jours sous la forme lun à dim (mon to sun)	jjj	ddd
Jours sous la forme lundi à dimanche (monday to sunday)	jjjj	dddd
Semaine sous la forme 01-52	ss	ww
Quantième sous la forme 001-365	qqq	qqq
Années sous la forme 00 à 99	aa	yy
Années sous la forme 0000 à 9999	aaaa	yyyy

Date Heure entrée : date et heure à convertir selon le format indiqué en Format date heure sortie. Cette date – heure doit être saisie selon le Format date heure entrée indiqué.

Décalage horaire : décalage à appliquer à la date – heure saisie en entrée ou à la date – heure courante si le mot clé NOW est indiqué. Le format du décalage est : +/-HHHhMM :

- le signe + indique un décalage dans le futur, le signe – indique un décalage dans le passé.
- HHH est le nombre d'heure à décaler (sur 3 digits maximum),
- "h" est un délimiteur qui doit être saisi tel quel,
- MM est le nombre de minutes à décaler (entre 0 et 59).

8.2.2 Exemples

```
% uxtim "hhMM ap" "0600 PM" "HH:MM"
18:00
```

"0600 PM" a été converti selon le format indiqué (hhMM ap), il n'y a pas de décalage demandé.

```
% uxtim "hhMM ap" "0600 PM"
0600 PM
```

"0600 PM" n'a pas été converti, il n'y a pas de format de sortie indiqué et pas de décalage.

```
% uxtim NOW "aaaa, mmmm, jj HH:MM AP"
2003, mars, 05 03:30 PM
```

La date et heure courante a été convertie au format indiqué, pas de décalage.

```
% uxtim NOW "aaaa, mmmm, jj HH:MM AP" "+2h50"  
2003, mars, 05 06:20 PM
```

La date et heure courante a été convertie au format indiqué, un décalage de 2h50 a été appliqué.

```
% uxtim "aaaa, mmmm, jj HH:MM AP" "2003, JANVIER, 21 06:26 PM" "jj/mm/aa"  
21/01/03
```

Le "21 janvier 2003 6h26 PM" a été converti au format indiqué, pas de décalage.

9 CL d'exécution

Ce chapitre présente les commandes utilisées dans le CL des Uprocs.

Les commandes décrites ci-dessous inscrivent une trace de leur exécution dans l'historique des exécutions (trace automate). Si l'historisation est désactivée dans la tâche ou si le paramètre noeud U_HISTORIC est positionné à "Non" (catégorie Réglages avancés), la commande s'exécutera normalement. La trace d'exécution sera visible dans la trace automate pendant la durée d'exécution de l'Uproc.

9.1 Présentation du CL

Le CL d'exécution désigne le fichier de commandes associé à l'Uproc (type CL_INT ou CL_EXT). Dans la suite de ce chapitre, toutes les procédures d'exploitation écrites en langage de commande (propre au système d'exploitation) seront désignées sous l'unique terme de "procédures".

Le CL permet d'accéder à la communication indispensable entre les procédures et Dollar Universe. Celui-ci fournit donc un ensemble de commandes qui permettent aux procédures d'acquérir les informations susceptibles de les intéresser ou de lui en transmettre.

Rappelons que les pré-traitements effectués par Dollar Universe avant le lancement effectif de l'exécution (pilottage) ne nécessitent aucune insertion de commande particulière dans le CL.

D'autre part, il est conseillé, pour des raisons de facilité, de manipulation et de maintenance de réaliser des procédures homogènes au plan fonctionnel et courtes (formules de lancement simplifiées, expression de conditions plus simples).

Les commandes proposées par Dollar Universe permettent d'obtenir des procédures parfaitement portables quels que soient le site ou la configuration utilisée, d'assurer une gestion cohérente de l'exploitation et d'exploiter dans les chaînes applicatives toute la logique induite par les concepts gérés par Dollar Universe (unité de gestion et relations entre elles, espaces, ...).

9.1.1 Normes d'écritures du CL

L'écriture du CL peut se faire soit sous le contrôle de Dollar Universe soit indépendamment avec l'éditeur de son choix.

Le CL peut utiliser, en dehors ou avec les commandes décrites ci-après, les variables logiques ou les variables d'environnement utilisées par Dollar Universe.

9.1.2 Le code retour du CL

Si Dollar Universe peut connaître la qualité technique de la terminaison d'une procédure, celle-ci peut cependant être, d'un point de vue applicatif, jugée incorrectement terminée.

Aussi et pour couvrir tous les cas possibles, Dollar Universe propose aux procédures de lui restituer un indicateur positionnant le code retour souhaité.

Deux procédures peuvent rendre un code retour : le pré traitement et le CL de l'Uproc.

Le pré traitement restitue son code retour dans la variable `unix_status`, l'exit de l'Uproc est restitué dans la variable `RESEX` ou `S_RESEX` sous OpenVMS. Ils peuvent être utilisés dans le CL de l'Uproc ainsi que dans le post traitement de la coquille batch.

Sous OpenVMS uniquement, la variable `S_RESEX` peut également être utilisée dans le CL pour définir si le travail s'est correctement terminé ou pas indépendamment de son code retour :

- `S_RESEX=="V"` (le job se terminera correctement indépendamment de son code retour)
- `S_RESEX=="I"` (le job s'incidentera indépendamment de son code retour)

Si la variable `S_RESEX` est définie avec une autre valeur, le code retour est pris en compte.

 Sur OpenVMS, si la variable `S_RESEX` vaut "V" ou "I" et aucun code de sortie n'est spécifié, les règles de gestion des statuts d'Uproc basées sur les valeurs des codes de retour ne sont pas disponibles.

9.1.3 Résumé des commandes

Les commandes spécifiques disponibles sont les suivantes :

Commande	Fonction
<code>uxsetabort</code>	Interdit l'arrêt de l'exécution du job par une interface d'exploitation de Dollar Universe
<code>uxset parm</code>	Passage de paramètres dans le cadre d'une session
<code>uxset info</code>	Affichage d'informations supplémentaires dans le suivi des exécutions
<code>uxset msg</code>	Ecriture de messages dans l'historique d'une Uproc
<code>uxset step</code>	Ecriture d'un jalon dans une Uproc
<code>uxset var</code>	Reconduction de la valeur d'une variable au sein d'une session
<code>uxset hvar</code>	Transmet les variables d'une Uproc à la suivante dans une session
<code>uxhideval</code> & <code>uxexechidden</code>	Commandes qui peuvent être mises en œuvre pour gérer les variables cachées Exécuter une commande en utilisant la valeur d'une ou de plusieurs variable cachées

Dans chacune de ces commandes la valeur prise par défaut des paramètres concernant les unités de gestion et les applications sont, systématiquement celles de l'environnement d'exécution.

9.2 Interdire l'arrêt de l'exécution d'une Uproc

L'objet de la commande **`uxsetabort`** est de définir une zone dans le script de l'Uproc qui ne peut pas être interrompue par une commande d'arrêt provenant d'une interface d'exploitation de Dollar Universe :

- UniViewer Console : liste des exécutions.
- Commande `uxend ctl`.

 Les commandes du système d'exploitation peuvent toujours être utilisées pour arrêter l'exécution mais ce n'est pas recommandé.

Cette commande ne peut être exécutée que dans le script d'une Uproc. Son exécution est tracée dans le log d'exécution de l'Uproc et dans la trace automate.

Si l'opérateur demande l'arrêt de l'exécution de l'Uproc pendant la zone protégée, un message l'informe du comportement paramétré :

- Soit la commande d'arrêt est exécutée à la fin de la zone protégée.
- Soit la commande d'arrêt n'est pas exécutée. Elle est annulée.

Si l'opérateur demande l'arrêt de l'exécution de l'Uproc en dehors de la zone protégée, elle est exécutée immédiatement, quel que soit les paramètres de configuration de la commande.

```
uxsetabort allow=on|off [queue=on|off]
```

Description des items

allow: délimite la zone protégée du script de l'Uproc. Obligatoire.

- allow=off marque le début de la zone protégée, pendant laquelle l'Uproc ne peut pas être arrêtée par une commande d'une interface de Dollar Universe.
- allow=on marque la fin de la zone protégée. Si cette commande est absente du script, la zone protégée s'étend jusqu'à la fin du script.

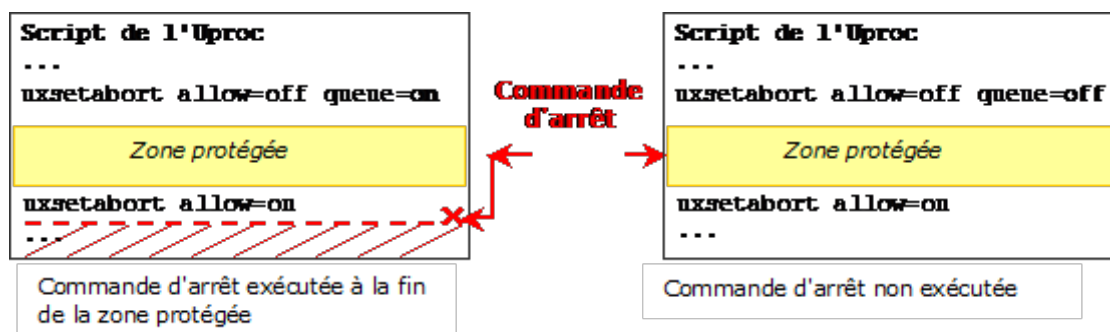
queue: indique si une commande d'arrêt demandée par un utilisateur doit être ou non mémorisée.

Optionnel. Ne peut être utilisé qu'avec allow=off.

- queue=on indique que la commande d'arrêt doit être mémorisée. Elle sera donc exécutée dès la fin de la zone protégée. Valeur par défaut.
- queue=off indique que la commande d'arrêt ne doit pas être mémorisée. Elle ne sera donc pas exécutée à la fin de la zone protégée.

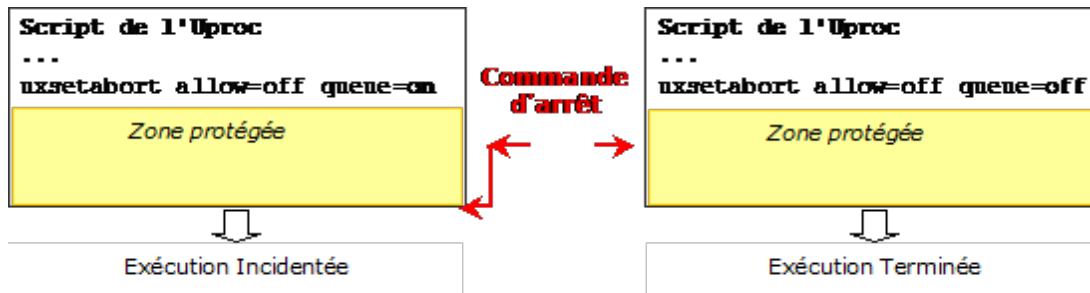
Exemples

Si la zone protégée est délimitée par une commande de début (uxsetabort allow=off) et une commande de fin (uxsetabort allow=on), le comportement est le suivant, selon la valeur de l'item queue :



Si queue=on, la commande d'arrêt est exécutée après la fin de la zone protégée.

Si la zone protégée s'étend jusqu'à la fin du script, le comportement est le suivant, selon la valeur de l'item queue :



Si `queue=on`, la commande d'arrêt est mémorisée et est exécutée à la fin du script de l'Uproc. L'état de l'exécution de l'Uproc est alors Incidenté. Si l'Uproc faisait partie d'une session, le chemin erreur de la session (s'il existe) sera exécuté.

9.3 Ecrire des jalons

La commande **uxset step** (UXSETSTEP sous OS/400) permet de définir des jalons (ou steps) au sein du CL d'exécution. Ces jalons peuvent également permettre de reprendre des exécutions après un incident sans ré-exécuter systématiquement l'ensemble de l'Uproc.

Les numéros de jalons sont enregistrés dans la base des événements au fur et à mesure que l'exécution de l'Uproc se déroule. Les jalons permettent donc, dans le cadre du suivi dynamique d'exploitation, de suivre la progression d'exécution d'une Uproc.

Dans un cas de reprise, Dollar Universe propose de saisir le numéro de jalon à partir duquel l'utilisateur désire voir démarrer son exécution de reprise (01 par défaut). La valeur du numéro de jalon saisie est disponible dans le CL de l'Uproc au travers de la variable `S_NUMJALON`. Cette variable doit être testée dans le CL, et suivant sa valeur, le CL doit dérouter le programme à la ligne désirée.

```
uxset step nn
```

Description des items

nn représente un numéro de jalon sur deux caractères numériques. Le passage au jalon sera horodaté dans la trace Dollar Universe d'exécution de l'Uproc.

Exemple

```
uxset step 2
```

Cette commande inscrite dans le CL de l'Uproc, aura pour effet de marquer cette ligne comme étant le jalon n°2. Lors de l'exécution de la procédure, le suivi de l'exploitation sous Dollar Universe permettra de visualiser le passage à ce jalon (et aux autres).

Dans l'annexe de ce manuel, des procédures d'exemples en ksh et csh illustrent l'utilisation des jalons.

9.4 Afficher des informations

La commande **uxset info** (UXSETINFO sous OS/400), exécutée dans le script d'une Uproc, permet de valoriser des informations utilisateur dans la liste des exécutions :

- Un champ texte optionnel : "Information".
- Un nombre entier optionnel : "Sévérité".

Les valeurs correspondantes sont affichées dans le suivi des exécutions si les valeurs sont différentes des valeurs précédentes (valeurs par défaut de l'Uproc ou commande précédente exécutée dans le script) :

- UniViewer Console : les valeurs sont affichées dans les colonnes "Informations" et "Sévérité" de la liste des exécutions,
- Interface commandes : pour afficher les valeurs, utiliser le mot clé FULL INFO de la commande uxlst ctl.

```
uxset info [value=VALUE] [severity=NUM]
```

Description des items

value= saisir le texte qui doit être affiché, encadré de guillemets s'il contient des caractères 'espace'. Longueur maximum de 50 caractères. Si la sévérité n'est pas spécifiée, la valeur par défaut est 0.

severity= est un nombre entier de 0 à 9999. La valeur par défaut est 0. Si la sévérité n'est pas spécifiée, la valeur par défaut est 0.

Exemple

L'exécution d'une Uproc dont le script contient cette commande :

```
uxset info value="mon job est en cours d'execution" severity=0004
```

Affiche "mon job est en cours d'execution" dans la colonne Informations et "0004" dans la colonne Sévérité de la liste des exécutions d'UVC.

9.5 Générer des messages

La commande **uxset msg** (UXSETMSG sous OS/400) permet de communiquer entre le CL d'une Uproc et la trace automate de son exécution. Elle permet ainsi, par exemple, d'insérer des consignes d'exploitation dans cette trace en fonction du déroulement de l'exécution de la procédure.

La trace automate d'exécution de l'Uproc est disponible dans l'interface de Dollar Universe "suivi d'exploitation" ou "historique d'exécutions" en sélectionnant cette Uproc et en demandant l'option "Trace Automate".

 Tous les messages générés par cette commande sont contenus dans les fichiers de l'historique des exécutions, celui-ci peut donc (en fonction du volume traité) prendre une taille importante.

```
uxset msg "var1" "var2" ... "var8"
```

Description des items

var désigne de 1 à 8 variables ou lignes de texte. Si var représente une variable, sa valeur sera traduite dans la trace, si var représente du texte elle sera transcrite à l'identique dans la trace Dollar Universe de l'exécution.

Chaque commande uxset msg affiche un message dans la trace automate. La longueur maximum d'un message est de 1600 caractères; le message pourra être découpé en lignes de 80 caractères.

Exemple

```
uxset msg " *** UG d'execution : " "${S_CODUG}"
```

Sous UNIX, cette commande aura pour effet l'inscription du message suivant dans la trace d'exécution de l'Uproc :

```
*** UG d'execution : A_PARIS
```

si A_PARIS est l'unité de gestion d'exécution de l'Uproc.

9.6 Passer des paramètres

La commande **uxset parm** (UXSETPARM sous OS/400) permet le passage de paramètres d'exécution entre des Uprocs dans le cadre d'une Session. Les paramètres positionnés par une Uproc pourront être utilisés par ses Uprocs enfants sous la forme standard: sous les numéros 1 à 30 ou sous la forme Dollar Universe dans les variables : S_P1 à S_P30.

Les paramètres passés par cette commande sont utilisables :

- sous la forme standard : dans le script des Uprocs uniquement,
- sous la forme Dollar Universe : dans le script des Uprocs ou dans les scripts lancés par la coquille batch : pré traitement et post traitement.

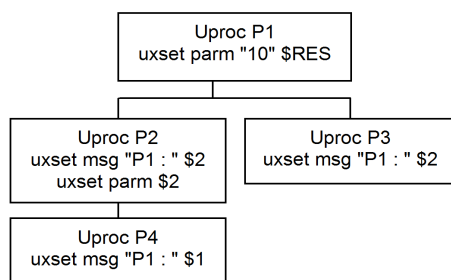
```
uxset parm "parm1" "parm2" ... "parm30"
```

Description des items

parmN désigne de 1 à 30 paramètres. Le nombre de paramètres est de 30 au maximum. D'une longueur maximale de 255 caractères chacun, la longueur totale (somme des longueurs + 2 fois le nombre de paramètres) doit être inférieure à 2074 caractères.

Exemple

Sur la Session suivante :



Si la commande `uxset parm "10" $RES` est exécutée par l'Uproc P1, l'Uproc P2 pourra utiliser le second paramètre par `S_P2`, l'Uproc P3 pourra utiliser les deux paramètres par `S_P1` et `S_P2`.

Pour que P2 puisse transmettre les paramètres à P4, il faut exécuter dans P2 une nouvelle commande `uxset parm`.

Récupérer des paramètres sous OS/400

Les paramètres ne sont pas récupérés implicitement dans les programmes OS/400 classiques. La commande UXGETPARM permet de récupérer la valeur d'un paramètre passé par une commande UXSETPARM ou UXORDRE dans une variable DCL.

```
CALL PGM(UXGETPARM) PARM( ' PARM=n ' ' DCLVAR=var ' )
```

Cette commande doit être utilisée dans un CL. Elle accepte deux paramètres :

- **PARM** : numéro du paramètre positionnel à récupérer : de 0 à 11
- **DCLVAR** : chaîne de longueur maximum 256 caractères. Variable CL dans laquelle la valeur du paramètre sera écrite.

 En qsh, le paramètre n est directement utilisable dans la variable \$n du shell.

9.7 Reconduire des variables

La reconduction de la valeur des variables au sein d'une session n'est pas automatique. Par défaut, si rien n'est fait, la valeur de la variable n'est pas reconduite et l'automate applique les règles de valorisation par défaut.

La reconduction de la valeur peut être demandée en exécutant la commande **uxset var** (UXSETVAR sous OS/400) dans le C.L. de l'Uproc ou dans le post-traitement (U_POST_UPROC).

La commande **uxset var** permet de reconduire la valeur des variables au sein de la session. Trois types de reconduction peuvent être demandées :

- en citant dans la commande uxset var les variables qui doivent être reconduites et éventuellement leur nouvelle valeur.
- de façon globale, en exécutant la commande uxset var sans autre complément ou bien uxset var level ALL, pour ne reconduire que les valeurs des variables de l'Uproc parent connues de l'Uproc enfant (les valeurs reconduites sont les valeurs connues au début de l'exécution de l'Uproc parent).
- en exécutant la commande uxset var level INTER, pour ne reconduire que les valeurs des variables reçues par l'Uproc parent et connues de l'Uproc enfant. Cette commande ne peut donc pas être exécutée dans une Uproc d'entête.

La reconduction des variables au sein d'une session est assurée par l'exécution de la commande. Si l'Uproc suivante dans la session n'est pas exécutée par l'automate (par exemple parce qu'elle est définie comme une tâche optionnelle), Dollar Universe transmet alors les variables à l'Uproc immédiatement suivante.

 Sous UNIX/Linux, le nom des variables ne doit pas commencer par un chiffre (restriction du système d'exploitation).

Les syntaxes présentées ci-dessous sont autorisées :

```
uxset var
uxset var level ALL
uxset var level INTER
uxset var VAR1=VAL1 VAR2 VAR3=VAL3
```

Description des items

VARn désigne le nom de la variable et **VALn** la valeur associée qui sera transmise à l'Uproc enfant.

- Si la valeur n'est pas indiquée (pas de signe égal) la dernière valeur de la variable, ou sa valeur par défaut, est transmise.
- Si une valeur contient des caractères non alphanumériques elle doit être encadrée par des guillemets, par exemple :

```
uxset var VARDATE=2010/11/30 VARTXT="TEST TEST" VARNUM=555
```

Les guillemets encadrant la chaîne "TEST TEST" sont requis pour conserver l'espace dans la chaîne "TEST TEST" ; si la chaîne ne contient pas d'espace ils sont inutiles.

Exemples

Exemple 1

Trois Uprocs A, B, C ayant les mêmes variables par défaut : Var1, Var2, Var3.

L'Uproc A a un script contenant la commande : uxset var level ALL

L'Uproc B a un script contenant la commande : uxset var level INTER

L'Uproc C a un script qui affiche la valeur des variables.

Dans ce cas, le script affiche les valeurs de Var1, Var2 et Var3 de l'Uproc A, qui ont été transmises à l'Uproc B par la commande "uxset var level ALL", puis passées à l'Uproc C par la commande "uxset var level INTER".

La commande "uxset var level INTER" ne reporte que les informations transmises par une Uproc parent. Le report peut être cascadié sur plusieurs Uprocs par la commande "uxset var level INTER".

Exemple 2

Trois Uprocs X, Y, Z.

Variables de l'Uproc X: Var1=Val1x, Var2=Val2x, Var3=Val3x

Variables de l'Uproc Y: Var1=Val1y, Var2=Val2y, Var3=Val3y

Variables de l'Uproc Z: Var1=Val1z, Var2=Val2z, Var3=Val3z

Toutes ont les mêmes valeurs par défaut: Var1=Val1, Var2=Val2, Var3=Val3

L'Uproc X a un script contenant la commande : uxset var Var1=NewValue

L'Uproc Y a un script contenant la commande : uxset var level INTER

L'Uproc Z a un script qui affiche la valeur des variables.

Dans ce cas, le script affiche les valeurs NewValue, Val2z et Val3z. Dans ce cas, l'Uproc X a transmis NewValue à l'Uproc Y, et NewValue a été reportée dans l'Uproc Z par la commande "uxset var level INTER".

Récupérer des variables sous OS/400

Les variables ne sont pas récupérées implicitement dans les programmes OS/400 classiques. La commande UXGETVAR permet de récupérer la valeur d'une variable passée par une commande UXSETVAR ou UXORDRE dans une variable DCL.

```
UXGETVAR VAR( 'UPRWRD' ) DCLVAR(&VARVALUE)
```

Cette commande doit être utilisée dans un CL. Elle accepte deux paramètres :

- **VAR** : nom de la variable à récupérer
- **DCLVAR** : chaîne de longueur 256 caractères maximum. Variable CL dans laquelle la valeur de la variable sera écrite.

 En qsh, la variable var est directement utilisable dans la variable \$var du shell.

9.8 Variables d'héritage

Les variables d'héritage sont des variables qui peuvent être transmises d'une Uproc parent (dans une Session) aux Uprocs enfants de la session. A la différence des variables d'Uproc, les variables d'héritage n'ont pas de taille limite et elles n'ont pas besoin d'être déclarées dans les paramètres de l'Uproc. Lorsqu'une variable d'héritage est saisie dans le script d'une Uproc, elle est automatiquement transmise aux Uprocs enfants lors de l'exécution de la Session.

- La commande **uxset hvar** est utilisée pour créer une variable d'héritage, assigner une valeur à cette variable et définir que cette variable sera transmise aux Uprocs enfants.
- La commande **uxset inheritance [keep|discard]** spécifie si les variables doivent être transmises aux Uprocs enfants ou non.
- La commande **uxunset hvar** spécifie quelles variables ne seront pas transmises aux Uprocs enfants.

Pour les Uprocs web services de type REST et SOAP, les variables d'héritage sont créées via UVC en tant que variables de sortie. Pour plus d'informations reportez-vous au Manuel Utilisateur de Dollar Universe Manager pour Web Services.

Tous les types d'Uprocs ne gèrent pas les variables d'héritage. Le tableau ci-dessous indique les types d'Uprocs pour lesquels les variables d'héritage sont ou ne sont pas supportées.

Variables d'héritage	Supportées	Non supportées
Type d'Uproc	WS_SOAP	CMD
	WS_REST	EJB
	ODB_JOB	FTP_GET
	ODB_TPL	FTP_PUT
	CL_INT	JMS_SEND
	CL_EXT	OAPP_PGM
	MSSQL	OAPP_SET
	MSSQL_REF	SAP_IPACK
		SAP_PCHAIN
		SAP_XBP2
		ZOS

Les syntaxes présentées ci-dessous sont autorisées :

```
uxset hvar VAR1=VAL1
uxset inheritance [keep]
uxset inheritance [discard]
uxunset hvar VARn
```

Description des items

VARn indique le nom de la variable et **VALn** la valeur qui sera transmise à l'Uproc enfant.

Dans une commande `uxset hvar`, la valeur d'une variable peut être assignée à l'aide d'un fichier texte :

```
text_file.txt | uxset hvar VAR1 VAR2=value VAR3
```

VAR2 aura pour valeur "value". Les valeurs de VAR1 et VAR3 sont extraites du fichier `text_file.txt`. Il n'y a pas de format spécifique pour entrer les valeurs dans un fichier texte.

Exemple du contenu d'un fichier texte:

```
Paris, France
2013/09/17
Val1, Val2, Val3
```

Si plusieurs variables doivent être transmises, elles doivent être séparées par un caractère espace :

```
uxset hvar VARDATE=2013/08/16 "VARTXT=Valeur Texte" VARNUM=555
```

Les guillemets " " entourant la variable "VARTXT" sont nécessaires pour prendre en compte le caractère espace de la chaîne "Valeur Texte". Si la valeur ne contient pas de caractère espace, les guillemets ne sont pas nécessaires.

9.8.1 Contraintes sur la valeur de la variable

Lors de la transmission à l'Uproc suivante, les variables d'héritage sont créées en tant que variables d'environnement dans le système (par l'`u_batch` de l'Uproc suivante), ainsi la valeur d'une variable d'héritage doit contenir les mêmes caractères que ceux autorisés pour une variable d'environnement de l'OS visé (la commande `set` est utilisée pour définir une variable d'environnement sous Windows, la commande `export` sous UNIX). Sinon, la variable d'héritage ne pourra pas être créée en tant que variable d'environnement par le système et l'Uproc sera incidentée.

Exemple 1 : les variables d'héritage suivantes définies dans le script d'une Uproc sont correctes :

```
uxset hvar "MY_VAR1=email<who@hello.com>"
ou
uxset hvar MY_VAR1="email<who@hello.com>"
```

Ces variables seront créées en tant que variables d'environnement sur le système par DUAS. Ceci correspond aux commandes système (correctes) ci-dessous :

Dans une fenêtre de commandes Windows :

```
C:\>set MY_VAR1="email<who@hello.com>"
```

Dans un shell UNIX :

```
root@myunixmachine:/# export MY_VAR1="email<who@hello.com>"
```

Exemple 2 : la variable d'héritage suivante définie dans le script d'une Uproc n'est pas correcte :

```
uxset hvar MY_VAR1=email<toto@hello.com>
```

Cette variable ne pourra pas être créée en tant que variable d'environnement par DUAS. Ceci correspond aux commandes système (incorrectes) ci-dessous :

Dans une fenêtre de commandes Windows:

```
C:\>set MY_VAR1=email<who@hello.com>
The syntax of the command is incorrect
```

Dans un shell UNIX :

```
root@myunixmachine:/# export MY_VAR1=email<who@hello.com>
-bash: syntax error near unexpected token `newline'
```

 Si la valeur contient un caractère non alphanumérique, il doit être précédé du caractère "\".

9.8.2 Affichage des variables d'héritage

A l'exception des Uprocs de type WS_SOAP et WS_REST, les variables d'héritage ne sont pas affichées dans UVC. Pour afficher les variables transmises, ouvrez le fichier journal de l'Uproc enfant.

Exemple:

```
$! **  INHERITANCE VARIABLES
      var1: val12345
_!=====
```

Si l'option pour conserver les variables est activée (par défaut ou commande `uxset inheritance keep`), les variables d'héritage seront transmises. Si l'option pour conserver les variables est désactivée (commande `uxset inheritance discard`), les variables ne seront pas transmises à l'Uproc enfant.

Si un utilisateur a référencé une variable qui existe déjà (conflit sur le nom), la valeur de la variable définie dans la catégorie ayant la priorité la plus élevée sera utilisée. Les catégories sont les suivantes :

1. Variables créées dans le script
2. Variables standard (S_NODENAME, S_AREA, S_NUMLANC)
3. Variables d'Uproc
4. Paramètres
5. Variables d'UG
6. Variables de nœud
7. Variables définies dans U_ANTE_UPROC
8. Variables définies dans unienv

9.8.3 Exemples

Une Session contient trois Uprocs A, B et C. L'uproc A est parent des Uprocs B et C.

Exécutez la commande suivante dans le script de l'Uproc A pour générer les variables d'héritage :

```
uxset hvar Var1=Val1A Var2=Val2A Var3=Val3A
```

Les variables de l'Uproc A sont : Var1=Val1A, Var2=Val2A, Var3=Val3A

Dans cette configuration, par défaut, Var1, Var2, Var3 et leurs valeurs sont transmises aux Uprocs enfants de la Session.

Exemple 1

Si la commande "`uxset inheritance [discard]`" est exécutée dans le script de l'Uproc A, aucune variable ne sera transmise à l'Uproc B.

Si la commande "`uxset inheritance [keep]`" est exécutée, l'Uproc B recevra toutes les variables transmises.

Exemple 2

Si la commande “uxunset hvar Var1 Var3” est exécutée dans le script de l'Uproc A, l'Uproc B ne recevra que la variable Var2 ayant pour valeur “Val2A”.

9.9 Variables cachées

Ces commandes permettent de définir des variables dont la valeur est cachée à l'utilisateur (interface, logs...).

Ces variables sont indépendantes des variables d'Uproc. Elles peuvent être utilisées n'importe où dans Dollar Universe.

Les variables et leur valeur sont stockées dans un fichier spécifique. Le nom d'une variable apparaît en ASCII, sa valeur est codée.

 L'environnement logique de la Société (unienv) doit être exécuté avant l'exécution d'une des commandes ci-dessous.

Créer une variable et sa valeur dans le fichier de définition des variables spécifié:

```
uxhideval [<ESP>] INSERT [file=<chemin_nom_fichier>] var=<nom_variable> val=<valeur_variable>
```

Lister les variables du fichier:

```
uxhideval [<ESP>] LIST [file=<chemin_nom_fichier>] var=<nom_variable>
```

Supprimer une variable et sa valeur du fichier des variables:

```
uxhideval [<ESP>] DELETE [file=<chemin_nom_fichier>] var=<nom_variable>
```

Récupérer la valeur de la variable dans la sortie standard:

```
uxhideval [<ESP>] EXTRACT [file=<chemin_nom_fichier>] var=<nom_variable>
```

Exécuter une commande en utilisant la valeur d'une ou de plusieurs variable cachées (la commande affiche le résultat de son exécution ainsi que la sortie standard de la commande passée en paramètre) :

```
uxexechidden [file=<chemin_nom_fichier>] <commande> !HIDDEN_<nom_variable>! !HIDDEN_<nom_variable>!
```

 Cette commande doit être exécutée dans un script d'Uproc soumis par Dollar Universe

Description des items

<ESP> : optionnel, espace pour laquelle la commande est définie. Par défaut la valeur de la variable d'environnement S_ESPEXE sera utilisée et prendra pour valeur l'Espace dans lequel le job est exécuté. Saisir :

- EXP pour l'espace d'exploitation,
- SIM pour l'espace Simulation,
- INT pour l'espace Intégration,
- APP pour l'espace Application.

<chemin_nom_fichier> : saisir le chemin et le nom du fichier qui va contenir la définition des variables.

Cet item est optionnel, par défaut le fichier `hidevalues.dta` situé dans le répertoire racine de la Société (`UNI_DIR_DATA`) sera utilisé. Maximum 256 caractères pour le chemin et maximum 256 caractères pour le nom du fichier. Le chemin et le nom du fichier ne doivent contenir ni espace ni tabulation.

Si cet item est utilisé dans la commande `uxexechidden`, il doit être placé en premier (voir exemple ci-après).

Lors de la première utilisation de la commande `uxhideval ... INSERT`, le fichier est créé avec les droits de l'utilisateur qui exécute la commande.

Si une commande ne peut pas accéder au fichier, elle se termine en erreur.

<nom_variable> : saisir le nom de la variable. Maximum 256 caractères. Le nom de la variable ne doit contenir si espace ni tabulation. Le caractère générique `*` peut être utilisé pour la commande `uxhideval ... LIST`.

<valeur_variable> : saisir la valeur de la variable. Maximum 256 caractères. Cette valeur sera cachée à l'utilisateur. La valeur de la variable ne doit contenir si espace ni tabulation.

<commande> : commande à exécuter. La commande utilise en paramètre la valeur de la variable cachée citée (!HIDDEN_<nom variable>!) située dans le fichier de définition des variables. Une même commande peut accepter plusieurs variables cachées en paramètres.

Exemples

Crée la variable `COUNTRY` avec la valeur cachée `New_Zealand` dans le fichier `$UXMGR/HiddenValues` pour l'espace d'exploitation.

```
$UNI_DIR_EXEC/uxhideval EXP INSERT var=COUNTRY val=New_Zealand
file=$UXMGR/HiddenValues
```

Crée la variable `DURATION` avec la valeur cachée `120` dans le fichier `$UXDIR_ROOT/hidevalues.dta` pour l'espace d'exploitation.

```
$UNI_DIR_EXEC/uxhideval EXP INSERT var=DURATION val=120
```

Liste toutes les variables (sans leur valeur) du fichier `$UNI_DIR_DATA/hidevalues.dta` pour l'espace d'exploitation.

```
$UNI_DIR_EXEC/uxhideval EXP LIST var=*
```

Affiche dans la sortie standard la valeur de la variable `DURATION` si celle-ci existe dans le fichier `$UXDIR_ROOT/hidevalues.dta` pour l'espace d'exploitation.

```
$UNI_DIR_EXEC/uxhideval EXP EXTRACT var=DURATION
```

Exécute la commande `"sleep"` avec comme paramètre la valeur de la variable cachée `DURATION` contenue dans le fichier par défaut pour l'espace d'exploitation.

```
$UNI_DIR_EXEC/uxexechidden sleep !HIDDEN_DURATION!
```

Exécute la commande `"/tmp/script.sh"` avec comme paramètre la valeur de la variable `COUNTRY` contenue dans le fichier `$UXMGR/HiddenValues`.

```
$UNI_DIR_EXEC/uxexechidden file=$UXMGR/HiddenValues /tmp/script.sh !HIDDEN_
COUNTRY!
```

Supprime la variable `COUNTRY` du fichier `$UXMGR/HiddenValues` si celle-ci existe pour l'espace d'exploitation.

```
$UNI_DIR_EXEC/uxhideval EXP DELETE var=COUNTRY file="$UXMGR/HiddenValues"
```

Affiche dans la sortie du job les deux valeurs des variables cachées, définies dans le fichier par défaut :

```
$UNI_DIR_EXEC/uxexechidden echo !HIDDEN_COUNTRY! - !HIDDEN_DURATION!  
New_Zealand - 120
```

10 Annexes

Ce chapitre présente des exemples de commandes et programmes.

10.1 Exemple d'utilisation des commandes du CL

Deux exemples sont fournis ci-dessous qui utilisent le ksh et le csh.

10.1.1 Les jalons de reprise et les messages dans un script UNIX ksh

```
#!/bin/ksh
STEP0 ()
# Variable S_NUMJALON = 0 at normal start of Uproc
{
$UNI_DIR_EXEC/uxset msg "    *** Passage STEP 0 *** "
}
STEP1 ()
{
$UNI_DIR_EXEC/uxset step "1"
$UNI_DIR_EXEC/uxset msg "    *** Passage STEP 1 *** "
# Use uxset msg to display normal comments and
# instructions in History Trace.
}
STEP2 ()
{
$UNI_DIR_EXEC/uxset step "2"
$UNI_DIR_EXEC/uxset msg "    *** Passage STEP 2 *** "
# Utililiser uxset msg pour afficher dans la trace
# historique les commentaires et instructions
# à l'opérateur.
}
STEP3 ()
{
$UNI_DIR_EXEC/uxset step "3"
$UNI_DIR_EXEC/uxset msg "    *** Passage STEP 3 *** "
}
STEP4 ()
{
$UNI_DIR_EXEC/uxset step "4"
$UNI_DIR_EXEC/uxset msg "    *** Passage STEP 4 *** "
}
STEP5 ()
{
$UNI_DIR_EXEC/uxset step "5"
$UNI_DIR_EXEC/uxset msg "    *** Passage STEP 5 *** "
}
# Lors d'une reprise la variable S_NUMJALON est alimentée manuellement à
# partir de l'écran de suivi
# d'exploitation. Sinon = 0
```

```
# During recovery the S_NUMJALON variable is manually loaded from the Job
Monitor screen, else=0
#
Iscn=$S_NUMJALON

let Iscn=Iscn
echo "Iscn = $Iscn"
while [ $Iscn -le 5 ]
do
    echo "Iscn = $Iscn"
    Fnct=STEP$Iscn
    echo "Fnct = $Fnct"
    eval $Fnct
    let Iscn=Iscn+1
done
exit 0
```

10.1.2 Les steps de reprise et les messages dans un script Unix CSH.

```
#!/bin/csh
#Tronc commun de la procédure
goto STEP$S_NUMJALON

# point de démarrage normal
STEP00:

STEP01:
$UNI_DIR_EXEC/uxset step "1"
$UNI_DIR_EXEC/uxset msg "    *** Passage step N° $S_NUMJALON ***"

STEP02:
$UNI_DIR_EXEC/uxset step "2"
$UNI_DIR_EXEC/uxset msg "    *** Passage step N° $S_NUMJALON ***"

STEP99:
exit 0
```

10.2 Annexes AS/400

Les sections décrites ci-après sont spécifiques à l'OS/400.

10.2.1 Exemples de programmes sous OS/400

Paramétrage

Administration

```
CALL PGM(UXSHW) PARM(NODE 'TNODE=Asterix')
```

Affiche la définition du nœud Asterix.

```
CALL PGM(UXLST) PARM(MU LOCAL)
```

Affiche la liste des unités de gestion résidant sur le nœud local.

```
CALL PGM(UXLST) PARM(DOM)
```

Affiche la liste des domaines définis et leurs libellés.

```
CALL PGM(UXSHW) PARM(MUD 'MU=SIEGE' 'APPL=*')
```

Affiche la définition des répertoires de l'Unité de Gestion SIEGE pour toutes les applications.

```
CALL PGM(UXSHW) PARM(USER 'USER=ADMIN')
```

Affiche la définition de l'utilisateur ADMIN.

Classes et ressources

```
CALL PGM(UXLST) PARM(CLASS)
```

Affiche la liste des classes définies.

```
CAL PGM(UXSHW) PARM(RES 'RES=D_*')
```

Affiche la définition des ressources.

Uprocs

```
CALL PGM(UXADD) PARM(UPR EXP 'UPR=TEST' 'LABEL=CHARGEMENT DU FICHIER' +  
'FPERIOD=D' 'APPL=BK' 'MEMO=0' 'NBPER=1')
```

Crée l'Uproc TEST version 000 label "Chargement du fichier" dans l'environnement courant (société, nœud) en espace exploitation, période fonctionnelle : jour, fichier de commande interne, application BK (domaine T), mémorisée 1 exécution 1 période.

Sessions

```
CALL PGM(UXSHW) PARM(SES EXP 'SES=D_LOAD_BCK' LNK)
```

Affiche la Session créée précédemment avec les Uprocs qui la composent et les liens entre elles.

Calendriers

```
CALL PGM(UXDLT) PARM(CAL EXP 'MU=SIEGE' 'SINCE=2008' 'BEFORE=2010')
```

Supprime les années 2008, 2009 et 2010 du calendrier de l'Unité de Gestion SIEGE de l'environnement courant (société, nœud et espace).

Tâches

```
CALL PGM(UXADD) PARM(TSK EXP 'SES=D_LOAD_BCK' +  
'UPR=D_LOAD_FIL' 'MU=SIEGE' TECHINF 'TYPE=S' 'USER=ADMIN')
```

Crée une Tâche à partir de la Session D_LOAD_BCK, Uproc entête D_LOAD_FIL sur l'Unité de Gestion SIEGE, non modèle dans l'espace d'exploitation. La Tâche est planifiée, le compte de soumission est ADMIN. Cette tâche ne peut pas créer de lancement car, dans l'état, aucune date n'a été définie.

```
CALL PMG(UXLST) PARM(TSK EXP 'UPR=D_LOAD_FIL' +  
'SES=D_LOAD_BCK' 'MU=SIEGE' FULL)
```

Affiche la liste l'ensemble des caractéristiques de la tâche.

Exploitation

Ressources

```
CALL PGM(UXCNR) PARM(RES 'RES=PRIO_ORDO' 'ESP=X' +  
'UPR=IU_DT2' 'MU=S01')
```

Libère la réservation faite sur la ressource PRIO_ORDO par la Tâche définie par l'Uproc IU_DT2, l'Unité de Gestion S01 dans l'espace exploitation.

Lancements

```
CALL PGM(UXLST) PARM(FLA FULL)
```

Affiche la liste des lancements et toutes les informations associées.

Événements

```
CALL PGM(UXLST) PARM(EVT EXP FULL)
```

Affiche la liste des événements de l'espace exploitation ainsi que toutes les informations associées.

Suivi d'exploitation

```
CALL PGM(UXPUR) PARM(CTL EXP 'UPR=D_*' 'MU=SIEGE' +  
'SINCE=(20130101,1200)' 'BEFORE=(20130201,1200)')
```

Purge le suivi des exécutions de toutes les exécutions de l'Uproc IU_TEST version 001 sur l'Unité de Gestion SIEGE entre le 1er janvier 2013 à 12 heures et le 1er février 2013 à 12 heures pour une date au format YYYYMMDD.

Automates

```
CALL PMG(UXEND) PARM(ATM EXP ALL)
```

Arrête tous les automates de l'espace exploitation.

Historique des exécutions

```
CALL PGM(UXSHW) PARM(HCX EXP 'UPR=D_LOAD_FIL' 'NUPR=*' +  
'SES=D_LOAD_BCK' 'NSESS=*' 'MU=SIEGE' VAR)
```

Affiche les variables de l'Uproc D_LOAD_FIL s'exécutant pour la session D_LOAD_BCK et l'Unité de Gestion SIEGE.

Statistiques

```
CALL PGM(UXDLT) PARM(STA EXP 'UPR=D_LOAD_FIL' 'MU=SIEGE')
```

Supprime la tâche de l'Uproc D_LOAD_FIL sur l'Unité de Gestion SIEGE de la liste des statistiques.

Gestion de batch

```
CALL PGM(UXORDRE) PARM('UPR=1ST_JOB' 'MU=SATURN' +  
'PARAM=34,/tmp,type:2')
```

Provoque la tâche constituée par l'Uproc 1ST_JOB sur l'Unité de Gestion SATURN avec les paramètres indiqués.

10.2.2 Langage de programmation CLP

Taille écran d'un fichier source OS/400

Vous avez sûrement remarqué le signe "+" à la fin de certaines lignes d'un fichier source OS/400. Le signe "+" signifie : "cette commande se poursuit à la ligne suivante".

Sur OS/400, la taille des fichiers est basée sur la longueur de la ligne affichée à l'écran. Le signe "+" est donc utilisé pour poursuivre la commande sur la ligne suivante lorsque le nombre maximum de caractères d'une ligne est atteint.

Pour le développement, vous pouvez travailler directement sur un fichier source sur le serveur, ou sur un fichier texte sur votre poste de travail local. Ce dernier implique de transférer le fichier en utilisant le navigateur d'iSeries, qui le convertira en fichier source, avec une longueur maximum de ligne déclarée par son interface.

Structure des programmes

Début

Un programme CLP commence par PGM.

Fin

Un programme CLP se termine par ENDPGM.

Labels

Des labels peuvent être utilisés pour définir des sauts au sein d'un programme CLP. La commande GOTO est utilisée pour sauter à la ligne marquée par ce label et continuer l'exécution à partir de ce point. La syntaxe est la suivante :

```
JALON00 :
```

Gestion des variables

Des variables locales peuvent être utilisées dans un programme CLP. Bien qu'il n'existe pas ce genre de variables d'environnement, il est toutefois possible de récupérer des variables de Dollar Universe dans un programme, en les déclarant et en obtenant les valeurs manuellement. Des variables peuvent également être passées à un programme comme paramètres lors du calcul d'un pointeur.

Déclarer une variable locale :

```
DCL          VAR(&name)      TYPE(*type)  LEN(n)
```

NAME: nom de la variable à déclarer.

TYPE: Type de la variable :

- *CHAR
- *DEC
- *LGL

Exemple – Déclarer la variable UNI_TEST de type *CHAR et de longueur 1

```
DCL          VAR(&UNI_TEST)  TYPE(*CHAR)  LEN(1)
```

Modifier une variable locale :

```
CHGVAR  VAR(&name)  VALUE('value')
```

NAME: Nom de la variable à modifier.

VALUE: Nouvelle valeur de la variable.

Exemple – Modifier VNOM avec la valeur IDF

```
CHGVAR      VAR(&VNOM)  VALUE('IDF')
```

Paramètres du C.L.

Une valeur qu'un programme d'appel passe à un programme appelé s'appelle généralement un paramètre de programme. Vous indiquez un paramètre de programme en indiquant une valeur ou un nom de variable pour le mot-clé de PARM dans le programme d'appel.

Le paramètre PARM du programme d'appel peut également contenir une liste de variables. Quand le cas se présente, le programme appelé reçoit le même nombre de valeurs, chacun dans un paramètre séparé.

Par exemple, la commande PGM suivante liste deux paramètres d'entrée :&Cusnbr et &Cusname

```
PGM PARM(&Cusnbr &Cusname)
```

Passer une valeur d'une variable dans un programme appelé, a le même effet qu'assigner une valeur initiale à cette variable. Puisque le programme d'appel passe déjà une valeur initiale, vous ne pouvez pas assigner une valeur initiale à une variable d'entrée dans un programme appelé.

Définir des paramètres dans un programme appelé :

Le CL ne supporte que trois types de variables :

- caractère (*CHAR).
- décimal (*DEC).
- logique (*LGL).

Par conséquent, toutes les variables d'entrée d'un programme appelé doivent appartenir à l'un de ces types.

Exemple:

```
PGM PARM(&Name &Number &Flag)
DCL &Name *CHAR 9
DCL &Number *DEC (15 5)
DCL &Flag *LGL
```

En outre, vous devez coder les paramètres de la commande PGM dans le même ordre dans lequel ils ont été passés.

```
CALL MYPROGRAM + PARM('Christine' 22.5 '1')
```

Passer un littéral à un programme appelé :

Pour passer un littéral à un programme appelé, codez-le dans le paramètre de PARM de la commande CALL. Si vous passez seulement un paramètre, il peut-être reconnu par sa position:

```
CALL MYPROGRAM 'Monday'
```

Vous pouvez spécifier jusqu'à 50 paramètres. Quand vous en listez plus d'un, vous devez entourer la liste de parenthèses, comme dans toute liste de valeurs :

```
CALL MYPROGRAM + PARM( 'Monday' 'Tuesday' + 'Wednesday' )
```

Les paramètres passés depuis une ligne de commande OS/400 doivent toujours être des littéraux, car les variables n'existent que dans les programmes.

Passer des littéraux de type caractères et des constantes logiques :

Quand des littéraux de type caractères sont passés, le programme appelé doit déclarer les variables d'entrée comme étant de type *CHAR.

Le programme appelé dans le dernier exemple doit alors contenir les commandes suivantes :

```
PGM PARM(&Day1 &Day2 &Day3)
DCL &Day1 *CHAR 9
DCL &Day2 *CHAR 9
DCL &Day3 *CHAR 9
```

 Notez que les variables sont déclarées en tant que variables *CHAR de neuf octets, même si deux littéraux contiennent moins de neuf caractères (lundi et mardi). Quand vous déclarez les variables utilisées comme paramètres, il est sage de les définir assez longs pour contenir la plus longue valeur qui pourrait leur être passée.

Quand une variable de type *CHAR est plus longue que la valeur qui lui est passée, la valeur est justifiée à gauche et complétée à droite avec des blancs.

Dans l'exemple précédent, les valeurs des variables &day1, &day2 et &Day3 pourraient être Monday, Tuesday, et Wednesday respectivement (complétées par des blancs).

Si le littéral passé est plus long que la variable d'entrée les caractères supplémentaires sont tronqués.

Passer des littéraux de type décimal :

Vous pouvez également passer des littéraux de type décimal comme paramètres d'une ligne de commande OS/400 ou d'un programme CL :

```
CALL MYPROGRAM + PARM(5 3.1416 129.95)
```

Quand vous passez des littéraux de type décimal, le programme appelé doit déclarer les variables d'entrée correspondantes comme étant de type *DEC (15 5) sinon, le programme appelé se terminera en erreur avec une erreur de donnée décimale.

Exemple:

```
PGM PARM(&Number1 &Number2 &Number3)
DCL &Number1 *DEC (15 5)
DCL &Number2 *DEC (15 5)
DCL &Number3 *DEC (15 5)
```

Passer des variables à un programme appelé :

En plus de passer des littéraux comme paramètres, vous pouvez passer des valeurs de variables depuis un programme CL.

Par exemple :

```
CALL PROGRAM1 PARM(&Name)
CALL PROGRAM2 PARM(&Name &Address &Phone)
CALL PROGRAM3 PARM(&Radius 3.1416 &Area)
```

Le premier exemple passé la valeur courante de la variable &Name au programme appelé.

Le second exemple passe la valeur courante des trois variables.

Le troisième exemple passe une combinaison de variables et de littéral au programme PROGRAM3.

Lorsque vous utilisez des variables comme des paramètres de programme, vous devez bien sûr déclarer les variables dans le programme appelé, comme dans l'exemple suivant :

```
PROGRAM1:
PGM DCL &Name *CHAR 10
DCL &Department *DEC (5 0)
CHGVAR &Name 'Mary'
CALL PROGRAM2 +
  PARM(&Name &Department)
RETURN
ENDPGM
```

Le nom des variables dans les programmes appelants et appelés n'ont pas besoin d'être identiques, mais le type des données et les longueurs doivent être cohérents. Notez que les variables décimales à passer n'ont pas besoin d'être définies (par exemple &department comme (15 5)), cependant, chaque variable décimale doit être définie avec le même type de données et la même longueur que la variable de programme correspondante à passer.

Les commandes CL du programme appelant et appelé peuvent être écrites dans n'importe quel ordre et ne doivent pas nécessairement déclarer les variables avec les mêmes noms. L'exemple suivant montre à quoi pourrait ressembler le programme PROGRAM2 :

```
PROGRAM2:
PGM PARM(&Alpha &Dept)
DCL &Dept *DEC (5 0)
DCL &Alpha *CHAR 10
IF (&Alpha = 'Mary') DO
CHGVAR &Dept VALUE(12600)
ENDDO
RETURN
ENDPGM
```

Si nécessaire, les paramètres doivent apparaître dans le même ordre que dans la commande CALL du programme PROGRAM1.

Commandes de soumission

CALL

- Commande synchrone.
- Des paramètres peuvent être passés au programme appelé.
- L'environnement OS/400 ne peut pas être défini (JOBQ, OUTQ, JOBQ, ...).

SMBJOB

- Commande asynchrone
- Des paramètres ne peuvent pas être passés au programme appelé.
- L'environnement OS/400 peut être défini : JOBQ, OUTQ, JOBQ, DATE, JOB.

Variables d'environnement de Dollar Universe sous OS/400 :

La commande SMBJOB étant asynchrone, la commande CALL doit être utilisée.

Dollar Universe fournit un mécanisme interne permettant de déclarer l'environnement OS/400 par l'intermédiaire de variables d'Uproc.

Les variables disponibles sont : UX<nom_OS/400>

- UXJOBQ
- UXJOB
- UXOUTQ
- UXJOB
- UXDATE...

Structure IF/THEN

Une structure de test "IF/THEN" est disponible dans le CLP. Le mot clé ELSE n'existe pas et doit être remplacé par une cascade de tests IF.

Format :

```
IF COND(état) THEN( faire quelque chose )
```

Etat : test qui doit retourner Vrai ou Faux.

Faire quelque chose: commande à exécuter si le test est Vrai. Plusieurs lignes peuvent être ajoutées en utilisant le signe '+' selon la taille de la ligne OS/400.

Exemple 1 : tester un numéro de jalon :

```
IF COND(&S_NUMJALON *EQ '00') THEN(GOTO +
                                CMDLBL(JALON00))
```

Exemple 2 : tester la valeur d'une chaîne :

```
IF COND((&PGMPARM *NE ' ') *AND (&CMD *NE ' ')) +
    THEN(CALL PGM(&CMD) PARM(' &PGMPARM'))
```

Etat de terminaison de l'Uproc

Dollar Universe interprète le dernier message envoyé avec l'identifiant 9898 juste avant le RETURN comme une erreur.

```
SNDPGMMSG MSGID(CPF9898) MSGF(QCPFMSG) MSGTYPE(*ESCAPE)
```

Un RETURN sans message préalable est interprété comme une terminaison correcte de l'Uproc.

Compilation

Le CLP étant un langage de programmation, le code source doit être compilé avant de pouvoir être appelé par une commande externe de Dollar Universe.

Dollar Universe ne peut pas exécuter un code source compilé, les programmes doivent donc être traités comme des scripts externes. Les scripts internes ne peuvent pas être utilisés.

Une exception peut être faite pour la ligne de commande. Une commande écrite sur une seule ligne peut être exécuté par un script interne, sans caractère blanc ni retour en fin de fichier.

```
Crtclpgm pgm ('LIBRARY\PROGRAM' 'LIBRARY/SOURCEFILE' ... )
```

Les paramètres sont :

- Program: fichier de sortie, résultat de la compilation.
- Library: librairie de sortie où le programme doit être intégré.

- Source File: Fichier d'entrée CL. Le type du fichier doit être *.SRC.
- Library: librairie contenant le fichier source.
- Source member: Membre d'un fichier source à compiler.

11 A propos d'Automic, de la communauté et des services

Ce sujet présente la société Automic Software et comment tirer parti de toutes les solutions que nous vous proposons.

Vous pouvez également accéder à la documentation en ligne d'Automic à l'adresse docs.automic.com.

11.1 La société Automic Software

Automic Software est dédiée à l'automatisation de l'entreprise.

Automic est la plateforme la plus complète pour l'automatisation de l'entreprise. Fondée en 1985, Automic a lancé la plate-forme d'automatisation indépendante la plus vaste, déployée à l'échelle mondiale qui renforce les applications l'infrastructure et l'entreprise. Maintenant que la consommation de l'informatique s'accélère, Automic ré-imagine comment les organisations s'intègrent dans les modèles de services de nouvelle génération comme le Cloud, Big Data et DevOps. Aujourd'hui, nos logiciels automatisent des dizaines de millions d'opérations par jour pour plus de 2000 clients dans le monde entier. Nous défions la pensée conventionnelle, permettant à nos clients d'être plus rapides, plus intelligents dans le contrôle des opérations. Automic – le standard dans l'automatisation de l'entreprise.

Pour en savoir plus, visitez notre site web www.automic.com.

11.2 La communauté Automic

Vous voulez communiquer avec d'autres utilisateurs Automic pour comparer des idées ou apprendre comment d'autres s'attaquent aux problèmes que vous rencontrez ?

Partagez avec d'autres utilisateurs du monde entier pour apprendre comment ils optimisent l'automatisation de l'entreprise à l'aide d'Automic. Dialoguez avec l'équipe Automic pour obtenir des trucs et astuces directement de la source.

Rejoignez la communauté Automic et devenez un initié Automic pour être parmi les premiers à recevoir des nouvelles sur les nouveaux produits et événements, parfois même avant qu'ils ne soient annoncés !

11.3 Centre de téléchargement

Assurez-vous que vous utilisez tout le potentiel de nos produits.

Le Centre de téléchargement est l'emplacement où vous trouverez les téléchargements des produits, la documentation et les informations sur les nouvelles mises à jour et la résolution des problèmes de votre solution Automic. Tout est regroupé : les mises à jour, les release notes et tous les manuels utilisateurs. Vous trouverez également la description des patches, les problèmes connus et les solutions de contournement.

11.4 Support technique

Vous pouvez faire confiance à notre équipe de support.

Notre équipe de professionnels est à votre service, n'importe où et n'importe quand. Plusieurs centres de supports situés en Europe, en Amérique du nord et en Asie Pacifique constituent le cœur de l'organisation du support Automic.

Vos experts Automic de nos équipes de support technique sont au plus à quelques heures de vol, quel que soit le continent sur lequel vos filiales et vos centres de données sont situés. Automic Software est conçue pour fournir une connectivité mondiale à des entreprises internationales. Vous utilisez les logiciels Automic à l'échelle mondiale, vous pouvez donc vous attendre à un service mondial.

11.5 Formation et services

Vous voulez en savoir plus sur les solutions Automic ?

Nous proposons une gamme de formations sur la façon de tirer le meilleur parti de votre solution Automic. Selon votre localisation, vous pouvez participer aux sessions de formations organisées au centre de formation d'Automic ou nous pouvons organiser des sessions de formation personnalisées sur le site de votre entreprise, selon ce qui vous convient le mieux.

Visitez le site du centre de formation pour obtenir des informations détaillées sur les cours proposés.