



ANNEXES

Formation Développement

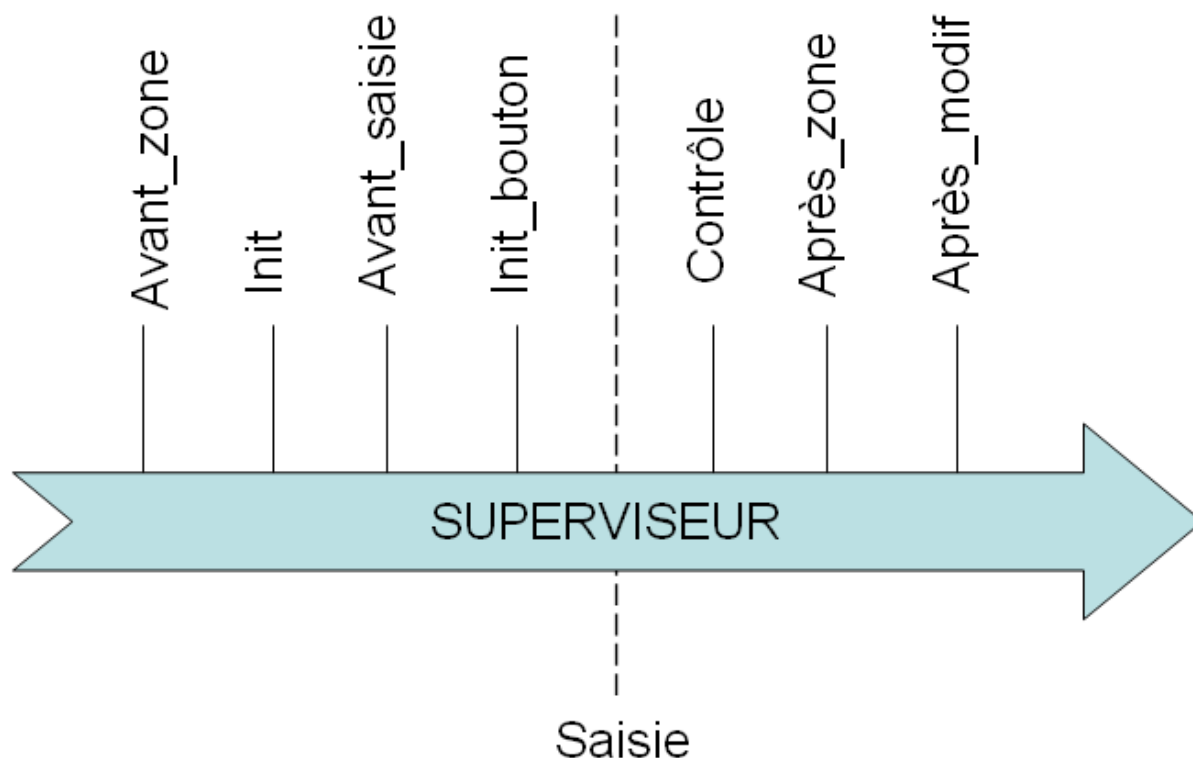
IUT PARIS 5 René DESCARTES

Table des matières

1.	LISTE ET DESCRIPTIONS DES ACTIONS SUR CHAMP	3
1.1.	DYNAMIQUE DES ACTIONS SUR CHAMP	3
1.2.	CONTEXTES D'APPELS ET EMPLOI	3
1.2.1.	Avant_zone (AV_)	3
1.2.2.	Init (D_).....	3
1.2.3.	Avant_saisie (AS_)	3
1.2.4.	Contrôle (C_).....	4
1.2.5.	Après_zone (AP_)	4
1.2.6.	Après_modif (AM_)	4
1.3.	MENU CONTEXTUEL (BOUTONS SUR CHAMPS).....	4
1.3.1.	Sélection (S_).....	4
1.3.2.	Init_bouton (IB_).....	4
1.3.3.	Bouton n (Bn_).....	4
1.4.	VARIABLES DE BAS DE TABLEAU (TYPE ABS)	4
1.4.1.	Avant_ligne (AVANT_)	4
1.4.2.	Après_ligne (APRES_).....	4
2.	LISTE ET DESCRIPTIONS DES ACTIONS MODELES	5
2.1.	MODELE OBJET	5
2.1.1.	Création - dynamique.....	5
2.1.2.	Création – contexte et emploi.....	5
2.1.3.	Modification – dynamique	6
2.1.4.	Modification – contexte et emploi	6
2.1.5.	Suppression – dynamique	7
2.1.6.	Suppression – contexte et emploi	7
2.1.7.	Gestion des boutons - dynamique.....	7
2.1.8.	Gestion des boutons – contexte et emploi	7
2.1.9.	Gestion des browser :	8
2.1.10.	Gestion des browser : le picking	8
2.2.	MODELE SAISIE FENETRE	8
2.2.1.	OUVRE	8
2.2.2.	TITRE	8
2.2.3.	DEBUT	8
2.2.4.	APRES_MODIF.....	8
2.2.5.	FIN	8
2.2.6.	FINSAI	8
2.2.7.	ABANDON	8
2.3.	MODELE TRAITEMENT STANDARD (UTILISABLE EN MODE BATCH)	9
2.3.1.	INIT.....	9
2.3.2.	INIT_DIA	9
2.3.3.	CONTROLE.....	9
2.3.4.	EXEC.....	9
2.3.5.	TERMINE.....	9
2.4.	MODELE SELECTION DANS TABLE	9
2.4.1.	SEL_TABLE.....	9
2.4.2.	VERF_TABLE.....	9
2.4.3.	FIN_TABLE	9
2.5.	MODELE SELECTION DANS LISTE.....	9
2.5.1.	SEL_LISTE.....	9
2.5.2.	VERF_LISTE	10
2.5.3.	FIN_LISTE.....	10

1. Liste et descriptions des actions sur champ

1.1. Dynamique des actions sur champ



1.2. Contextes d'appels et emploi

1.2.1. Avant_zone (AV_)

Contexte : Avant l'affichage de la zone. Permet de définir le format, affecter une couleur, conditionner l'affichage.

Emploi :

Eviter l'affichage et la saisie de la zone : mkstat = 2.

Idem + réinitialisation du champ : mkstat = 1

1.2.2. Init (D_)

Emploi :

Initialisation de la zone en alimentant la variable VALEUR.

1.2.3. Avant_saisie (AS_)

Contexte : Avant la saisie de la zone.

Emploi :

Permet d'éviter la saisie du champ : mkstat = 2.

Permet d'effacer et d'éviter la saisie du champ : mkstat = 1.

Permet de supprimer des valeurs d'une menu local modifiable : GMENLOC(n) = 1.

1.2.4. Contrôle (C_)

Contexte : après la saisie de la zone.

Emploi :

Permet de contrôler la validité du champ saisi. Et d'afficher un message d'erreur.

Permet de bloquer la saisie tant que le champ saisi est invalide : mkstat = 2 ou mkstat=1

1.2.5. Après_zone (AP_)

Contexte : après la saisie d'une zone.

Emploi :

Permet l'alimentation et l'affichage d'un champ suivant.

Exemple : [M :ABC]champ = « toto » : Affzo [M :ABC]champ

1.2.6. Après_modif (AM_)

Contexte : après la saisie si la zone a été modifiée et est valide.

Emploi :

Identique à Après_zone.

1.3. Menu contextuel (boutons sur champs)**1.3.1. Sélection (S_)**

Contexte : Fenêtre de sélection.

Remarque : après la sélection, le curseur se positionne automatiquement sur le champ suivant. Pour bloquer le curseur, il suffit de positionner mkstat à 2.

1.3.2. Init_bouton (IB_)

Emploi :

Permet d'attribuer un texte à un bouton. Si aucun texte n'est attribué à un bouton, il est désactivé.

Attribution d'un libellé au bouton 4 : GBOUT4 = « texte »

Désactivation du bouton 2 : Raz GBOUT2.

1.3.3. Bouton n (Bn_)

Emploi :

Exécution du bouton n.

Positionner mkstat à 4 pour déclencher l'activation du bouton fenêtre 'Enregistrer'.

1.4. Variables de bas de tableau (type ABS)**1.4.1. Avant_ligne (AVANT_)**

Contexte : Avant la saisie d'une ligne dans un bloc de type tableau.

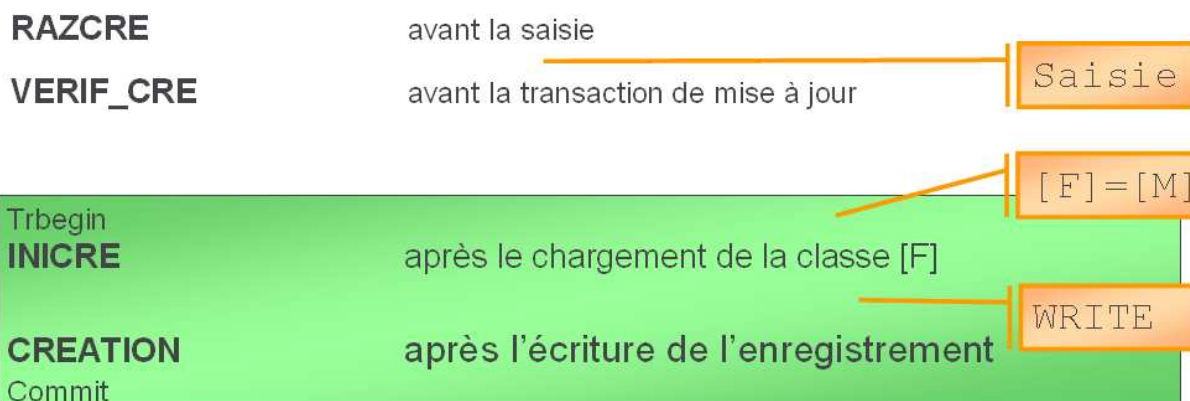
1.4.2. Après_ligne (APRES_)

Contexte : Après la saisie d'une ligne dans un bloc de type tableau.

2. Liste et descriptions des actions modèles

2.1. Modèle OBJET

2.1.1. Création - dynamique



APRES_CRE après la transaction de mise à jour

AB_CREATION si abandon de la transaction

2.1.2. Création – contexte et emploi

- RAZCRE (RAZDUP pour la duplication) : sert à initialiser l'écran de saisie (valeur et activation/désactivation des zones)
- VERIF_CRE : sert à effectuer des contrôle, à interrompre la création d'un enregistrement en positionnant la variable OK à 0.
- INICRE : sert à alimenter la classe [F]. Il est possible d'interrompre la création en positionnant la variables GOK à 0.
- CREATION : sert à créer, mettre à jour, supprimer des enregistrements dans des tables secondaires. Il est possible d'interrompre la transaction en positionnant la variable GOK à 0.
- APRES_CRE : sert à créer, mettre à jour, supprimer des enregistrements dans des tables secondaires. Il est possible de lancer l'impression d'un état.
- AB_CREATION

2.1.3. Modification – dynamique

V I S U	LIENS	après la lecture de l'enregistrement	[M] = [F]
	AVANT_MOD	dès le début de la saisie	Saisie
	VERIF_MOD	avant la transaction de mise à jour	
	AVANT_MODFIC	après la lecture enreg. Avec verrouillage	[F] = [M]
	INIMOD	après le chargement de la classe [F]	REWRITE
	MODIF	après la ré-écriture de l'enregistrement	
	APRES_MOD	après la transaction de mise à jour	
	AB_MODIF	si abandon de la transaction	

2.1.4. Modification – contexte et emploi

- LIENS : sert à alimenter des variables, lire des tables supplémentaires, alimenter les écrans de saisie, activer / désactiver des champs des écrans de saisie (Actzo, Diszo, Grizo).
- AVANT_MOD : sert à contrôler l'autorisation de la saisie. Il est possible d'empêcher toute modification en positionnant la variable OK à 0.
- VERIF_MOD : sert effectuer des contrôles globaux. Il est possible d'interrompre le traitement en positionnant la variable OK à 0.
- AVANT_MODFIC (peu utilisé): il est possible d'interrompre la transaction en positionnant la variable GOK à 0.
- INIMOD : sert à alimenter la classe [F] (la table). Il est possible d'interrompre la transaction en positionnant la variable GOK à 0.
- MODIF : sert à créer, mettre à jour, supprimer des enregistrements dans des tables secondaires. Il est possible d'interrompre la transaction en positionnant la variable GOK à 0.
- APRES_MOD : sert à mettre à jour des tables secondaires.
- AB_MODIF

2.1.5. Suppression – dynamique

[M] = [F]

VERF_ANU avant la transaction de suppression

ANNULE

avant la suppression de l'enregistrement

DELETE

AP_ANNULE après la transaction de suppression

2.1.6. Suppression – contexte et emploi

- VERF_ANU : sert à contrôler l'autorisation de la suppression. Il est possible d'interrompre le traitement en positionnant la variable OK à 0.
- ANNULE : sert à créer, mettre à jour, supprimer des enregistrements dans des tables secondaires. Il est possible d'interrompre la transaction en positionnant la variable GOK à 0.
- AP_ANNULE

2.1.7. Gestion des boutons - dynamique

SETBOUT

Désactivation des boutons

Utiliser le sous-pro **VIREBOUT** pour enlever le code bouton(s) de la variable **CHAINE**

AVANTBOUT

Avant l'exécution du bouton

le code bouton est disponible dans la variable **BOUT**

Action Dico déclaré dans la fenêtre

EXEBOUT




Après l'exécution du bouton

le code bouton est disponible dans la variable **BOUT**





2.1.8. Gestion des boutons – contexte et emploi

- SETBOUT : sert à désactiver des boutons bas d'écran (par défaut tous les boutons sont activés). Désactivation du bouton de suppression : Call VIREBOUT(CHAINE, «A») From GOBJET. Liste des codes boutons à utiliser : C (créer), D (Dupliquer), M (Enregistrer), A (Supprimer), F (Fin).

2.1.9. Gestion des browser :

-  OUVRE_BOITE : sert à stocker le nom des listes gauches
-  FILTRE : sert à filtrer les enregistrements de la liste principale
-  FILGAUCHE : sert à filtrer les enregistrements de la liste principale et des listes secondaires.

2.1.10. Gestion des browser : le picking

-  DEB_PICK : début de sélection d'une ligne ou d'un groupe de lignes. Il est possible d'interrompre le traitement en positionnant la variable OK à 0.
-  PICKE : après la sélection d'un élément. Il est possible d'interrompre le traitement en positionnant la variable OK à 0.
-  DEPICK : après avoir désélectionné un élément. Il est possible d'interrompre le traitement en positionnant la variable OK à 0.
-  FIN_PICK : à la fin de sélection d'une ligne ou d'un groupe de lignes.

2.2. Modèle Saisie Fenêtre

2.2.1. OUVRE

Contexte : en tout début de traitement, avant l'ouverture du masque.

Emploi : Sert à déclarer, alimenter des variables locales, globales, ouvrir des tables. Il est possible d'interrompre le traitement en positionnant la variable FIN à 1.

2.2.2. TITRE

Contexte : avant l'affichage de la fenêtre.

Emploi : Sert à modifier le titre de la fenêtre en alimentant la variable TIT.

2.2.3. DEBUT

Contexte : après l'affichage de la fenêtre.

Emploi : Sert à pré-charger l'écran de saisie. Il est possible d'interrompre le traitement en positionnant la variable FIN à 1.

2.2.4. APRES_MODIF

Contexte : après la saisie de tout champ.

2.2.5. FIN

Contexte : après la saisie de la fenêtre.

2.2.6. FINSAI

Contexte : En fin de traitement.

Emploi : sert à rafraichir la fenêtre appelante.

2.2.7. ABANDON

Contexte : En cas d'abandon.

2.3. Modèle Traitement Standard (Utilisable en mode Batch)

2.3.1. INIT

Contexte : en tout début de traitement.

Emploi : sert à déclarer, alimenter des variables locales, globales, ouvrir des tables. Il est possible d'interrompre le traitement en positionnant la variable FIN à 1.

2.3.2. INIT_DIA

Contexte : exécutée dans le cas de l'utilisation d'une boîte de dialogue.

2.3.3. CONTROLE

Contexte : après la saisie des paramètres de lancement dans la fenêtre de critères.

Emploi : sert à effectuer des contrôles avant d'exécuter le traitement. Il est possible d'interrompre le traitement en positionnant la variable FIN à 1.

Il est également possible d'afficher un message d'attente en positionnant la variable TMP à 1.

2.3.4. EXEC

Contexte : une fois tous les contrôles vérifiés.

Emploi : sert à lancer le traitement. Il est possible d'interrompre le traitement en positionnant la variable FIN à 1.

2.3.5. TERMINE

Contexte : après l'action EXEC.

2.4. Modèle Sélection dans table

2.4.1. SEL_TABLE

Contexte : avant l'affichage de la fenêtre de sélection

Emploi : sert à définir la table utilisée par défaut.

On définit les colonnes de la fenêtre de sélection avec :

NBCOL : nombre de colonnes

COL(indice) : nom du champ alimentant la colonne

TIT(indice) : titre de la colonne

TIT(0) : titre de la fenêtre de sélection

On définit un filtre de sélection en alimentant la variable CRITERE.

2.4.2. VERF_TABLE

Contexte : après la saisie du choix

Emploi : sert à effectuer des contrôles du choix et à affecter des valeurs dans d'autres champs de l'écran.

2.4.3. FIN_TABLE

Contexte : en fin de traitement.

Emploi : sert à ôter un éventuel filtre, à intervenir sur la valeur sélectionnée (variable VALEUR).

2.5. Modèle Sélection dans liste

2.5.1. SEL_LISTE

Contexte : avant l'affichage de la fenêtre de sélection.

Emploi :

Sert à charger la fenêtre de sélection avec :

NBTEX : le nombre de valeurs

TEX(indice) : nom de champ

TEXTE(indice) : valeur affichée dans la fenêtre de sélection

2.5.2. VERF_LISTE

Contexte : après la saisie du choix.

Emploi : sert à contrôler la valeur sélectionnée.

2.5.3. FIN_LISTE

Contexte : en fin de traitement.

Emploi : sert à ôter un éventuel filtre, à intervenir sur la valeur sélectionnée (variable VALEUR).