

# OPTIMA PLC

Manuel utilisateur

26/03/2018

**Optimalog**  
[www.optimalog.com](http://www.optimalog.com)



# Table des matières

<b>Partie I Introduction</b>	<b>1</b>
1 A quoi sert Optima PLC ?.....	1
2 Principes généraux.....	1
Phases du développement.....	1
Licences d'utilisation, identification du client.....	2
Contenu d'un répertoire de projet Optima PLC.....	2
Contenu d'une librairie de projet Optima PLC.....	3
Compilation et génération de code .....	4
Modification du code en ligne.....	4
Gestion des droits .....	4
3 Installation d'Optima PLC.....	4
4 Première utilisation : exemple de création d'un projet.....	8
<b>Partie II Atelier de programmation</b>	<b>11</b>
1 Description des menus.....	11
Menu = .....	11
Langue logiciel.....	11
Langue programmation.....	11
Option Licence.....	11
Menu Projet.....	13
Sélection projet.....	13
Liste des POU.....	14
Editeur Optima PLC.....	14
Gestionnaire de fichiers .....	15
Fenêtre système.....	15
Imprimer .....	15
Quitter .....	16
Menu Fichiers.....	16
Nouvelle librairie.....	16
Ouvrir une librairie .....	16
Sauver toutes les librairies.....	16
Menu Edition.....	16
Chercher .....	16
Remplacer .....	17
Occurrence suivante.....	17
Menu Supervision.....	17
Menu Compilation.....	18
Destination .....	19
Conditions de fonctionnement pour une destination distante.....	19
Paramètres pour une destination distante.....	20
Compilation complète.....	20
Compilation et modification en ligne .....	21
Exécution .....	22
Menu Visuel.....	22
Document suivant.....	22
Document précédent.....	22
Mode vidéo.....	22
Calculatrice.....	23
Calendrier/Horloge.....	23
Format Date/Heure.....	23
Liste des fenêtres .....	24
Options Editeur .....	24
Options générales .....	24
Editeur Pascal.....	25
Editeur ST .....	25

<b>Menu Options.....</b>	<b>26</b>
Compilation générale.....	26
Constantes système.....	27
Editeur de configuration.....	27
Interfaces matérielles.....	27
Fichiers de sortie texte.....	28
Tableaux de mesures.....	30
Divers.....	31
Configuration du démarrage.....	31
Gestion des utilisateurs.....	31
Session Windows pour exécution.....	32
Paramétrage des sessions Windows.....	33
Fichiers défaut.....	33
<b>Menu Aide.....</b>	<b>33</b>
Aide.....	33
A propos d'Optima PLC Workshop.....	34
<b>2 Description de la barre d'icônes.....</b>	<b>35</b>
Barre d'outil de gestion du projet.....	35
Barre d'outil de gestion de l'affichage des fenêtres.....	35
<b>3 Utilisation des librairies.....</b>	<b>36</b>
<b>Principes généraux des librairies.....</b>	<b>36</b>
Contenu d'une librairie.....	36
Tâche.....	36
Variables globales.....	38
Types.....	38
Bloc de programme.....	38
Bloc de fonction.....	38
Fonction.....	39
Sous-groupe.....	39
Arborescences des POU.....	39
Arbre de conception "projet".....	39
Arbre de conception "librairies".....	39
Structure (arbre d'exécution).....	40
Librairies "Projet".....	41
Ordre d'apparition des POU.....	41
Propriétés des POU.....	41
Page Déclarations.....	41
Page Instructions.....	43
Page Type fonction.....	45
Page Langage.....	45
Page Tâche.....	46
Page Infos.....	47
Page Propriétés.....	48
Page Droits.....	50
Tableau des pages visibles en fonction des POU.....	50
Librairies Optima PLC.....	51
<b>Création d'une librairie.....</b>	<b>51</b>
Menu de gestion des librairies.....	51
Créer un POU.....	52
Menu de gestion des POU.....	53
Création de diagrammes Ladder.....	54
Barre d'outil des diagrammes ladder.....	54
Gestion des contacts.....	55
Gestion des sorties.....	55
Négation.....	57
Gestion des diagrammes.....	57
Opérations diverses.....	57
Zoom et impression.....	57
Utilisation de la sélection.....	57
Utilisation de blocs dans un diagramme ladder.....	58
Menus contextuels des diagrammes ladder.....	59
Exemple de création d'un diagramme ladder.....	60
Création de diagrammes FBD.....	60

Barre d'outil d'un diagramme Blocs de fonction.....	61
Menus contextuels d'un diagramme blocs de fonction.....	61
Définition du type de bloc.....	62
Utilisation de variables en entrée et en sortie de bloc.....	63
Chaînage de blocs.....	63
Exemple de diagrammes FBD.....	63
Création de diagrammes SFC (Grafcet).....	64
Barre d'outil d'un diagramme SFC.....	65
Menus contextuels des diagrammes SFC.....	66
Création d'une action.....	66
Qualificateur d'action.....	67
Qualificateur d'action N.....	68
Qualificateurs d'action S et R.....	68
Qualificateur d'action L.....	69
Qualificateur d'action D.....	69
Qualificateur d'action P.....	70
Qualificateurs d'action P1 et P0.....	70
Qualificateur d'action SD.....	71
Qualificateur d'action DS.....	72
Qualificateur d'action SL.....	72
Paramétrage d'une action booléenne.....	73
Paramétrage d'une action "in situ".....	74
Paramétrage d'une action "externe".....	75
Inhibition d'un appel d'action.....	76
Créer un commentaire d'étape.....	77
Manipulation des actions.....	78
Copie des éléments d'un diagramme SFC.....	78
Suppression d'éléments d'un diagramme SFC.....	79
Exemple de création d'un diagramme SFC.....	79
Réinitialisation d'un diagramme SFC.....	84
Création de bloc ST, IL et SFC textuel.....	84
Manipulation des POU d'une librairie.....	85
<b>Impression.....</b>	<b>85</b>
Impression de diagramme.....	85
Impression de POU.....	85
Impression d'une librairie.....	85
Impression d'un projet.....	85

## Partie III Atelier de supervision 85

<b>1 Menu principal de l'application.....</b>	<b>86</b>
<b>Menu fichier.....</b>	<b>86</b>
Menu fichier - Nouveau.....	86
Menu fichier - Sélection projet.....	86
Menu fichier - Enregistrer.....	86
Menu fichier - Ouvrir.....	86
Menu fichier - Enregistrer sous.....	86
Menu fichier - Quitter.....	86
<b>Menu Edition.....</b>	<b>86</b>
Menu édition - Annuler.....	86
Menu édition - Couper.....	86
Menu édition - Copier.....	86
Menu édition - Coller.....	86
Menu édition - Supprimer.....	86
Menu édition - Tout sélectionner.....	86
<b>Menu Outils.....</b>	<b>86</b>
Menu Création de raccourcis.....	86
Menu Traductions utilisateur - Ouvrir un fichier langage.....	87
Menu Traductions utilisateur - Nouveau fichier langage.....	87
Menu Traductions utilisateur - Extraction Auto textes de la page.....	87
Menu Explorateur Apigraf IP.....	88
Menu Gestionnaire d'alarmes.....	89
<b>Menu Affichage.....</b>	<b>89</b>
Menu Affichage - Barres d'outils.....	89



Menu Affichage - Fenêtre Fonction.....	89
Menu Affichage - Explorateur d'objets.....	89
Menu Affichage - Fenêtre Alignement.....	89
<b>Menu Objet.....</b>	<b>90</b>
Menu Objet - Ajouter.....	90
Menu Objet - Supprimer.....	90
Menu Objet - Importer.....	91
<b>Menu Macros.....</b>	<b>92</b>
Menu Macros - Macros locales.....	92
Menu Macros - Macros PLC.....	92
Menu Macros - Macros Utilisateur.....	92
<b>Menu Diagnostics.....</b>	<b>92</b>
Clients TCP/IP données PLC.....	92
Clients OPC.....	93
Clients TCP/IP données Alarmes.....	94
Tableaux de mesure liés en mémoire.....	94
<b>Menu Options.....</b>	<b>95</b>
Menu Options - Options générales.....	95
Menu Options - Edition configuration.....	97
Menu Options - Gestion des utilisateurs.....	97
<b>Menu Utilisateurs.....</b>	<b>98</b>
Changer d'utilisateur.....	98
Changer le mot de passe.....	98
Liste des utilisateurs.....	98
<b>Menu Version.....</b>	<b>99</b>
<b>Menu Aide.....</b>	<b>99</b>
<b>2 Création de pages de supervision.....</b>	<b>99</b>
Principes généraux.....	99
Création d'une page de supervision.....	100
Paramétrage d'une page de supervision.....	101
Aspect.....	101
Police.....	102
Fond.....	103
Grille.....	104
Position / Taille.....	104
Options.....	105
Macros.....	108
<b>Barres d'outils de l'atelier de supervision.....</b>	<b>108</b>
<b>Manipulation des objets de supervision.....</b>	<b>109</b>
Création d'un objet de supervision.....	109
Sélection d'un objet ou d'un groupe d'objets de supervision.....	110
Déplacement et changement de taille d'un objet de supervision.....	110
Changement de taille d'un objet ou d'un groupe d'objets à l'aide de la fonction zoom.....	111
Suppression d'objets de supervision.....	111
Mise en avant plan ou en arrière plan d'un objet.....	112
Copie d'objets à l'aide de la souris.....	112
Saisie des macros pour les objets.....	112
<b>Objets usuels pour la création de pages de supervision.....</b>	<b>113</b>
Objets de l'onglet STANDARD.....	113
Texte.....	113
Case à cocher.....	115
Radio group.....	117
Flèches Haut/Bas.....	119
Objets de l'onglet BOUTONS.....	122
Bouton simple.....	122
Bouton API.....	125
Bouton poussoir type 1.....	130
Bouton à LED.....	132
Bouton de Fonction.....	135
Bouton de Fonction translucide.....	140
Bouton poussoir industriel.....	144
Bouton poussoir.....	147
Objets de l'onglet FENETRES DE SAISIE.....	148

Fenêtre de saisie.....	148
Fenêtre de saisie spéciale.....	153
Fenêtre de saisie Date/Heure.....	157
Fenêtre d'édition avec menu déroulant.....	160
Objets de l'onglet CONTROLES.....	164
Classeur .....	164
Défilement.....	166
Visionneur HTML.....	167
Panneau .....	169
Liste déroulante.....	171
Zone de clic.....	174
Objets de l'onglet DESSIN.....	176
Forme .....	176
Ligne .....	178
Dégradé .....	179
Objets de l'onglet GRAPHIQUE.....	181
Graphe .....	181
Histogramme.....	192
Echelle .....	195
Camembert.....	197
Objets de l'onglet ANIMATIONS / IMAGES.....	199
Animation.....	199
Insertion images successives.....	202
Texte défilant.....	205
Bitmap animé.....	207
Image .....	209
Objets de l'onglet JAUGES.....	211
Jauge .....	211
Vu mètre .....	213
Barre de progression.....	215
Jauge circulaire simple.....	216
Jauge circulaire.....	219
Vu mètre à aiguille.....	223
Objets de l'onglet CURSEURS / TRACKBARS.....	225
Curseur .....	225
Advanced TrackBar .....	228
Barre de défilement.....	231
Objets de l'onglet INTERRUPTEURS.....	233
Bouton Marche/Arrêt.....	233
Lever sans couleur.....	235
Bouton d'arrêt d'urgence.....	235
Objets de l'onglet SELECTEURS.....	236
Sélecteur 2/3 positions.....	236
Sélecteur 2/3 positions sans étiquette.....	238
Objets de l'onglet VOYANTS.....	238
Voyant type 1.....	238
Voyant simple.....	240
Voyant double.....	242
Colonne 3 couleurs.....	244
LED .....	246
Objets de l'onglet EXCEL.....	248
Liaison EXCEL.....	248
Objets de l'onglet CUVE / VANNE.....	251
Cuve .....	251
Vanne .....	254
Vanne bitmap.....	256
Objets de l'onglet Heure/Date.....	258
Affichage Heure.....	258
Affichage Date.....	259
Calendrier .....	261
Fenêtre de saisie Date/Heure.....	263
Objets de l'onglet ALARMES.....	263
Historique des alarmes.....	263
Objets de l'onglet ENREGISTREUR.....	264

Enregistreur de données.....	264
Courbe Enregistreur .....	266
Objets de l'onglet MULTIMEDIA .....	281
Média player .....	281
Flash ShockWave player.....	284
Objets de l'onglet AUTRES.....	286
Recette .....	286
Afficheurs 7 segments.....	289
OLE Container.....	291
Gestionnaire d'évènement.....	293
<b>Explorateur d'objets.....</b>	<b>295</b>
Utilisation de l'explorateur d'objets.....	295
Masquer un objet en conception.....	295
Masquer tous les objets en conception.....	296
Annuler le masquage d'un objet en conception.....	296
Annuler le masquage de tous les objets en conception.....	296
Configurer un objet.....	296
Supprimer un objet.....	297
<b>Appel d'une page de supervision depuis une autre page.....</b>	<b>297</b>
<b>Appel d'une page de supervision avec passage de paramètres.....</b>	<b>297</b>
<b>Fermeture d'une page depuis un bouton .....</b>	<b>297</b>
<b>Gestion des mots de passe.....</b>	<b>298</b>
<b>3 Objets personnalisés.....</b>	<b>298</b>
Enregistrement d'objets Personnalisés.....	298
Suppression d'objets personnalisés .....	299
Importation d'objets personnalisés.....	299
<b>4 Explorateur d'objets.....</b>	<b>300</b>
Explorateur d'objets.....	300
<b>5 Utilisation des macros.....</b>	<b>302</b>
<b>Macros locales.....</b>	<b>302</b>
BringToFrontSuperGWindow (Macro locale).....	302
ChnMdp (Macro locale).....	302
ChnUserLanguage (Macro locale).....	302
Close (Macro locale).....	302
CopyPageBitmap (Macro locale).....	303
DiagSocketPLC.....	303
ExecProg (Macro locale) .....	304
HideCursor (Macro locale).....	304
HideSuperGWindow (Macro locale).....	304
PrintPage (Macro locale).....	305
RunHelp (Macro locale).....	305
RunHelpIndex (Macro locale).....	305
RunHelpTopic (Macro locale).....	305
Show Cursor (Macro locale).....	305
Show SuperGWindow (Macro locale) .....	305
SuperG (Macro locale) .....	306
VisuText (Macro locale) .....	306
WFCopy (Macro locale).....	306
WFCreateDirectory (Macro locale).....	306
WFDelete (Macro locale) .....	306
WFDeleteDirectory (Macro locale).....	306
WFRename (Macro locale) .....	306
WFSetFileAttr (Macro locale).....	307
WNetSChangeFileName (Macro locale).....	307
WNetSCopyFileRemote (Macro locale).....	307
WNetSCreateDirectory (Macro locale) .....	307
WNetSDeleteDirectory (Macro locale).....	307
WNetSReadAndMoveFile (Macro locale).....	307
WNetSWriteAndMoveFile (Macro locale).....	308
WNetSReadFile (Macro locale).....	308
WNetSWriteFile (Macro locale).....	308
WNetSSetFileAttr (Macro locale).....	308
<b>Macros externes (exécutées sur le PLC).....</b>	<b>308</b>

BIP (Macro PLC).....	309
DECR### (Macro PLC).....	309
INCR### (Macro PLC).....	309
INVS A### (Macro PLC).....	309
SET...= (Macro PLC).....	309
<b>Macros utilisateur .....</b>	<b>309</b>
Généralités .....	309
Développement d'une DLL Utilisateur .....	309
Exemple de DLL Utilisateur .....	312
Utilisation d'une DLL Utilisateur dans un projet.....	314
Suppression d'une DLL utilisateur d'un projet.....	314
Consultation des DLL utilisateur .....	314
<b>6 Gestion des traductions.....</b>	<b>315</b>
<b>Ecriture des fichiers de langage .....</b>	<b>315</b>
Création d'un nouveau fichier de langage.....	315
Utilisation des fichiers de langage dans les pages de supervision.....	315
Changement de langue.....	315
Exemple de traduction .....	315
<b>7 Gestion des utilisateurs.....</b>	<b>315</b>
Généralités .....	315
<b>8 Exécution des pages de supervision.....</b>	<b>316</b>
Fichiers créés dans un projet de supervision.....	316
Exécution d'une page depuis l'atelier .....	316
Exécution d'une page avec son icône .....	316
<b>9 Clavier virtuel .....</b>	<b>316</b>
Utilisation du clavier virtuel.....	316
<b>10 Utilisation de liens DDE.....</b>	<b>317</b>
Principes généraux.....	317
Exemple : Création d'un lien DDE avec MICROSOFT® Excel.....	317
Autorisation de modification de variables automates via un lien DDE.....	317
<b>11 Accès aux variables PLC depuis un programme écrit en DELPHI.....</b>	<b>318</b>
Principes généraux.....	318
Création d'un programme DELPHI5 utilisant le paquet USERAPIPLC.....	318
Fonctions accessibles depuis le paquet USERAPIPLC.....	318
Fonction LinkToAPI.....	318
Fonction IsLinkedToAPI.....	319
Fonction AllLinksOK.....	319
Fonction UnLinkToAPI.....	319
Procédure ExecMacro.....	319
Fonction PLCState .....	320
<b>12 Variables locales OPTIMA VIEW.....</b>	<b>321</b>
Principes généraux.....	321
Description des variables locales.....	321
Définition du nombre de variables locales.....	321
Visualisation des variables locales .....	322
<b>13 Serveur OPC : ServApiOPC.....</b>	<b>322</b>
Généralités .....	322
Application ServApiOPC.....	322
Accès aux données du serveur OPC.....	324
Nom du serveur OPC.....	324
Définition d'un groupe.....	324
Définition d'un item.....	324
Définition des données en lecture/écriture.....	324
<b>14 Utilisation de l'atelier en mode indépendant.....</b>	<b>325</b>
<b>Installation et démarrage de l'application de Supervision.....</b>	<b>325</b>
Installation de l'application.....	325
Démarrage de l'application.....	326
<b>Fenêtre de sélection de projet.....</b>	<b>326</b>
Sélection d'un projet.....	326

## Partie IV Runtime Optima PLC 328

<b>1 Généralités.....</b>	<b>328</b>
Démarrage de l'application.....	328
Arrêt de l'application.....	329
Gestion des utilisateurs .....	329
Statut de l'application.....	329
Performances et temps-réel.....	330
<b>2 Description de l'environnement.....</b>	<b>331</b>
Fenêtre de base.....	331
Rappel de la fenêtre.....	333
Barre de statut.....	333
<b>3 Description des menus.....</b>	<b>334</b>
<b>Menu Contrôle .....</b>	<b>334</b>
Menu Options.....	334
Option Licences.....	336
Option Quitter.....	337
<b>Menu Utilisateurs .....</b>	<b>337</b>
<b>Menu Automate .....</b>	<b>337</b>
Option Supervision.....	338
Option Tâches .....	338
Option Graficets.....	338
Option Variables standard.....	339
Option Expressions .....	340
Option Explorateur.....	341
Option Sorties Texte.....	341
Option Ports sériels .....	342
Option Drivers.....	343
<b>Menu Fenêtres.....</b>	<b>343</b>
<b>Menu Diagnostics.....</b>	<b>344</b>
Option Debugger.....	345
Option Performances.....	345
Option Journal de bord.....	347
Option Serveur OPC.....	348
Option Liste des sockets.....	349
Option "Générer un enregistrement de diagnostic".....	349
<b>Menu Aide.....</b>	<b>350</b>
<b>4 Problèmes possibles au démarrage du runtime .....</b>	<b>351</b>
Pas de fichier de données "retain" au démarrage de l'application.....	351
Modification de la structure des variables "retain".....	351
Absence de données concernant les utilisateurs.....	353
Problème de démarrage d'un driver de matériel.....	353

## Partie V Gestion des utilisateurs 354

<b>1 Présentation .....</b>	<b>354</b>
<b>2 Menu Utilisateurs.....</b>	<b>355</b>
Option Changer d'utilisateur.....	355
Option Changer le mot de passe .....	355
Option Liste des utilisateurs .....	356

## Partie VI Programmation 356

<b>1 Types de données.....</b>	<b>357</b>
Entiers .....	357
Réels .....	357
Durée .....	358
Horodatage.....	358
Chaîne de caractères.....	359
Caractère.....	359
Cordon de bits .....	359
Types génériques.....	360

Types dérivés .....	360
Structure de données .....	360
Types énumérés .....	361
Plages de valeurs .....	361
Tableaux de données .....	361
<b>2 Variables prédéfinies dans Optima PLC .....</b>	<b>361</b>
Tableaux de variables .....	361
Cas particuliers .....	363
Alarmes A .....	363
Temporisations DT .....	363
Variables diverses .....	363
<b>3 Déclarations .....</b>	<b>364</b>
Déclarations de variables .....	364
Attributs de variables .....	365
Instances de POUs .....	365
<b>4 Valeurs constantes .....</b>	<b>366</b>
Constantes numériques .....	366
Constantes chaînes de caractères .....	366
Constantes de durée .....	367
Constantes de date .....	367
Valeurs constantes typées .....	368
<b>5 Initialisation de variables .....</b>	<b>368</b>
Initialisation des types et variables simples .....	368
Initialisation de types et variables multi-éléments .....	368
<b>6 Programmation en texte structuré (ST) .....</b>	<b>369</b>
Expressions .....	369
Instructions .....	370
Affectation .....	371
Appel de fonction .....	371
<b>7 Programmation en liste d'instructions (IL) .....</b>	<b>371</b>
Opérateurs, modificateurs, opérands .....	371
Appel de fonctions .....	373
Appel de blocs de fonction .....	374
<b>8 Programmation en langage Apigraf .....</b>	<b>375</b>
Instruction booléenne type .....	375
Condition (expression booléenne) .....	376
Exemple de conditions et d'actions .....	377
Détection de fronts .....	378
Mode d'évaluation des conditions booléennes .....	380
Opérations numériques .....	380
Ecriture sur fichier de sortie texte .....	382
Instructions Grafcet .....	382
Déclaration d'une variable de séquence .....	382
Définition du code d'une séquence .....	382
Définition d'une étape de séquence .....	382
Fin d'une séquence .....	383
Saut à une étape de la séquence .....	383
Exemples de séquences .....	383
Transposition d'un Grafcet en langage Apigraf .....	384
Différences avec le langage Apigraf originel .....	385
<b>9 Programmation avancée .....</b>	<b>387</b>
Techniques de gestion du temps .....	387
Affichage d'informations, d'erreurs, de demandes de confirmation .....	389
Utilisation des tableaux de mesures standards .....	389
Utilisation des fichiers de sortie texte .....	390

## Partie VII Modifications "en ligne" 390

<b>1 Application .....</b>	<b>390</b>
Modifications du programme de l'application .....	390
Modalités de chargement des modifications .....	392

2 Modifications de la supervision.....	392
--	-----

## Partie VIII Exemples 392

1 Création d'un programme.....	392
2 Exemple 1 : Démarrage moteur.....	393
Fonctionnement.....	393
Déclarations.....	393
ProgrammeST.....	394
ProgrammeFBD.....	395
ProgrammeLD.....	395
ProgrammeSFC.....	395
3 Exemple 2 : Porte automatique.....	396
Fonctionnement.....	396
Déclarations.....	396
Fonctionnement.....	397
Validation du code.....	398
4 Exemples fournis avec Optima PLC.....	400

## Partie IX Drivers Optima PLC 400

1 Advantech.....	400
Carte Advantech PCL-728.....	400
Paramètres.....	400
Utilisation.....	401
Carte Advantech PCI-1710.....	401
Paramètres.....	402
Utilisation.....	403
Valeurs de statut.....	403
Carte Advantech PCI-1711.....	404
Paramètres.....	404
Utilisation.....	405
Valeurs de statut.....	405
Carte Advantech PCI-1715.....	406
Paramètres.....	406
Utilisation.....	407
Valeurs de statut.....	407
Carte Advantech PCI-1720.....	408
Paramètres.....	408
Valeurs de statut.....	408
Carte Advantech PCI-1723.....	409
Paramètres.....	409
Valeurs de statut.....	409
Carte Advantech PCI-1733.....	410
Paramètres.....	410
Valeurs de statut.....	410
Carte Advantech PCI-1734.....	410
Paramètres.....	411
Valeurs de statut.....	411
Carte Advantech PCI-1747U.....	411
Paramètres.....	411
Utilisation.....	412
Valeurs de statut.....	413
Carte Advantech PCI-1751.....	413
Paramètres.....	413
Valeurs de statut.....	414
Carte Advantech PCI-1762.....	414
Paramètres.....	415
Valeurs de statut.....	415
Carte Advantech PCI-1780U.....	415
Paramètres.....	415
Valeurs de statut.....	415
Modules Adam.....	416

Adam 4017, 4018.....	416
Paramètres .....	416
Valeurs de statut.....	416
Adam 4050, 4053, 4055.....	417
Paramètres .....	417
Valeurs de statut.....	418
<b>2 Hilscher .....</b>	<b>418</b>
<b>Carte Hilscher CIF générique.....</b>	<b>418</b>
Installation.....	418
Paramètres .....	419
Etapas du paramétrage.....	419
Valeurs de statut.....	430
Stations .....	431
Hilscher - Station déportées générique.....	431
Paramètres .....	431
Valeurs de statut.....	432
Hilscher - Station déportées 16I 16O.....	432
<b>Carte Hilscher CIF CAN Layer 2.....</b>	<b>432</b>
<b>Carte Hilscher CIF 50-En Modbus/TCP mode IO (serveur).....</b>	<b>432</b>
Installation.....	433
Paramètres .....	434
Valeurs de statut.....	434
<b>Carte Hilscher CIF 50-En Modbus/TCP mode message (client).....</b>	<b>435</b>
Installation.....	435
Paramètres .....	436
Valeurs de statut.....	436
<b>3 Heidenhain.....</b>	<b>437</b>
<b>Carte IK220.....</b>	<b>437</b>
Installation.....	437
Paramètres .....	437
Utilisation .....	437
valeurs de statut.....	438
<b>4 ICP DAS .....</b>	<b>438</b>
<b>Carte ICP DAS PCI-P16R16.....</b>	<b>438</b>
Paramètres .....	439
Valeurs de statut (COPY).....	439
<b>5 Phoenix Contact.....</b>	<b>439</b>
<b>Module Modbus/TCP type FL IL 24 BK-PAC.....</b>	<b>439</b>
Paramètres .....	439
Utilisation .....	440
<b>6 Port I/O PC.....</b>	<b>441</b>
<b>Port I/O PC - 8 entrées TOR.....</b>	<b>441</b>
Paramètres .....	441
Utilisation .....	441
<b>Port I/O PC - 8 sorties TOR.....</b>	<b>442</b>
Paramètres .....	442
Utilisation .....	442
<b>7 Simulation.....</b>	<b>443</b>
<b>Driver de simulation standard.....</b>	<b>443</b>
Paramètre .....	443
Utilisation dans la supervision.....	443
Accès par zone de mémoire partagée.....	444
<b>Driver de simulation Solid Dynamics Motion.....</b>	<b>444</b>
Paramètres .....	444
Utilisation dans la supervision.....	444
Accès par zone de mémoire partagée.....	445
<b>8 Wago .....</b>	<b>445</b>
<b>Module Modbus/TCP type 750-341, 342, 352, 841, 842.....</b>	<b>445</b>
Paramètres .....	445
Utilisation .....	446
Valeurs de statut.....	447



<b>9 Réseau Ethernet standard.....</b>	<b>447</b>
<b>Module Modbus/TCP générique.....</b>	<b>447</b>
Paramètres .....	447
Valeurs de statut.....	449

## **Partie X Questions fréquentes 449**

<b>1 Questions générales.....</b>	<b>450</b>
<b>Pourquoi utiliser le PC comme automate ?.....</b>	<b>450</b>
<b>Optima PLC permet-il de programmer un automate du marché ?.....</b>	<b>450</b>
<b>Optima PLC est-il un outil de simulation ?.....</b>	<b>450</b>
<b>Questions concernant le temps réel.....</b>	<b>450</b>
Optima PLC est-il un système temps-réel ?.....	450
Optima PLC peut-il fonctionner sans le noyau temps-réel ?.....	450
Pourquoi le noyau temps-réel est-il désactivé par défaut ?.....	450
Optima PLC nécessite-t-il du matériel supplémentaire pour fonctionner en temps réel ?.....	451
Je pensais qu'il était possible de faire du temps-réel avec Windows, je ne vois pas ce qu'Optima PLC apporte de plus ?.....	451
Pourquoi ne pas utiliser plutôt un système temps-réel du marché ?.....	451
Optima PLC permet-il de faire du temps réel avec tous les PC ?.....	451
Optima PLC exploite-t-il les ressources des PC multiprocesseurs ?.....	451
Optima PLC permet-il de faire du temps-réel avec tous les matériels interfacés ?.....	452
Quelles performances obtient-on sans le noyau temps-réel ?.....	452
<b>Est-il possible de passer automatiquement d'un langage de la norme à un autre ?.....</b>	<b>452</b>
<b>Est-il possible de modifier une application sans l'arrêter ?.....</b>	<b>452</b>
<b>J'ai déjà travaillé avec Apigraf, puis-je porter facilement un projet Apigraf sous Optima PLC ?.....</b>	<b>452</b>
<b>Où puis-je télécharger la dernière version d'Optima PLC ?.....</b>	<b>453</b>
<b>Optima PLC respecte-t-il la norme IEC 61131-3 ?.....</b>	<b>453</b>
<b>Où puis-je trouver des informations sur la norme IEC 61131-3 ?.....</b>	<b>0</b>
<b>2 Questions techniques.....</b>	<b>453</b>
<b>Matériel .....</b>	<b>453</b>
Quelle configuration est nécessaire pour faire fonctionner Optima PLC ?.....	453
Que peut-il arriver si je n'ai pas assez de mémoire sur le PC ?.....	453
<b>Comment faire.....</b>	<b>453</b>
Comment obtenir des informations sur la licence d'utilisation d'Optima PLC ?.....	453
Comment limiter le risque que l'utilisateur corrompe les données ou le système sous Windows ?.....	453
Comment empêcher l'utilisateur de modifier les paramètres de fonctionnement dans le runtime ?.....	454
Comment utiliser des temporisations dans Optima PLC ?.....	454
Comment faire afficher des messages d'avertissement pour l'utilisateur ?.....	454
Comment forcer un diagramme SFC à son étape initiale ?.....	454
Comment écrire des données dans un fichier pour les relire avec un tableur ?.....	454
Comment saisir des valeurs de paramètres sur mon écran tactile ?.....	454
Comment faire démarrer mon application automatiquement au démarrage du PC ?.....	454
Comment faire communiquer mon application Optima PLC avec une autre application ?.....	454
Comment faire afficher simplement des messages de défaut ?.....	455
Comment configurer la session utilisateur pour exécuter une application Optima PLC.....	455
Configuration des sessions sous Windows XP.....	456
Configuration des sessions sous Windows Vista jusqu'à Windows 10.....	456
<b>Résolution de problèmes.....</b>	<b>457</b>
Je définis des mnémoniques sur des variables standards mais ils n'apparaissent pas.....	457
J'ai une erreur de compilation sur le nom d'une fonction décrite dans la documentation.....	457
Un écran bleu s'affiche ou le PC redémarre pendant l'exécution du runtime.....	457
J'ai un problème avec Optima PLC, qui dois-je contacter ?.....	458
<b>Divers .....</b>	<b>458</b>
Comment puis-je réduire le temps de compilation de mon application ?.....	458

## **Partie XI Annexes 459**

<b>1 Optima APIFTP Servers.....</b>	<b>459</b>
Page principale.....	459
définition des partages.....	459
<b>2 Codes de résultat APIFTP .....</b>	<b>460</b>
opInProgress .....	460

opOK .....	460
opProtocolError .....	460
opConnectionError .....	460
opConnectionAborted .....	461
opConnectionTimeOut .....	461
opNoResource .....	461
opDestinationUnreachable .....	461
opVersionAnswerTimeOut .....	461
opSourceNotFound .....	461
opUnableToOpenSource .....	461
opUnableToReadSource .....	461
opExistFile .....	461
opPathNotFound .....	461
opAccessDenied .....	461
opUnableToCreateDest .....	462
opUnableToWriteDest .....	462
opFileSizeError .....	462
opUnableToRenameDest .....	462
opSourceIsDir .....	462
opSourceIsVolumeID .....	462
opSourceIsWriteProtected .....	462
opUnableToDeleteSource .....	462
opUnableToRenameSource .....	462
opUnableToCreateDir .....	462
opUnableToDeleteDir .....	462
opDirAlreadyExists .....	462
opUnableToGetVersion .....	462
opUnknownParameter .....	462
opUnableToExecFile .....	462
opProtocolVersionError .....	463
opError .....	463

## Index

464

# 1 Introduction

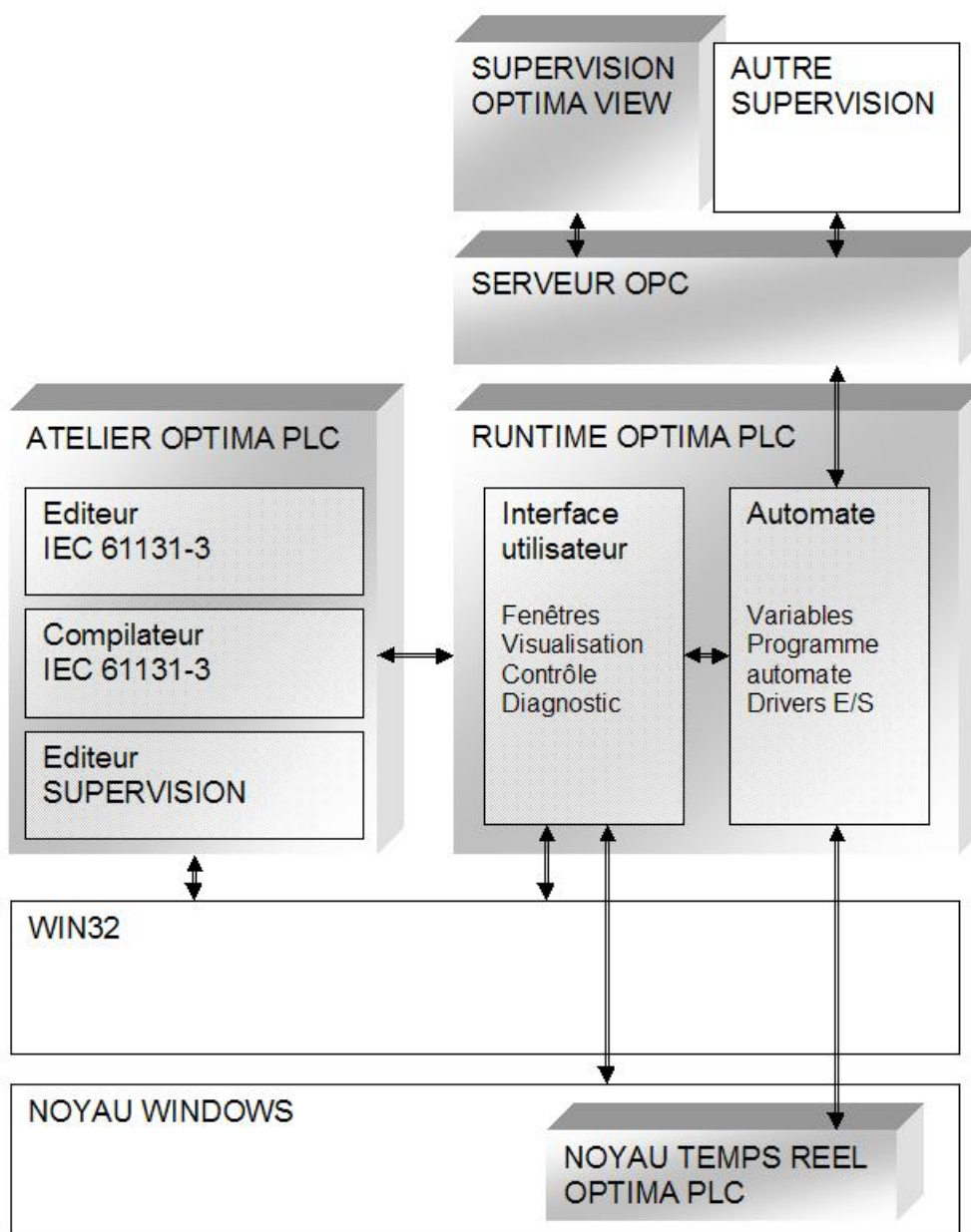
## 1.1 A quoi sert Optima PLC ?

La destination principale d'Optima PLC est d'assumer le rôle d'automate programmable industriel (API, ou PLC en anglais) sur un PC standard. Il permet également de mettre rapidement en oeuvre une supervision graphique associée à cet automate, ainsi que d'autres outils de supervision proposés par Optimalog.

Optima PLC comporte un atelier de développement conforme au standard de programmation d'automates IEC 61131-3, un atelier de supervision, et un système permettant de mettre au point et d'exécuter l'application sous Windows en temps-réel.

## 1.2 Principes généraux

Le schéma suivant montre la structure générale des composants d'Optima PLC.



### 1.2.1 Phases du développement

L'utilisation d'Optima PLC se fait schématiquement en trois phases :

- La programmation de l'application

Pour cela l'atelier de développement dédié permet de programmer et d'organiser des "unités", appelées POU, selon les principes de la norme IEC 61131-3. Ces POU sont regroupés au sein d'un "projet". L'atelier permet également de construire une supervision (interface homme-machine) associée au programme.

- La compilation de l'application

Lorsque l'application est développée, le programmeur utilise la fonction de compilation. Cette opération permet à l'atelier de développement de générer du code exécutable optimisé (il n'y a pas d'interprétation des instructions).

Si des erreurs de compilation sont détectées durant cette phase, elles sont pointées pour permettre au programmeur de les corriger.

- L'exécution de l'application

Enfin, l'application est exécutée grâce à la mise en place du noyau temps-réel dédié d'Optima PLC et du runtime, lui permettant d'optimiser le lancement des tâches définies par le programmeur.

### 1.2.2 Licences d'utilisation, identification du client

Le droit d'utilisation du logiciel est assuré par la présence d'une clé USB insérée dans l'une des prises du PC pendant le fonctionnement de l'atelier et du runtime.

Les deux types de clé ci-dessous sont possibles :



Clé Ithéa



Clé Optimalog

La clé peut contenir une licence pour l'atelier, ou une licence pour le runtime, ou une licence pour les deux. La clé peut également contenir une licence pour les autres logiciels Optimalog.

Une licence d'utilisation comporte entre autre un code client (numérique), unique pour chaque client d'Optimalog. Ce code est reporté dans les objets créés avec Optima PLC, ce qui permet d'en "signer" la création, et de gérer les droits que l'auteur d'un programme souhaite accorder ou non aux autres clients : visualisation, modification, etc.

La clé Optimalog peut en plus contenir des fichiers, en particulier le programme d'installation.

En l'absence de clé de protection, Optima PLC peut être utilisé avec certaines limitations :

- Votre identifiant est "Unknown".
- L'atelier est utilisable pendant un maximum de 30 jours.
- Le temps d'exécution du runtime est limité à une heure, après quoi il doit être relancé.
- Un dialogue d'avertissement est affiché lors du lancement de l'atelier et du runtime.

Attention : lorsque la période d'évaluation de 30 jours est écoulée, Optima PLC ne peut plus être lancé sans clé de protection.

### 1.2.3 Contenu d'un répertoire de projet Optima PLC

Un projet est un dossier (portant le nom du projet) dans lequel tous les fichiers nécessaires à la création et à l'exécution du programme automate sont stockés.

Le répertoire du projet contient les fichiers suivants au minimum :

Main.opl	Librairie contenant les programmes principaux (par défaut)
----------	--

xxx.opl	Librairies définies ou importées par l'utilisateur dans le projet
NomDuProjet.ini	fichier de configuration
NomDuProjet.inb	sauvegarde du fichier de configuration
OptimaPLC.dsk	sauvegarde du bureau de l'environnement de développement

Après compilation et exécution :

OptimaPLC.prg	programme exécutable
Debug.dat	fichier de sortie texte, ouvert automatiquement au démarrage du runtime et sauvé et fermé lors de l'arrêt du runtime
Debug data xxxx.xml	Fichier contenant les données complètes de l'automate sous forme xml. La zone xxxx correspond à la date et heure de création du fichier. Ce fichier peut être créé à la demande de l'utilisateur (menu Diagnostics du runtime) ou bien est généré automatiquement à la sortie de l'application si celle-ci a été compilée avec l'option "informations de mise au point".

Supervision : les fichiers sont placés dans le sous-répertoire PSGW du projet

NomDuProjet.psgw	Page de supervision principale
autres fichiers psgw	Pages de supervision appelées depuis la page principale

Fichiers de description de bureau, dans le sous-répertoire DESK du projet :

Admin.dsk	description du bureau de l'administrateur
xxx.dsk	description du bureau de l'utilisateur xxx, si la gestion des utilisateurs est activée.




Fichiers des variables "retain" de l'application, dans le sous-répertoire RETAIN du projet :





Fichiers .retain	Fichiers sauvegardés toutes les 5 minutes. Le nom correspond à la date et à l'heure de sauvegarde.
------------------	---

## 1.2.4 Contenu d'une librairie de projet Optima PLC

Une librairie de projet est stockée sous la forme d'un fichier .OPL (Optima PLC Library). Un projet Optima PLC peut contenir plusieurs librairies. Certaines librairies standards sont systématiquement chargées et utilisées (system.opl, standard.opl, etc).

Une librairie est constituée de POU's de différents type :

	<b>Tâche</b>	POU de cadencement de l'exécution des blocs de programme
	<b>Variables globales</b>	POU de déclarations de variables disponibles pour l'ensemble des POU's de programmation
	<b>définitions de types</b>	POU de déclarations de type de variables disponibles pour l'ensemble des POU's de programmation

	<b>bloc de programme</b>	POU de programmation principal
	<b>bloc de fonction</b>	Sous-programme, avec ou sans valeur de retour, qui mémorise l'état de ses variables internes entre 2 appels, ainsi pour un même jeu de paramètres d'entrée le résultat renvoyé par ce bloc peut être différent (exemple du PID : pour une même consigne la sortie de ce bloc de régulation évolue jusqu'à ce qu'elle atteigne la consigne)
	<b>fonction</b>	sous-programme dont la valeur de retour reste identique pour un même jeu de paramètres d'entrée (pas de mémorisation des variables internes)
	<b>sous-groupe</b>	POU de structuration de la librairie.

Un projet contient une ou plusieurs librairies contenant chacune une ou plusieurs tâches, faisant appel à un ou plusieurs programmes qui peuvent eux-mêmes faire appel à des blocs de fonctions et à des fonctions.

### 1.2.5 Compilation et génération de code

Pour être en mesure d'exécuter l'application développée, Optima PLC procède en deux étapes préliminaires :

- Vérification du code et contrôle de la cohérence des POU : à ce stade, l'application est entièrement contrôlée par Optima PLC.
- Génération du code exécutable : les POU sont compilés en code directement exécutable optimisé. Le résultat de cette compilation est le fichier OptimaPLC.prg placé dans le répertoire de l'application.

### 1.2.6 Modification du code en ligne

Optima PLC permet d'effectuer certaines modifications de l'application et de les mettre en service "en ligne", c'est à dire sans interrompre le fonctionnement de l'application.

Pour cela, Optima PLC procède de la manière suivante :

- Vérification du code et contrôle de la cohérence des POU (comme pour la génération de code normale).
- Vérification des modifications effectuées depuis la mise en route de l'application. A ce stade, certaines modifications détectées peuvent interrompre la mise en ligne du nouveau code.
- Génération du code exécutable : le code généré est directement exécutable par l'unité centrale et optimisé.
- Chargement et mise en service du code exécutable modifié, avec interruption minimale de l'application (moins d'une milliseconde).

### 1.2.7 Gestion des droits

L'auteur d'un POU "signé" sa création avec le code contenu dans la clé de protection. Lorsqu'un autre client (disposant d'une clé différente, avec un autre code client) souhaite visualiser ou modifier ce POU, ses droits peuvent être limités aux seules actions autorisées par l'auteur.

De cette manière, un intégrateur peut par exemple, s'il le souhaite, protéger sa propriété intellectuelle contre la consultation, ou empêcher toute modification à son code pendant la période de garantie de son installation.

Voir ["Page droits"](#)  dans les options de POU.

## 1.3 Installation d'Optima PLC


Optima PLC doit être installé sur un PC dont le système d'exploitation est Windows XP ou Vista.

Si vous disposez d'une clé Ithéa et s'il s'agit d'une première installation, le driver de la clé de protection USB Ithéa sera installé simultanément. **Important : la clé Ithéa ne doit être insérée dans une prise USB que lorsque la première installation est terminée.**



Clé Ithéa

Vous pouvez effectuer l'installation à partir de différents supports :

<b>Clé Optimalog</b> 	La clé sert à la fois de clé de protection et de support des fichiers d'installation. Insérer la clé dans un connecteur USB du PC. Windows procède automatiquement à l'installation de la clé, mais ne lance pas le logiciel d'installation d'Optima PLC. Ouvrez le volume correspondant à la clé avec l'explorateur de fichiers puis lancez le programme d'installation qui s'y trouve.
<b>CD ROM</b>	Insérez le CD ROM dans le lecteur. Le démarrage automatique vous propose l'installation d'Optima PLC ainsi que d'autres logiciels tiers parties.
<b>Programme d'installation téléchargé</b>	Exécutez le programme et suivez les indications décrites ci-dessous.

Si vous ne possédez pas de clé de protection, Optima PLC fonctionnera en mode évaluation.

## Importance de la session Windows

Attention, il est vivement conseillé d'effectuer l'installation depuis une session Windows dans laquelle vous disposez des droits d'administrateur. L'ensemble de l'installation pourra être effectué sans problème, et les répertoires d'exemples et de projets seront placés dans :  
"Documents partagés \Optima PLC".

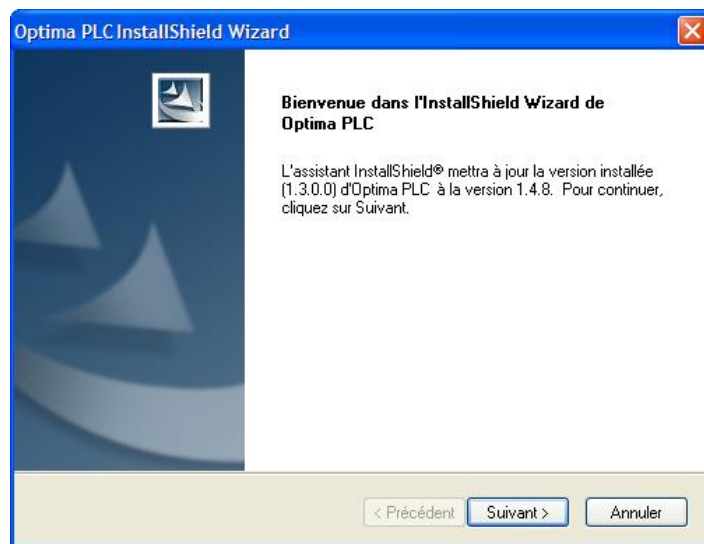
De plus, l'installation sera ensuite accessible à l'ensemble des utilisateurs.

Par contre, si l'installation est effectuée depuis une session ne disposant pas des droits d'administrateur :

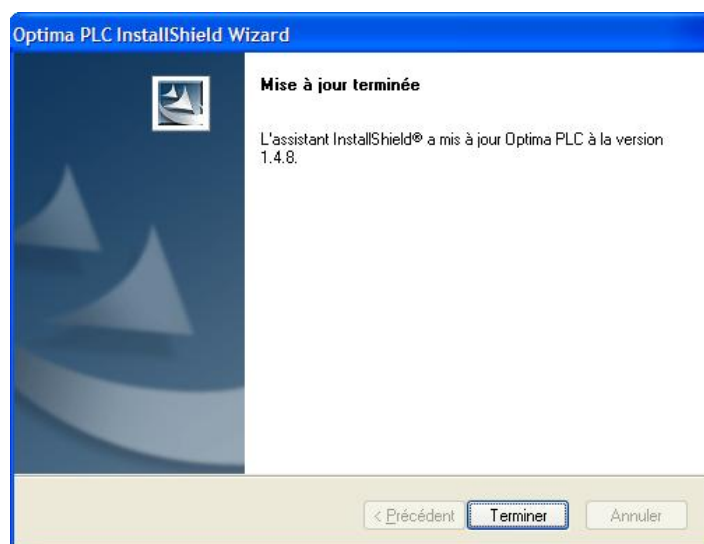
- certains fichiers ne pourront pas être installés dans les répertoires de Windows (serveur OPC par exemple).
- l'installation ne sera accessible qu'à l'utilisateur en cours
- les répertoires d'exemples et de projet seront placés dans :  
"Mes documents\Optima PLC".

## Démarrage de l'installation

Lancer le programme d'installation fourni.



Une fois l'installation terminée, un redémarrage est proposé, ou bien un écran comme le suivant est affiché s'il ne s'agit pas de la première installation :



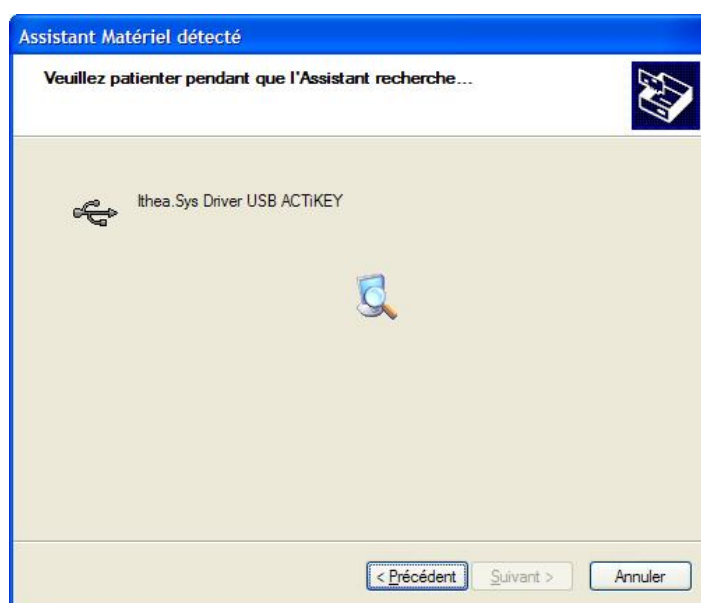
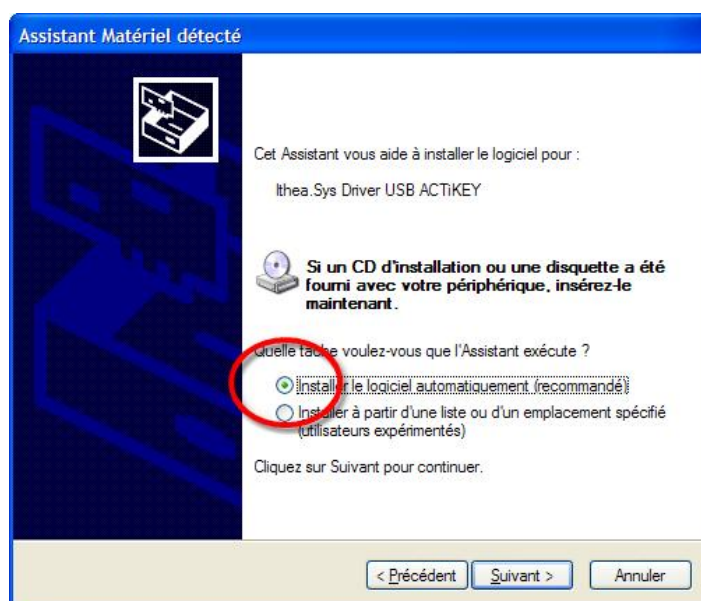
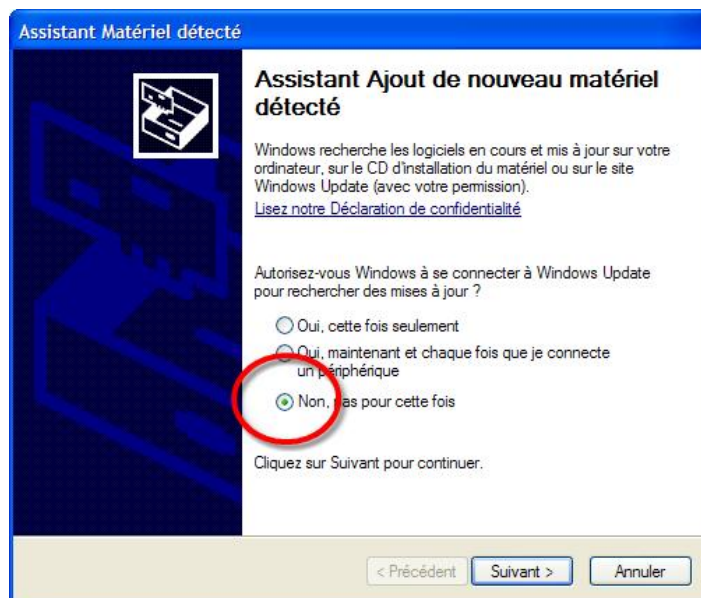
Lors de la première installation, un redémarrage est nécessaire afin de rendre opérationnel le serveur OPC intégré à Optima PLC. Si l'installation n'est pas la première effectuée, il est possible de lancer immédiatement le logiciel depuis le menu "Démarrer".

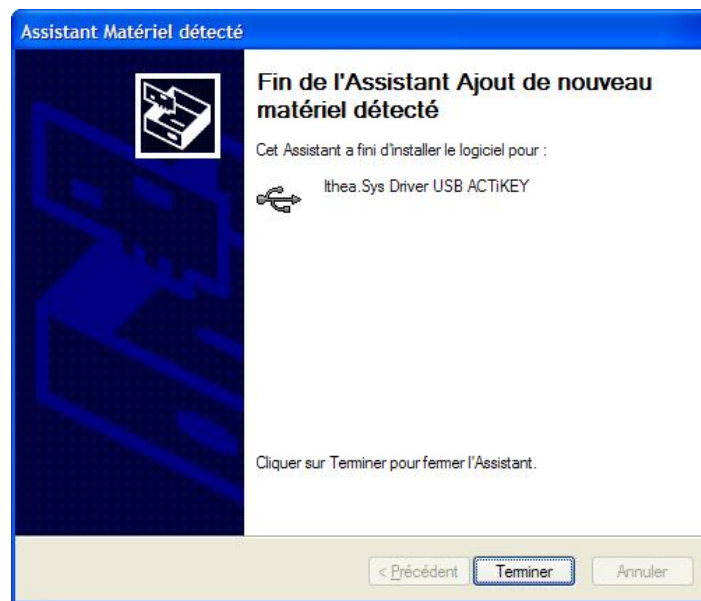
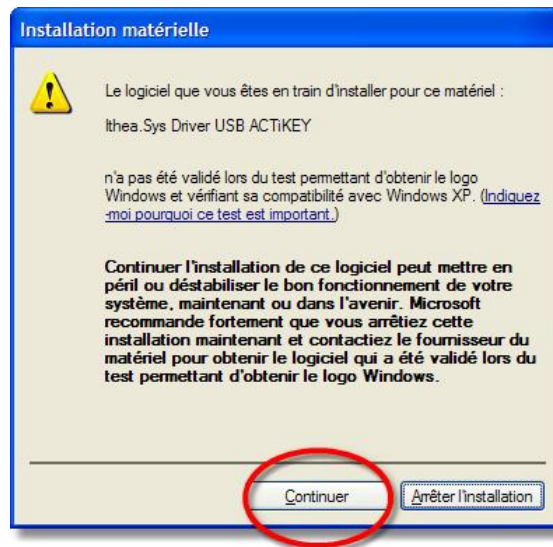
## Installation de la clé lthea

Si vous ne disposez pas de clé lthea, l'installation est maintenant terminée. Si par contre vous possédez ce type de clé, et s'il s'agit d'une première installation, Windows doit finaliser l'installation de la clé USB de protection lthea : après le redémarrage du PC, insérer maintenant cette clé dans l'une des prises USB.

Les fenêtres suivantes s'affichent, cocher à chaque fois la case indiquée puis le bouton "suivant" :







L'installation est maintenant complètement terminée, le logiciel peut être lancé depuis le menu "Démarrer".

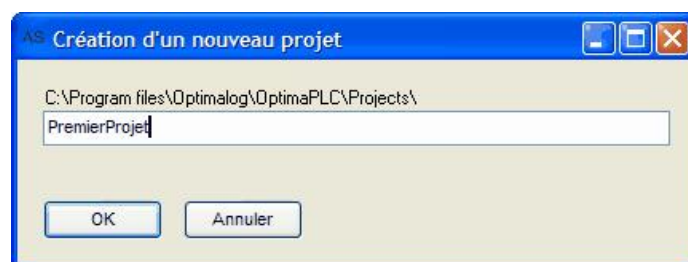
## 1.4 Première utilisation : exemple de création d'un projet

Pour démarrer l'atelier de développement Optima PLC, double-cliquer sur l'icône présent sur le bureau.

Dès le démarrage, une boîte de dialogue permet la sélection de projet.

Dans la fenêtre arborescente qui apparaît, sélectionner le dossier dans lequel le projet doit être créé (normalement "Projects") puis cliquer sur le bouton "Nouveau projet".

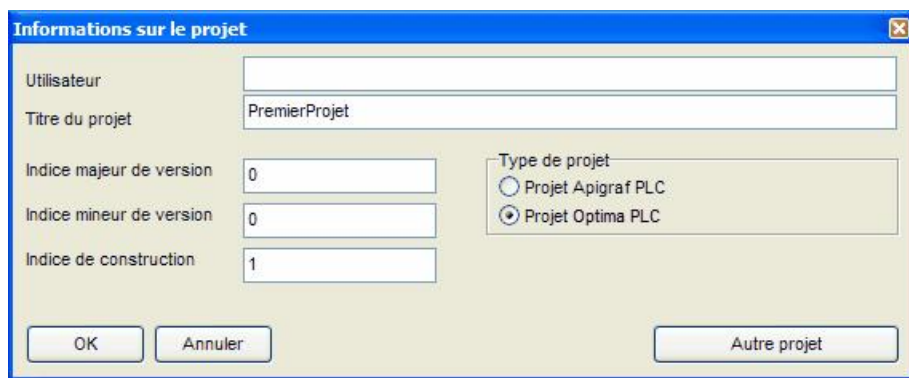
La fenêtre qui apparaît alors permet de donner un nom à votre projet :



Dans l'exemple ci-dessus, le projet créé est PremierProjet.

Après avoir confirmé la création du projet, une nouvelle fenêtre permet de saisir les informations relatives à ce projet

à savoir :

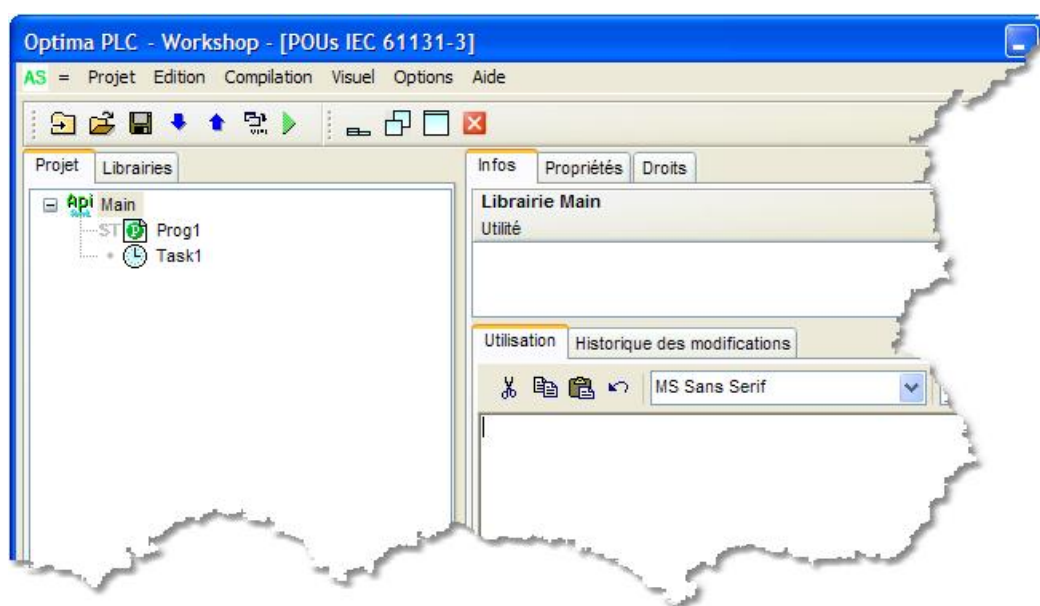


- Le nom de l'utilisateur
- Le titre du projet
- Le type de projet (choix entre un projet Apigraf PLC et un projet Optima PLC)
- Le numéro de version (composé de 3 indices, V0.0.1 à la création)

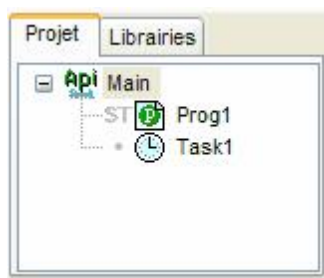
Le projet est maintenant créé, la programmation est l'étape suivante.

L'environnement de développement se compose :

- Du menu de l'éditeur
- D'une barre d'icônes pour les fonctions les plus courantes (exemple : Tout sauver, Exécuter...)
- D'un panneau à onglets permettant d'accéder l'organisation du programme et aux bibliothèques d'objet de programmation standard d'Optima PLC
- D'un second panneau à onglets pour programmer les différents POU du projet (le nombre d'onglets varient en fonction du bloc sélectionné dans le panneau des bibliothèques)



Lors de la création du projet, des éléments sont automatiquement insérés dans l'onglet "Projet" :

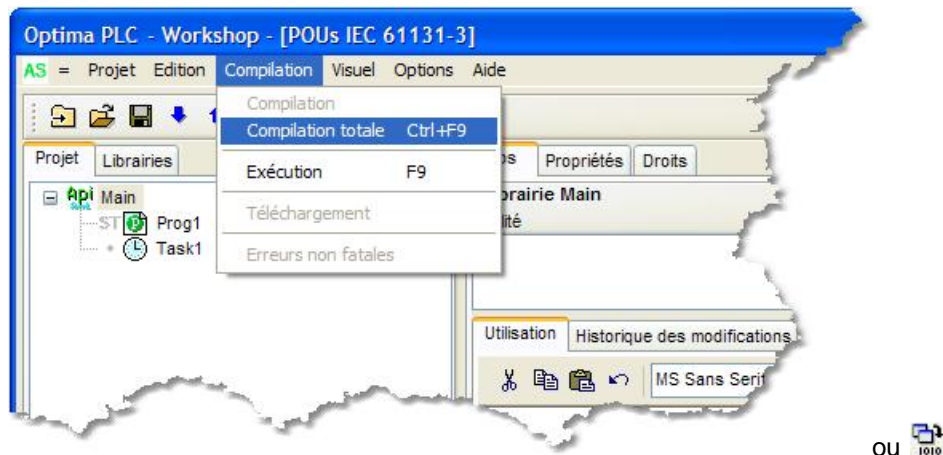


Un dossier "main" correspond à la librairie principale du projet.

Dans cette librairie, deux POU sont créés. Ils composent le programme minimum (c'est-à-dire que le programme peut déjà être compilé et exécuté) :

- Une tâche : ce POU permet de définir les blocs de programmes qui doivent être exécutés, leur priorité ainsi que la période ou l'événement qui cadence l'exécution
- Un bloc de programme : pour écrire le programme (par défaut, programmation en langage ST)

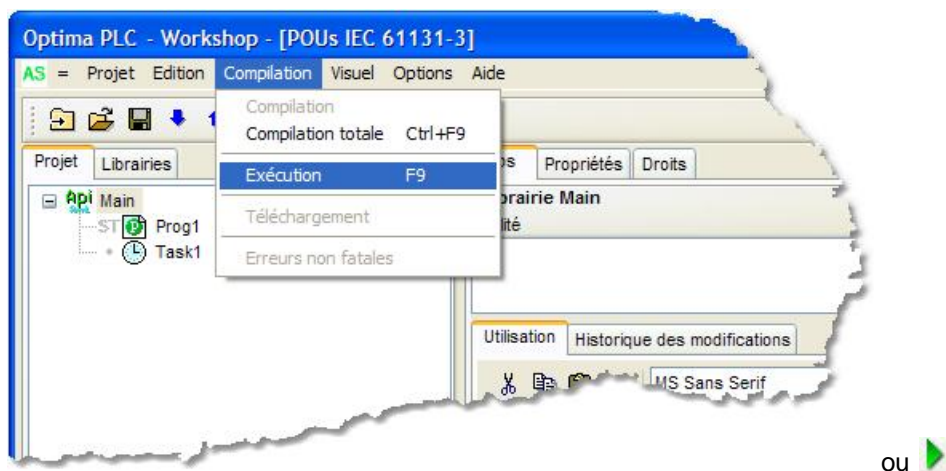
Dès à présent le programme peut être compilé et exécuté (même si pour le moment ce programme ne fait rien). Pour compiler le programme, sélectionner le menu " Compilation → Compilation totale " ou cliquer sur l'icône appropriée.



ou

Un message s'affiche : " Veuillez patienter, compilation en cours. ". Lorsque la compilation est terminée, un nouveau message indique si celle-ci a été réalisée avec ou sans erreur. Pour le moment aucune ligne de programme n'ayant été écrite dans le projet, la compilation se termine sans erreur.

Pour exécuter le programme, sélectionner le menu " Compilation → Exécution " ou cliquer sur l'icône appropriée.



ou

L'application est alors démarrée, l'environnement du Runtime apparaît :



Il se compose :



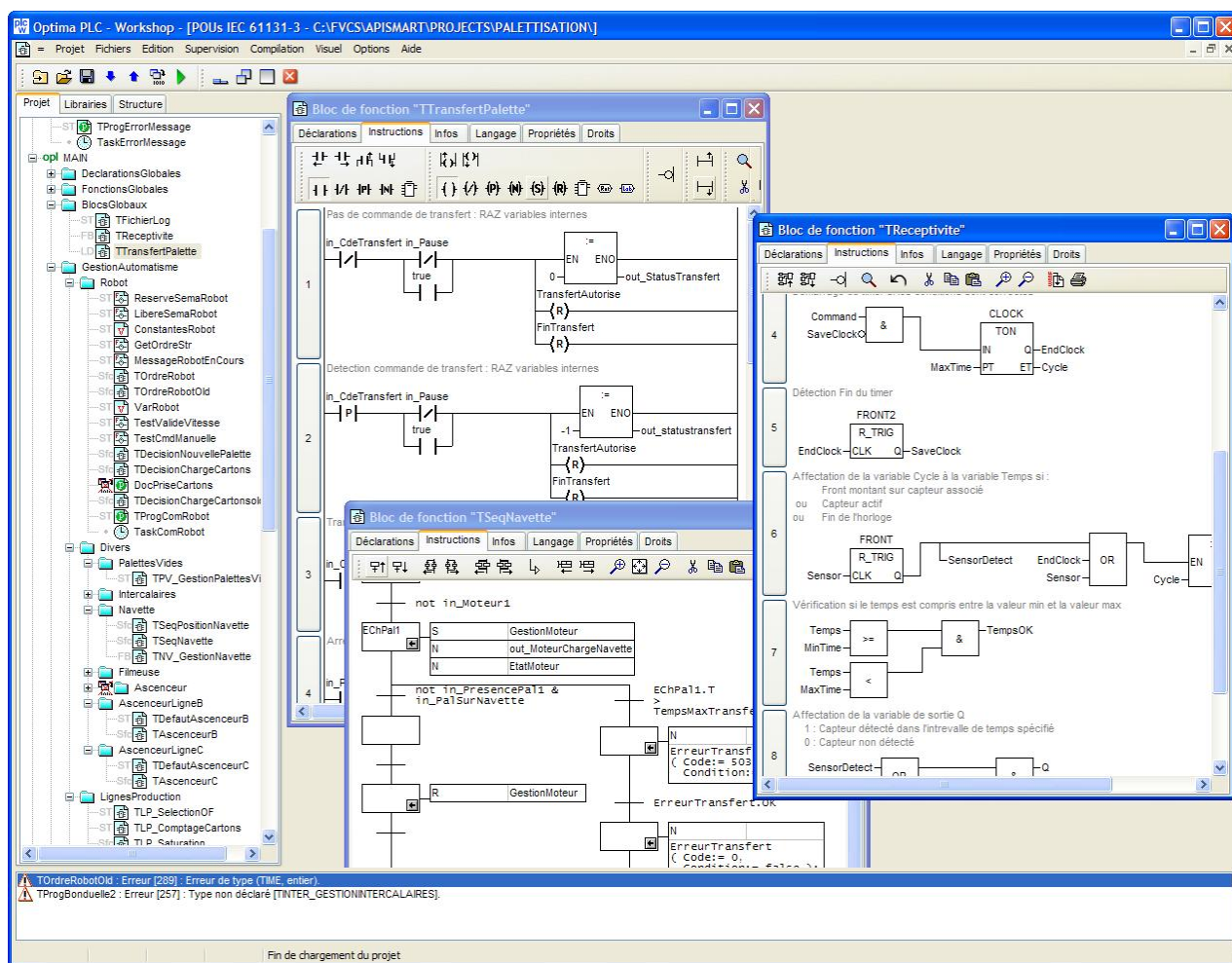
- D'un menu
- D'une barre d'état (en bas)
- Le centre de la fenêtre est dédiée aux fenêtres du runtime (exemple : visualisation des variables)

Pour quitter le runtime, sélectionner le menu " Contrôle → Quitter " et confirmer la demande d'arrêt de l'application.

L'environnement d'Optima PLC vous est maintenant plus familier. Les chapitres suivants fournissent de plus amples détails sur la programmation et sur l'utilisation des environnements de développement et du runtime.

## 2 Atelier de programmation

Ce chapitre détaille l'utilisation de l'atelier de programmation Optima PLC, qui permet de définir les POU's d'une application selon les préconisations de la norme IEC 61131-3.



### 2.1 Description des menus

#### 2.1.1 Menu =

##### 2.1.1.1 Langue logiciel

Cette option permet de sélectionner le langage utilisé pour les affichages dans l'atelier.

##### 2.1.1.2 Langue programmation

Le changement de langage de programmation n'est actuellement disponible que lorsque le projet est un projet de type Apigraf.

##### 2.1.1.3 Option Licence

Cette option permet d'afficher les informations concernant la licence d'utilisation du logiciel.

Optima PLC - Licences d'utilisation

Optima PLC

Localisation

☐ Aucune clé

☒ Clé USB locale

☐ Clé USB distante

☐ PC

No de série 105277

Type de licence Gold

Numéro de licence 612002

Début/fin de validité 09/12/2006

Nom utilisateur Dupont Automatismes

Code utilisateur 113

Fin de maintenance logicielle

Fin d'assistance technique

☒ Atelier logiciel

☒ Runtime

☐ Mono-application

☒ OPTIMA View

Nb Max I/O TOR

Nb Max I/O Ana

Actualiser

Fermer

Modification Licences

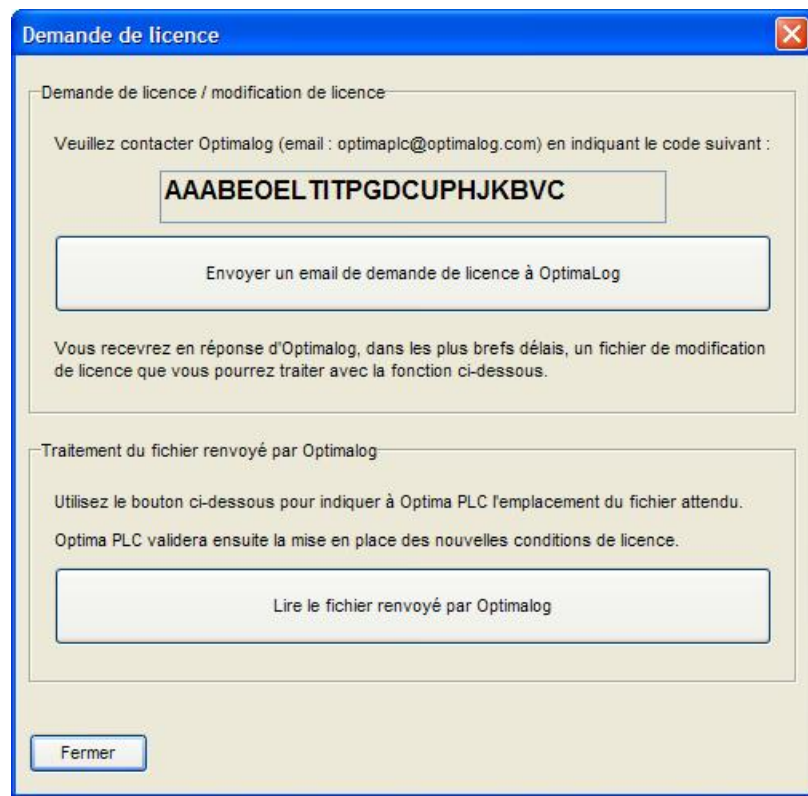
Lorsqu'une nouvelle clé de protection est insérée, l'actualisation n'est pas instantanée. Pour la forcer, cliquer sur le bouton "Actualiser" de la fenêtre, les informations de licence sont alors lues depuis la clé.

Les informations affichées indiquent les caractéristiques générales de la licence (droits d'utilisation de l'atelier, du runtime...) et les dates d'échéances s'il y en a. Si aucune date n'est mentionnée dans un champ, il n'y a pas de limite de temps.

La date de fin de maintenance logicielle indique que la licence ne permettra pas de faire fonctionner une version du logiciel dont la date de création se situe au delà.

Pour prolonger les limites de validité, merci de contacter votre distributeur ou OptimaLog.

Le bouton "Modification Licences" permet d'accéder à un nouvel écran :

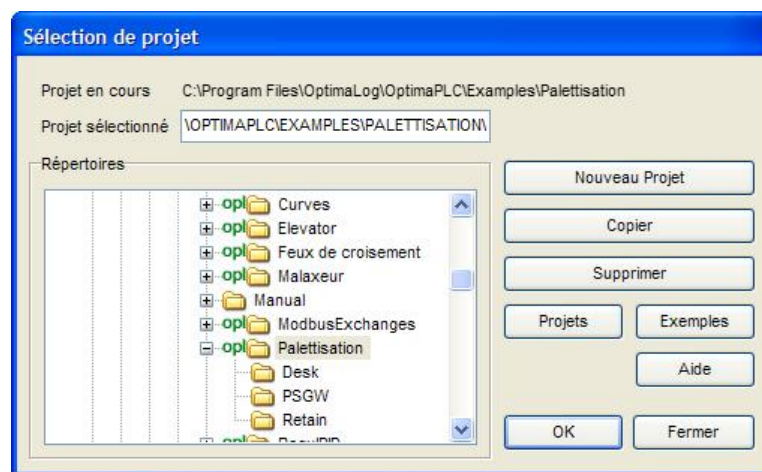


Le code mentionné doit être communiqué à Optimalog, en même temps que la demande de modification de licence (nouvelle licence, upgrade, extension de durée, etc). Le code peut être copié dans le presse-papier, ou envoyé par email en cliquant sur le bouton 'Envoyer un email...".

Lorsque la demande est acceptée et traitée par Optimalog, un fichier est envoyé pour valider la modification. Cliquer sur le bouton "Lire le fichier renvoyé par Optimalog" pour sélectionner ce fichier et pour effectuer la modification.

## 2.1.2 Menu Projet

### 2.1.2.1 Sélection projet



La boîte de dialogue "Sélection de projet" est constituée des éléments suivants :

- L'indicateur du projet en cours (au démarrage d'Optima PLC, ce champ est vide)
- Une boîte de saisie pour sélectionner le projet (nom tapé par l'utilisateur ou mis à jour lors d'une sélection dans l'arborescence des répertoires)
- Un panneau "Répertoires" pour sélectionner un projet existant dans l'arborescence du disque ou un

répertoire pour créer un nouveau projet. Les répertoires constituant des projets Optima PLC valides sont repérés par un symbole "OPL".

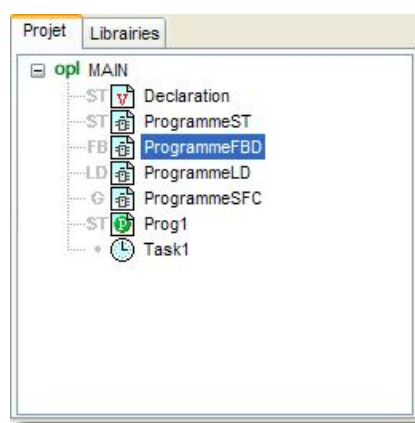
- Un bouton "Nouveau projet" pour créer un projet : le répertoire dans lequel il doit être créé doit avoir été sélectionné avant de cliquer sur ce bouton, le nom du projet doit être composé de caractères alphanumériques.
- Un bouton "Copier" pour copier un projet existant sous un autre nom, le projet source n'est pas supprimé à la fin de la copie (remarque : les bibliothèques copiées ne sont pas renommées)
- Un bouton "Supprimer" pour supprimer le projet sélectionné, deux confirmations sont nécessaires avant que le projet ne soit réellement supprimé
- Deux boutons "Projets" et "Exemples" pour sélectionner les répertoires de projet et d'exemples créés par défaut dans le répertoire Optima PLC
- Un bouton "OK" pour valider le choix du projet (un double-clic avec la souris produit une action équivalente)
- Un bouton "Fermer" pour fermer la fenêtre et revenir au projet en cours

Lorsqu'un projet est sélectionné, une nouvelle boîte de dialogue apparaît :

Cette fenêtre permet de définir (et visualiser) les informations relatives au projet, telles que :

- Le nom de l'utilisateur
- Le titre du projet
- Le type de projet (choix entre un projet Apigraf PLC et un projet Optima PLC)
- Le numéro de version (composé de 3 indices majeur, mineur et de construction, V0.0.1 à la création)

### 2.1.2.2 Liste des POU



L'éditeur se compose alors d'un menu, d'une barre d'icônes, d'un panneau à onglet pour visualiser sous une forme arborescente le contenu des bibliothèques (à gauche), d'un second panneau à onglet (à droite) pour afficher et modifier les données des POU des bibliothèques et d'une barre d'affichage (en dessous) pour les résultats de compilation.

### 2.1.2.3 Editeur Optima PLC

L'éditeur apparaît comme un simple éditeur de fichiers texte, la barre de menu comprend alors 2 options supplémentaires :



- Fichier : pour créer, ouvrir, fermer ou sauver des fichiers texte
- Recherche : pour chercher ou remplacer des mots dans le fichier ouvert

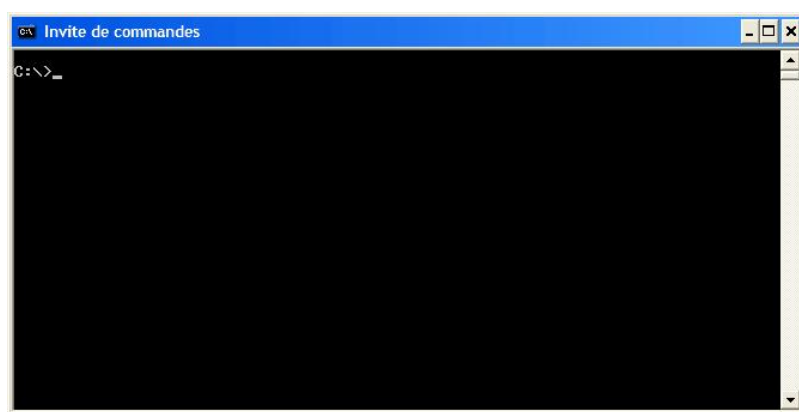
#### 2.1.2.4 Gestionnaire de fichiers

Ouverture de l'explorateur de fichiers de Windows. Le focus est placé sur le répertoire du projet courant, ou bien sur le répertoire des fichiers de log.

Si le projet est un projet Apigraf, l'explorateur de fichiers Apigraf est affiché.

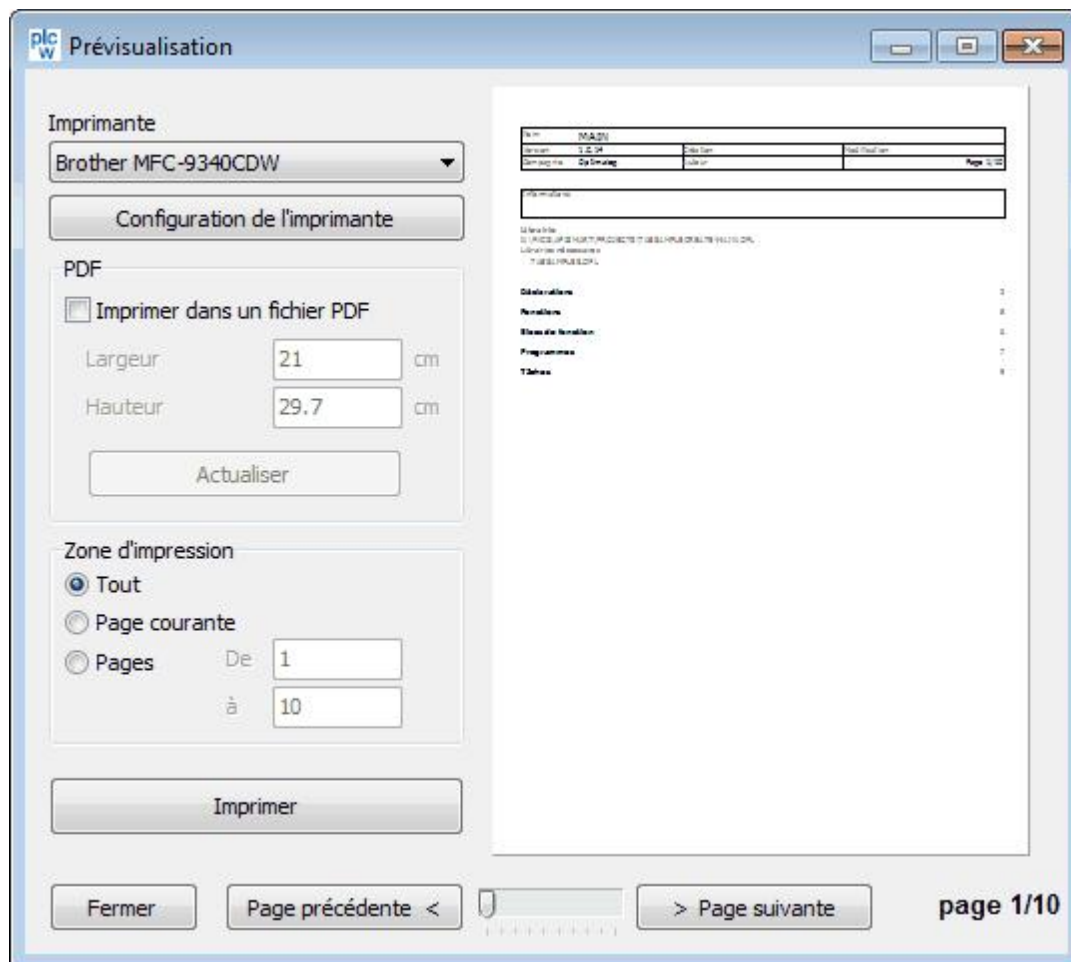
#### 2.1.2.5 Fenêtre système

Une fenêtre système peut être ouverte directement depuis Optima PLC sans passer par les menus de Windows. Le répertoire du projet en cours est automatiquement sélectionné comme répertoire courant.



#### 2.1.2.6 Imprimer

La boîte de dialogue "Imprimer" permet d'imprimer les éléments spécifiques du projet. Il est également possible de créer un fichier au format pdf.



### 2.1.2.7 Quitter

L'atelier de développement Optima PLC ne se ferme qu'après avoir demandé à l'utilisateur s'il souhaite sauvegarder la ou les bibliothèques en cours de modification.

## 2.1.3 Menu Fichiers

### 2.1.3.1 Nouvelle librairie

Cette option crée une nouvelle librairie Optima PLC, placée après toutes celles déjà créées. Le nom de la librairie doit être entré à ce moment. L'atelier crée alors un fichier .opl vide portant ce nom.

### 2.1.3.2 Ouvrir une librairie

Option permettant d'ouvrir une librairie déjà existante sur le disque et de l'intégrer dans le projet.

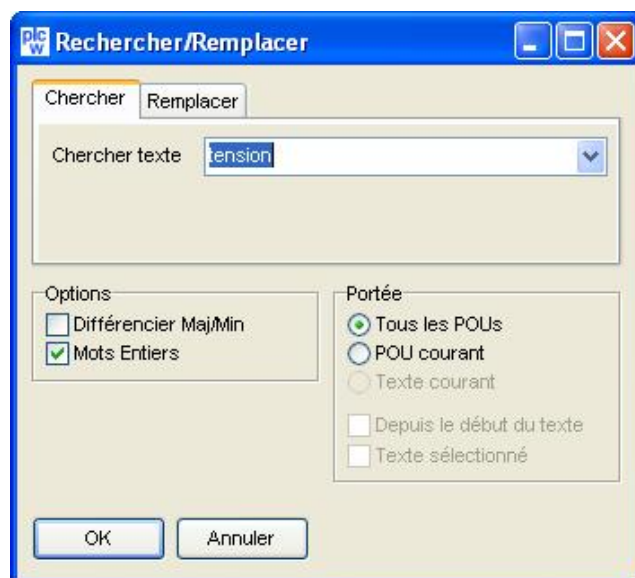
### 2.1.3.3 Sauver toutes les bibliothèques

Sauvegarde toutes les modifications effectuées dans toutes les bibliothèques ouvertes.

## 2.1.4 Menu Edition

### 2.1.4.1 Chercher

Cette option permet de rechercher un texte dans le projet.



### 2.1.4.2 Remplacer

Cette option permet de remplacer un texte par un autre dans le projet.



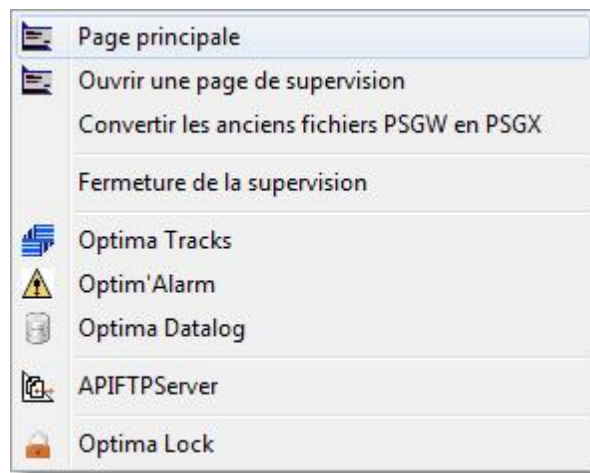
### 2.1.4.3 Occurrence suivante

Lorsqu'un texte est recherché ou à remplacer, cette option permet de sélectionner l'occurrence suivante du mot.

La touche de fonction F3 permet également d'effectuer cette opération.

### 2.1.5 Menu Supervision

Ce menu permet l'accès à l'atelier de supervision intégré à Optima PLC.



- Page principale

Cette option appelle l'atelier de supervision en affichant la page principale. Si cette page n'existe pas encore, elle est créée.

La page principale correspond au fichier situé dans le sous-répertoire PSGW, et dont le nom est celui du projet. L'extension des pages de supervision est .PSGW (page de supervision graphique windows).

- Ouvrir une page de supervision

Cette option permet de choisir une page de supervision à éditer.

- Convertir les anciens fichiers PSGW en PSGX

Conversion de tous les fichiers psgw du répertoire PSGW vers le nouveau format PSGX, correspondant au format utilisé par les nouveaux outils de supervision.

- Fermeture de la supervision

Cette option provoque la fermeture de l'atelier de supervision.

- Optima Tracks, Optim'Alarm, Optima Datalog

Lancement des outils complémentaires de supervision.

- APIFTPServer

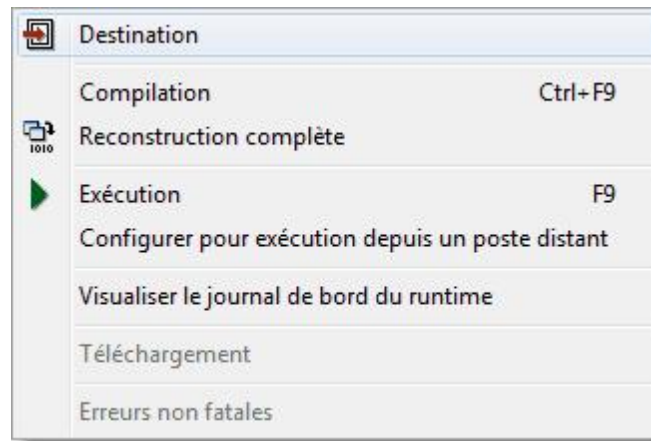
Lancement du programme de communication de fichiers utilisé en commun avec Apigraf

- Optima Lock

Outil de verrouillage du poste

### 2.1.6 Menu Compilation

Ce menu comporte les différentes options permettant de définir la destination de l'application, de la compiler et de l'exécuter.

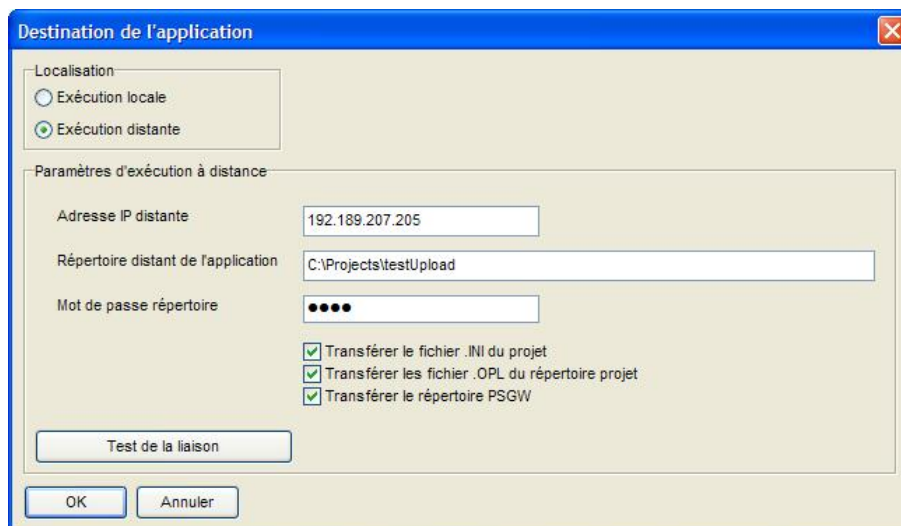


### 2.1.6.1 Destination

Cette option permet de définir l'emplacement final de l'application. Cet emplacement peut être :

- local : l'application est générée et exécutée sur le même PC que l'atelier de développement.
- distant : l'application est générée localement, puis transférée à l'adresse définie.

Dans les deux modes, l'application peut être lancée, stoppée et modifiée en ligne depuis l'atelier.



#### 2.1.6.1.1 Conditions de fonctionnement pour une destination distante

Pour que l'exécution à distance puisse fonctionner correctement, les conditions suivantes doivent être remplies.

- Optima PLC doit être installé sur le poste distant, avec la même version que sur le poste local.
- Optima APIFTP Servers doit être en fonctionnement sur le poste distant. Ce programme est normalement installé en même temps qu'Optima PLC.
- La configuration des partages dans Optima APIFTP Servers doit permettre un accès complet au répertoire des exécutables d'Optima PLC, ainsi qu'au répertoire destiné à contenir l'application. De plus, les mots de passe doivent être identiques pour ces deux répertoires.

Dans le cas d'une installation standard, les répertoires sont définis de la manière suivante :

C:\Program Files\OptimaLog\Optima PLC\bin	répertoire des exécutables
C:\Documents and Settings\All Users\Documents\Optima PLC\Projects	répertoire d'application

Si l'installation a été faite en session non administrateur, les répertoires sont :

C:\Documents and Settings\[user]\Application Data \OptimaLog\Optima PLC\bin	répertoire des exécutables
C:\Documents and Settings\[user]\Mes documents \Optima PLC\Projects	répertoire d'application

- Si le PC distant est protégé par un firewall, il doit autoriser les ports TCP 10260 (APIFTP) et 10580 (communication entre atelier et runtime).

La fenêtre de paramétrage comporte un bouton "test" permettant de vérifier si l'exécution à distance est possible. Ce bouton ne teste pas la liaison nécessaire à la modification en ligne.

#### 2.1.6.1.2 Paramètres pour une destination distante

- Adresse IP distante

Adresse du PC destination. Cette adresse peut être locale ou accessible via Internet.

- Répertoire distant de l'application

Chemin d'accès au répertoire dans lequel les fichiers de l'application doivent être copiés.

- Mot de passe répertoire

Mot de passe défini dans les partages sur le PC distant avec Optima APIFTP Servers.

- Transferts

Définition des fichiers transférés lors du lancement de l'application distante. Ces fichiers ne sont pas transférés lors d'une modification en ligne.

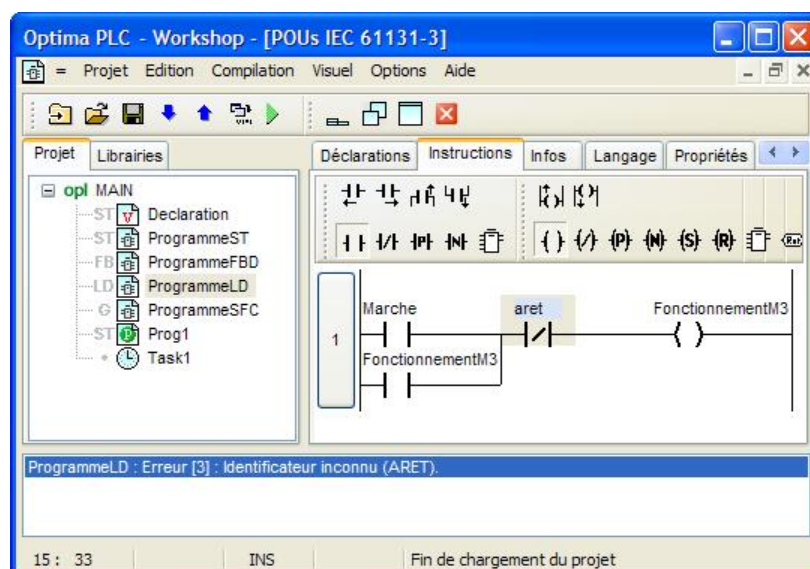
Attention : le transfert du fichier .ini est nécessaire lors de la première exécution, mais ensuite écrase les informations écrites localement par l'application.

#### 2.1.6.2 Compilation complète

La compilation complète recompile tous les POU et fichiers du projet. Si une erreur de compilation survient, la compilation est stoppée sur la position correspondante.

En cas d'erreur, les indications nécessaires à la mise au point s'affichent dans la barre d'affichage en bas de l'éditeur.

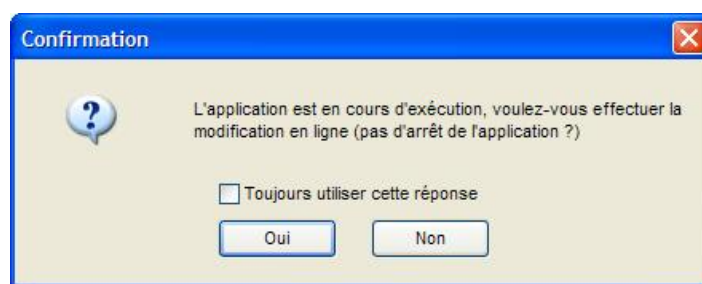
Dans l'exemple ci-dessous, la variable "Arret" a été frappée avec une erreur (un r manque). L'erreur est signalée sur la ligne du bas. Un double clic sur cette ligne renvoie directement à la position d'erreur.



### 2.1.6.3 Compilation et modification en ligne

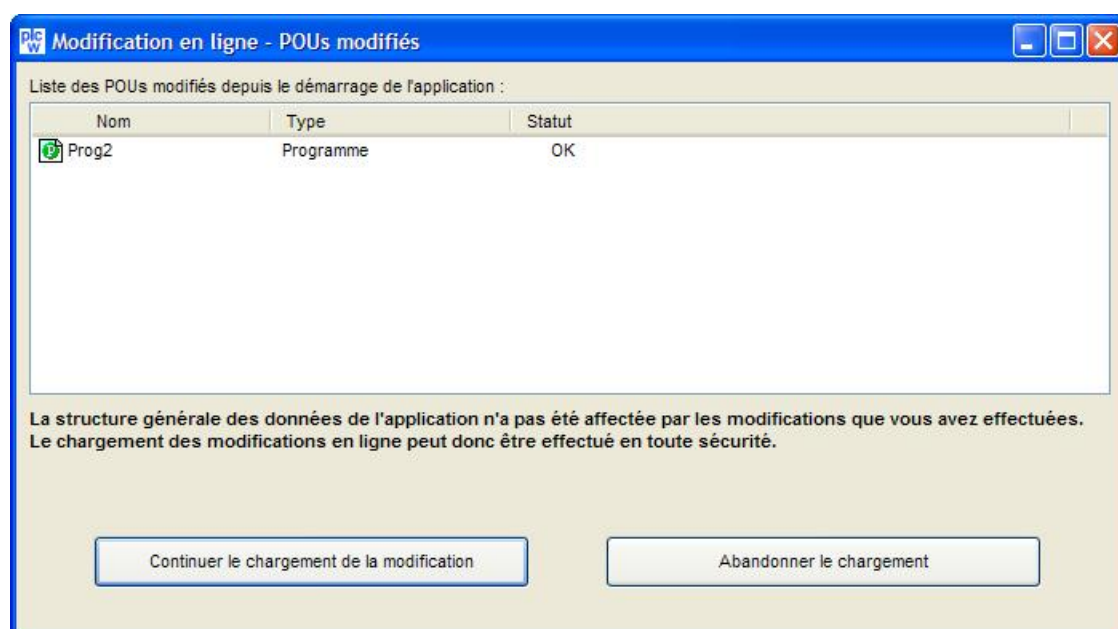
Cette option permet de compiler l'application alors que celle-ci est déjà en cours de fonctionnement.

Une première confirmation est demandée à ce stade :



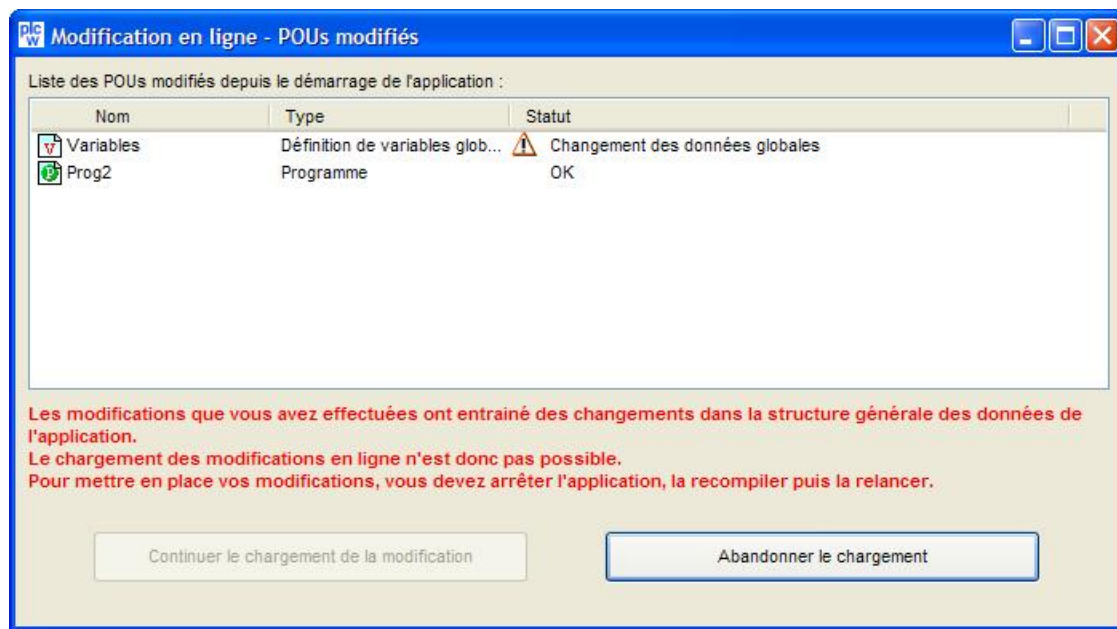
Si cette confirmation est donnée, Optima PLC vérifie la cohérence des POU's ainsi que les modifications apportées à l'application depuis son lancement.

Si tout est correct, une fenêtre comme celle-ci est affichée :



Dans ce cas, cliquer sur "continuer le chargement de la modification" provoque la génération du code exécutable, et transmet la demande de chargement de ce nouveau code à l'application. C'est celle-ci qui prend en charge la mise en place du code modifié, après une dernière modification de l'utilisateur.

Si par contre des modifications non autorisées sont détectées, une fenêtre telle que celle-ci est affichée :



Dans ce cas le chargement en ligne des modifications ne peut pas être effectué.

Une description complète des modifications autorisées ou non est disponible au chapitre Modifications "en ligne".

#### 2.1.6.4 Exécution

Dans le cas d'un projet Optima PLC, si une modification a été effectuée dans une partie du projet, l'application est tout d'abord recompilée, puis exécutée localement.

L'exécution en local n'est pas possible dans le cas d'un projet Apigraf.

### 2.1.7 Menu Visuel

#### 2.1.7.1 Document suivant

Cette option n'est disponible qu'avec l'éditeur de fichiers texte d'Optima PLC (et non la liste des POU).

Lorsque plusieurs fichiers sont ouverts, il est possible de mettre en avant-plan le document suivant.

#### 2.1.7.2 Document précédent

Cette option n'est disponible qu'avec l'éditeur de fichiers d'Optima PLC (et non la liste des POU).

Lorsque plusieurs fichiers sont ouverts, il est possible de mettre en avant-plan le document précédent.

#### 2.1.7.3 Mode vidéo

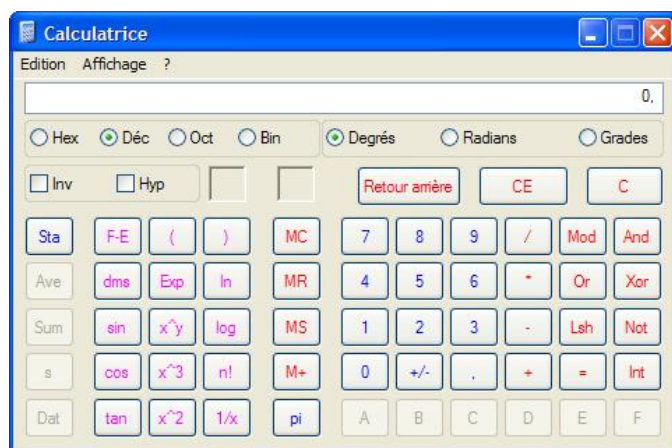
Il est possible d'accéder rapidement aux propriétés d'affichage de Windows depuis Optima PLC.





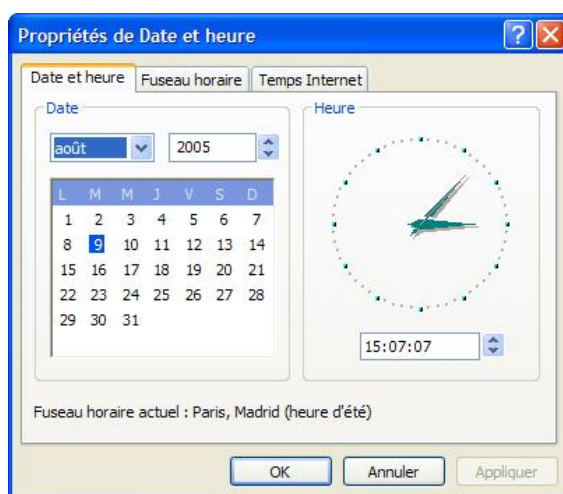
#### 2.1.7.4 Calculatrice

Il est possible d'accéder rapidement à la calculatrice de Windows depuis Optima PLC.



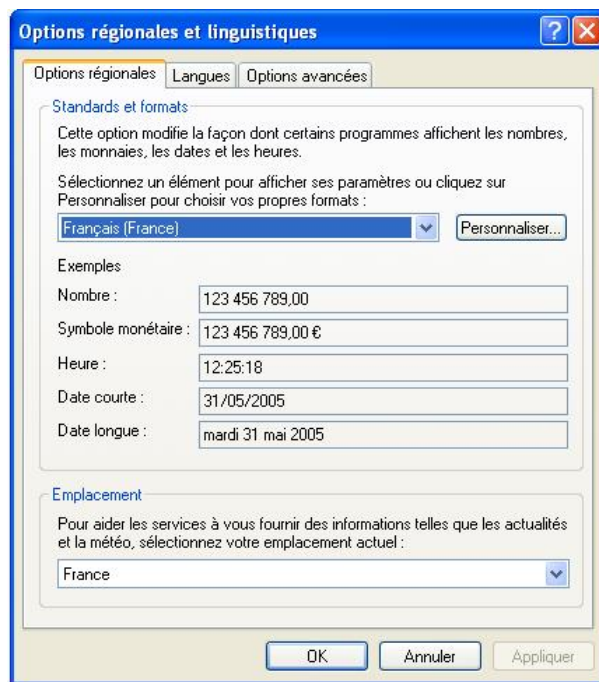
#### 2.1.7.5 Calendrier/Horloge

Il est possible d'accéder rapidement aux propriétés date et heure de Windows depuis Optima PLC.



#### 2.1.7.6 Format Date/Heure

Il est possible d'accéder rapidement aux options régionales et linguistiques de Windows depuis Optima PLC.



### 2.1.7.7 Liste des fenêtres

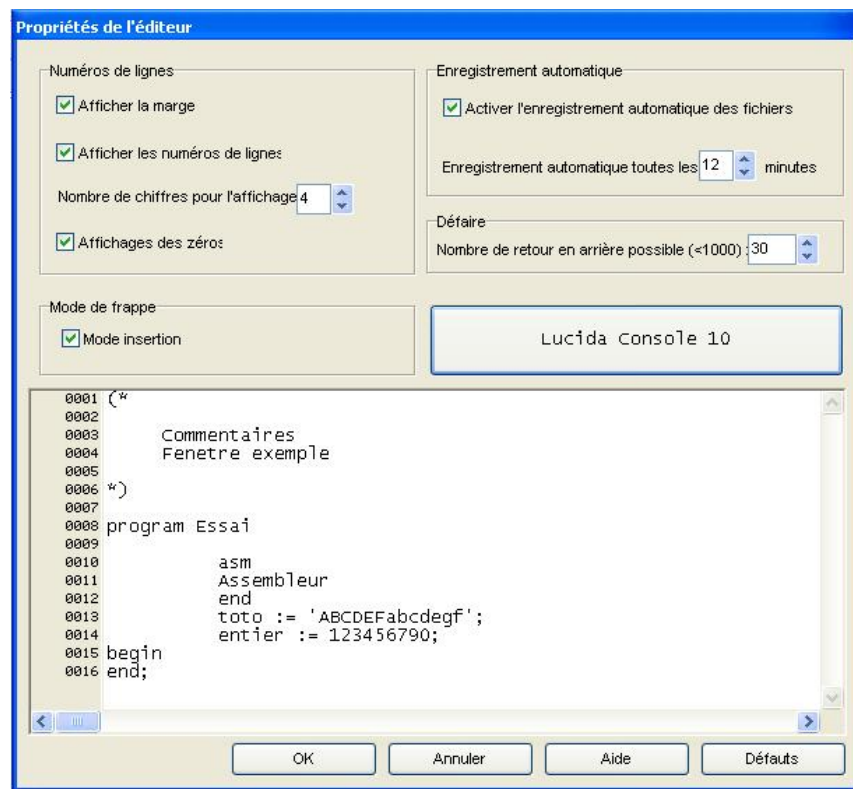
Cette option n'est disponible qu'avec l'éditeur de fichiers d'Optima PLC (et non la liste des POU).

La liste permet de savoir rapidement quels sont les fichiers ouverts mais également de mettre en avant-plan le fichier que l'on sélectionne dans la liste.

### 2.1.7.8 Options Editeur

#### 2.1.7.8.1 Options générales

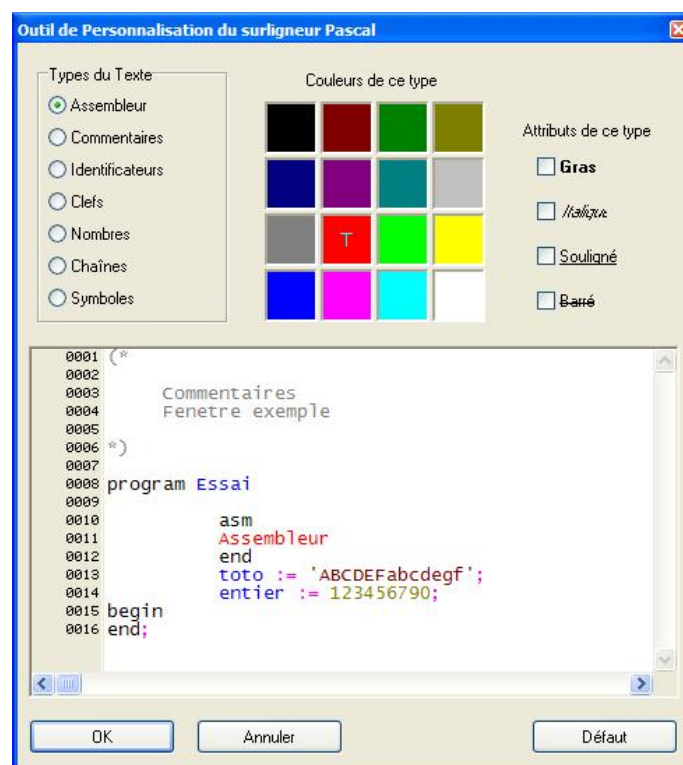
- Numéros de lignes : il est possible d'afficher la marge et le numéro des lignes (sous différents formats)
- Enregistrement automatique : l'enregistrement peut être effectué de façon automatique, dans ce cas il est possible de préciser la période d'enregistrement
- Défaire : l'annulation des dernières frappes est évidemment possible, le nombre d'annulations possible est configurable
- Mode de frappe : la frappe peut se faire en mode insertion ou non
- Police : un bouton indique le nom de la police utilisée, en cliquant dessus il est possible de modifier cette police
- Fenêtre exemple : il est possible de voir en temps réel l'aspect que prendra l'éditeur avec les options sélectionnées



### 2.1.7.8.2 Editeur Pascal

La boîte de dialogue "Outil de personnalisation du surligneur Pascal " permet de spécifier pour chaque type de texte quelle est la couleur et quels sont les attributs de la police qu'il faut utiliser pour mettre en évidence ce type de texte.

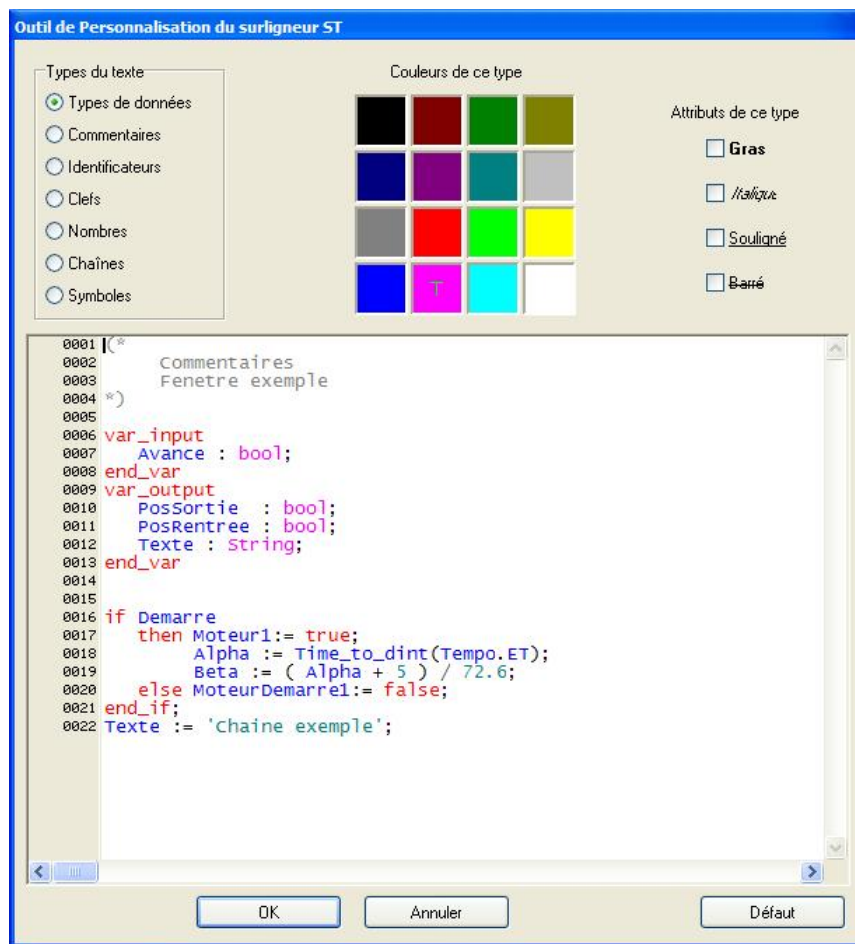
Une fenêtre d'exemple permet de voir en temps réel l'aspect que prendra le texte avec les options sélectionnées.



### 2.1.7.8.3 Editeur ST

La boîte de dialogue " Outil de personnalisation du surligneur ST " permet de spécifier pour chaque type de texte quelle est la couleur et quels sont les attributs de la police qu'il faut utiliser pour mettre en évidence ce type de texte.

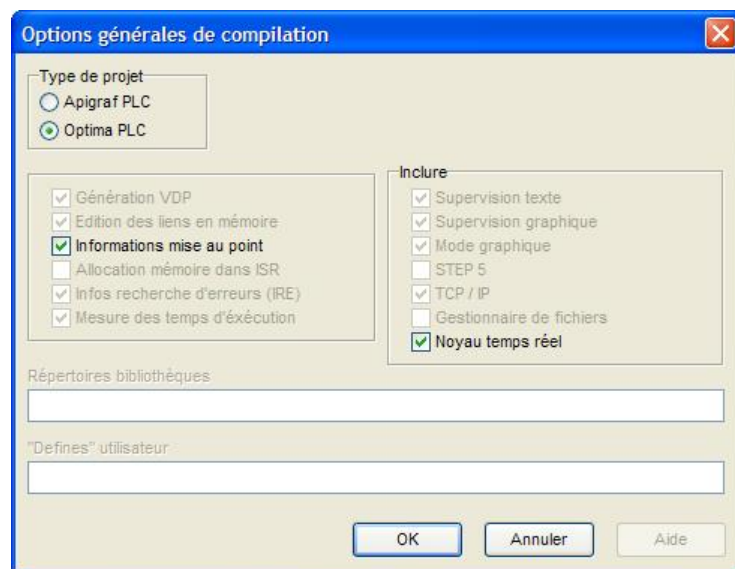
Une fenêtre d'exemple permet de voir en temps réel l'aspect que prendra le texte avec les options sélectionnées.



## 2.1.8 Menu Options

### 2.1.8.1 Compilation générale

Dans un projet Optima PLC, les options suivantes sont disponibles :



- Informations de mise au point

Lorsque cette option est cochée, le compilateur génère du code de vérification supplémentaire pour détecter d'éventuelles erreurs de limites, en particulier sur les indices de tableau. Le programme est donc légèrement plus lent, cependant ces vérifications peuvent détecter rapidement certains problèmes de programmation. Son utilisation en phase de mise au point est fortement recommandée.

- Noyau temps réel

Cette option permet de désactiver le noyau temps réel intégré dans Optima PLC. L'application peut continuer à fonctionner si ce noyau est désactivé.

### 2.1.8.2 Constantes système

Il est possible de définir le nombre maximum de variables automates utilisables dans le programme.

Paramètre	Valeur
NoMaxR	499
NoMaxRR	499
NoMaxL	499
NoMaxG	199
NoMaxSA	499
NoMaxC	99
NoMaxA	199
NoMaxDT	199
InitNbInterSeq	1024
Base de temps	0.001
Taille pile	65520
Taille pile seq	16384
Adresse carte RAM	0

### 2.1.8.3 Editeur de configuration

L'éditeur de configuration permet de visualiser les données contenues dans le fichier d'initialisation du projet.

Editeur de fichiers de configuration Optima PLC/Apigraf

Fichier: C:\Program Files\OptimaLog\OptimaPLC\Examples\Elevator\Elevator.ini

Sections:

- APPLI
  - DEFDLG
  - LANGUAGE
  - MAINFORM
  - OPTIONS
    - ALARMS
    - COMS
    - EVENTSLOG
  - WORKSHOP
    - COMMON
      - OPTIONS
        - COMPILE
          - ST
          - LIBRARIES
  - PROJECT
  - OPTIMAPLC
  - SUPERVISION

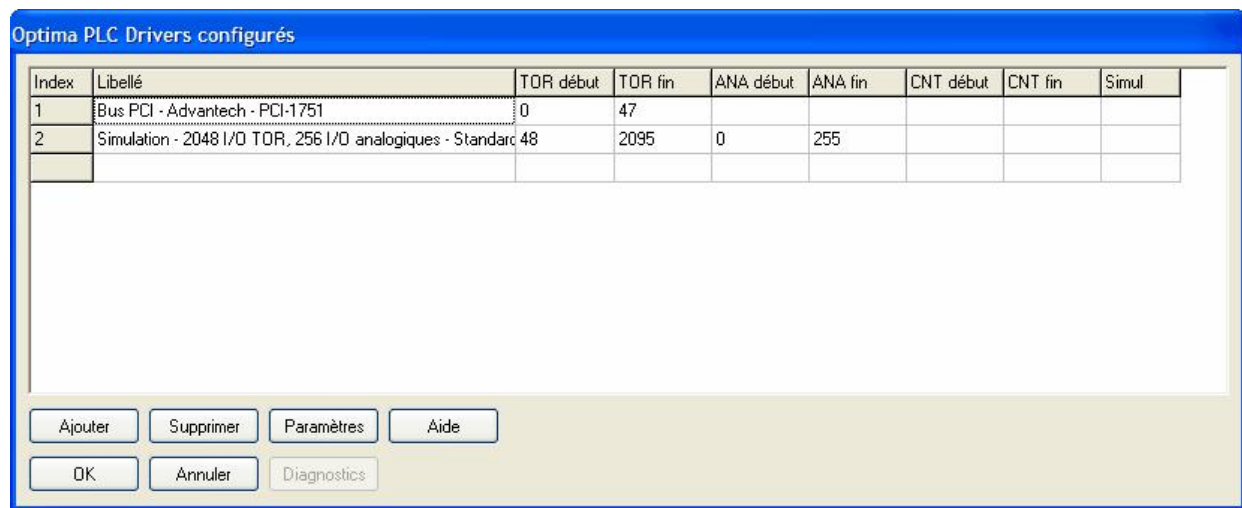
Variables:

Nom	Valeur
-----	--------

Ajouter Supprimer Editer OK Annuler

### 2.1.8.4 Interfaces matérielles

Dans un projet Optima PLC, les drivers définis permettent la gestion des entrées sorties numériques et analogiques.



La boîte de dialogue permet d'ajouter, supprimer ou configurer les interfaces matérielles. Les entrées sorties peuvent également être mise en simulation.

Le bouton "Ajouter" permet l'affichage de la liste des interfaces matérielles disponibles. Lorsque le choix est validé, la boîte de dialogue de paramétrage s'affiche aussitôt ; certains paramètres sont obligatoires pour créer l'interface (exemples : adresse de la carte, adresse IP...).

Lorsqu'une interface matérielle est ajoutée, des numéros d'entrées-sorties et de canaux sont automatiquement réservés. La plage de numéros s'affiche dans la liste des interfaces, cependant pour connaître l'affectation des numéros aux entrées, aux sorties, aux canaux d'entrées et à ceux de sortie, il est nécessaire de consulter l'aide associée à l'interface sélectionnée.

### 2.1.8.5 Fichiers de sortie texte

Cette option permet de définir les accès aux fichiers de sortie texte utilisés dans l'application. Ce terme regroupe aussi bien les sorties sur disque réelles que les sorties sur imprimantes parallèles (LPT1, LPT2, etc.). Ici, seules les sorties sur fichier disque sont concernées.

Les sorties sont effectuées "au fil de l'eau", c'est à dire que les données écrites sont ajoutées à la suite des précédentes.

Seules les sorties sur disque peuvent être configurées ici. Avec ce type d'accès, les fichiers croissent à chaque nouvelle écriture de données. Ces données doivent donc être réinitialisées, par exemple en utilisant les fonctions propres aux fichiers : RenameFile (renommer le fichier), DeleteFile (suppression du fichier).

Dans le programme automate, les données peuvent être écrites dans les fichiers grâce aux fonctions TOWrite et TOWriteLn (idem TOWrite avec retour à la ligne).

#### Exemple :

```
if SA10
then TOWriteLn( Essai, R1, '$T', R2 );
    SA10:= false;
end_if;
```

Lorsque le flag SA10 passe à 1, la valeur des registres R1 et R2, séparés par une tabulation, sont écrits dans le fichier dont le nom interface est "Essai". Un retour à la ligne suivante est automatiquement réalisé.

La boîte de dialogue de configuration permet de paramétrer les fichiers de sortie texte (sur disque) utilisés dans l'application.









Cette information constitue l'identificateur du tableau dans l'application, autrement dit son code d'accès. Les opérations sont toujours faites en indiquant cet identificateur.

L'identificateur ne doit comporter que des lettres ou des chiffres, et commencer par une lettre.

### Libellé

Ce champ est utilisé lors de l'affichage des informations sur le tableau, et permet de rappeler sa signification.

### Unités

Les unités des valeurs placées dans le tableau doivent être spécifiées ici.

### Coefficient de conversion des unités

Ce coefficient permet de corriger les valeurs stockées dans le tableau pour affichage. Par exemple, si les valeurs sont mémorisées en millisecondes, mais si les affichages doivent être faits en secondes, alors les unités sont "secondes", et le coefficient de conversion est 0.001.

### Nombre de points

Ce paramètre indique la taille du tableau. Chaque valeur pouvant être stockée dans le tableau occupe 4 octets (valeurs entières 32 bits).

### Type de données

Ce paramètre permet d'indiquer le type de données contenues dans le tableau. Il est aussi utilisé pour l'affichage dans les courbes, dans le cas où le tableau sert d'abscisse. Les valeurs possibles sont les suivantes :

Normal (DInt)	Valeurs générales entières DInt
FacteurSeconde (DInt)	Les valeurs numériques correspondent à des temps absolus en seconde. Dans ce cas, les valeurs affichées sont l'heure et la date correspondantes.
FacteurSeconde sans date (DInt)	Même chose que l'option précédente, mais la date n'est pas affichée (heure uniquement).
Valeur booléenne sur 8 bits (bool)	Valeurs booléennes
Entiers 16 bits signés (Int)	Valeurs générales entières Int
Durée sur 64 bits (Time)	Valeurs de durée
Heure de la journée sur 64 bits (Time_of_Day)	Heure du jour
Date et heure sur 64 bits (Date_and_Time)	Valeurs de date et heure
Réel court sur 32 bits (Real)	Valeurs générales Real
Réel long sur 80 ou 64 bits (LReal)	Valeurs générales LReal

- **Ajouter**

Cette commande permet d'ajouter un nouveau tableau dans la liste. La boîte de paramétrage apparaît (voir ci-dessus), permettant de définir les paramètres du tableau.

Le nouveau tableau est ajouté dans la liste à la position sélectionnée au préalable.

- **Supprimer**

Cette commande supprime, après une confirmation, le tableau de sortie sélectionné dans la liste.

## 2.1.8.7 Divers

### 2.1.8.7.1 Configuration du démarrage

Les options de configuration du démarrage concernent le choix du projet et l'ouverture de fichiers dans l'éditeur Optima PLC.

### 2.1.8.7.2 Gestion des utilisateurs

Cette option permet de choisir le mode de gestion des utilisateurs, dans lequel une identification est demandée préalablement à l'accès aux écrans de l'atelier. Ce mode offre une meilleure sécurité dans un environnement multi-utilisateurs, comme cela est souvent le cas sur une machine de production.

Attention : il ne faut pas confondre la gestion des utilisateurs Optima PLC, proposée ici, avec celle de Windows. Windows permet de définir des utilisateurs différents ayant chacun la possibilité d'ouvrir une session, avec des droits différents. L'utilisation multi-utilisateur d'Optima PLC ne peut en revanche s'appuyer sur cette notion de session Windows, qui imposerait une lourdeur incompatible avec l'application considérée.

Lorsque ce mode est activé, les droits attachés à l'utilisateur lui restreignent l'accès à certaines fonctions et menus (encore une fois, il s'agit des droits propres à Optima PLC, et non aux utilisateurs Windows). De plus, un menu spécial de gestion des utilisateurs est affiché.

### 2.1.8.7.3 Session Windows pour exécution

Cette option permet d'ouvrir une boîte de dialogue avec laquelle la session Windows utilisée pour l'exécution de l'application peut être paramétrée.

Les paramètres correspondent aux informations de login de l'utilisateur dont les droits doivent être utilisés pour exécuter l'application. Ces paramètres sont mémorisés dans le fichier de configuration de l'application (le mot de passe est crypté).

Cette configuration n'est pas nécessaire si la session utilisée pour lancer l'application est une session administrateur. Les droits d'administrateur sont en effet indispensables pour autoriser l'accès à certaines ressources du PC.

Une session non administrateur est toutefois conseillée pour limiter les risques d'accès à des données non autorisées, de destruction de données, de corruption de configuration, etc... Dans ce cas, l'application doit fonctionner dans une session administrateur configurée ici.

Depuis Vista, la session Administrateur de base du PC est idéale pour cela, car elle ne nécessite pas d'élévation et permet de configurer un démarrage automatique de l'application.

Le bouton "Test" permet de vérifier les paramètres. Si la session configurée est correcte, la fenêtre suivante s'affiche :



#### 2.1.8.7.4 Paramétrage des sessions Windows

Cette option donne accès à la fonction de paramétrage des sessions de Windows.



Cette boîte de dialogue n'est disponible que si Optima PLC est lancé actuellement depuis une session administrateur.

Pour définir la session avec laquelle Windows démarre automatiquement par défaut, décocher la case "Les utilisateurs doivent entrer un nom d'utilisateur et un mot de passe pour utiliser cet ordinateur".

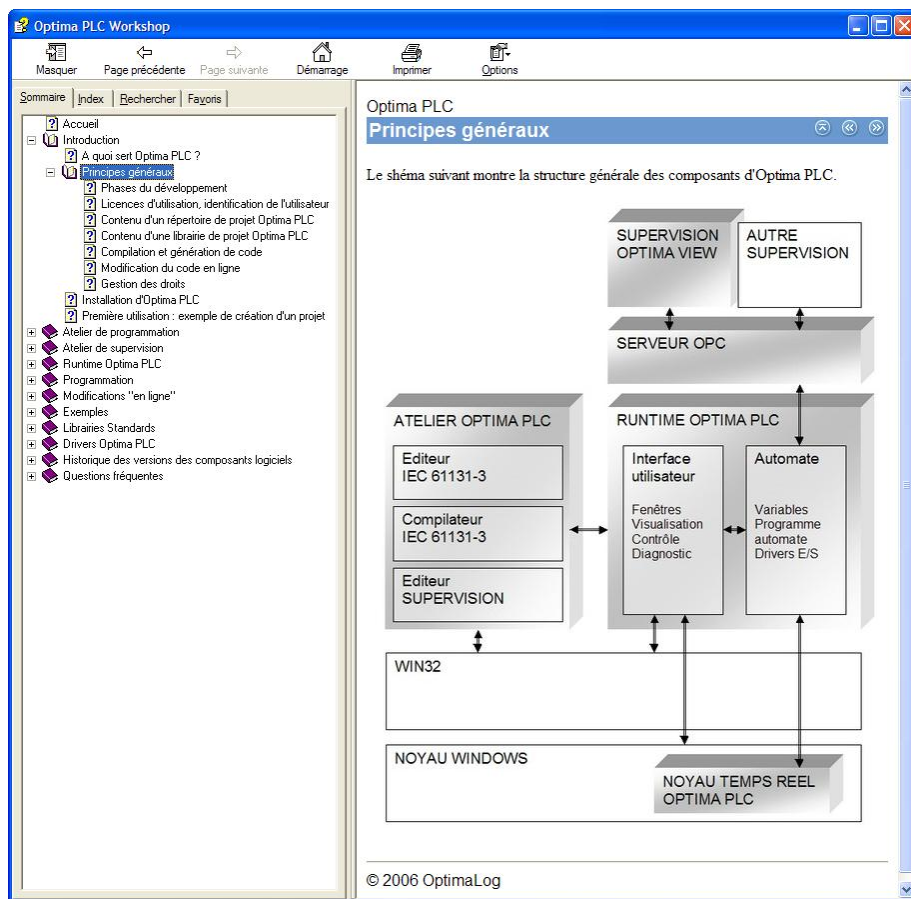
#### 2.1.8.7.5 Fichiers défaut

Il est possible de définir le type de fichier proposé par défaut lors de la demande d'ouverture d'un fichier dans l'éditeur Optima PLC.

### 2.1.9 Menu Aide

#### 2.1.9.1 Aide

Il est possible de visualiser ce fichier d'aide depuis le progiciel Optima PLC.

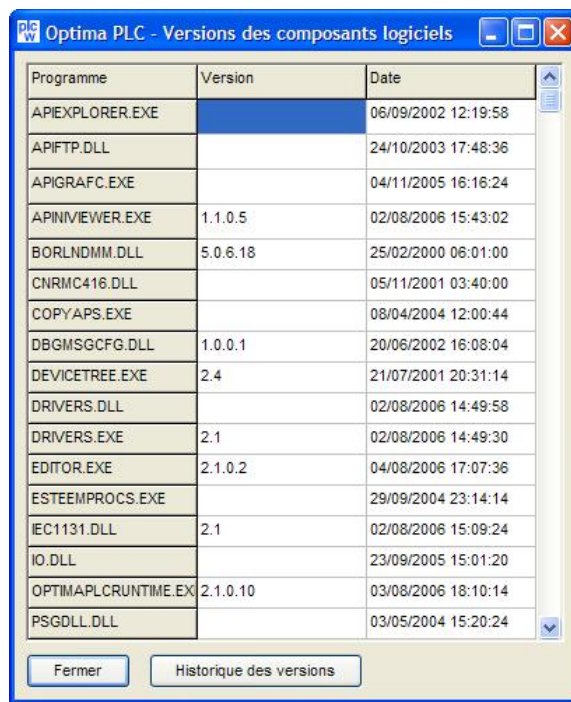


### 2.1.9.2 A propos d'Optima PLC Workshop

Cette option permet d'obtenir les informations de version du logiciel.



Le bouton Versions provoque l'affichage de la liste des composants logiciels et de leur version.



## 2.2 Description de la barre d'icônes

### 2.2.1 Barre d'outil de gestion du projet

	<b>nouvelle librairie</b> : il est possible d'ajouter une librairie dans un projet. La librairie est créée et ajoutée à la suite des librairies existantes. Le nom de la nouvelle librairie doit être entré à ce moment, et le fichier .opl correspondant vide est créé sur le disque.
	<b>ouvrir une librairie</b> : il suffit de sélectionner un fichier avec l'extension .opl, la librairie correspondante est alors intégrée au projet.
	<b>enregistrer</b> : les librairies modifiées sont sauvegardées. L'appui sur la touche F2 a le même effet.
	<b>descendre</b> : le POU sélectionné est déplacé vers le bas dans l'arborescence de la bibliothèque
	<b>monter</b> : le POU sélectionné est déplacé vers le haut dans l'arborescence de la bibliothèque
	<b>compiler</b> : une compilation complète de tous les POU est effectuée, suivie d'une génération du code exécutable. Si l'application est en cours de fonctionnement, cette icône permet d'activer la <b>modification en ligne</b> .
	<b>exécuter</b> : le programme, actuellement à l'arrêt, est compilé s'il ne l'est pas déjà, puis est exécuté.
	<b>stopper</b> : le programme actuellement en fonctionnement est arrêté après une confirmation.

### 2.2.2 Barre d'outil de gestion de l'affichage des fenêtres

Ces options concernent uniquement les fenêtres d'édition des POU ouvertes dans la zone de droite de la fenêtre principale.

	permet de réduire toutes les fenêtres
	les fenêtres sont affichées avec leur taille et position normales
	toutes les fenêtres sont maximisées
	toutes les fenêtres sont fermées

## 2.3 Utilisation des librairies
















### 2.3.1 Principes généraux des librairies

Une librairie Optima PLC est un fichier contenant des éléments de programmation (POUs). L'extension de ce type de fichier est .OPL (Optima PLC Library).

Un projet pouvant être constitué de plusieurs librairies, ces librairies permettent d'obtenir une bonne modularité de programmation, en regroupant par thème les POUs dans des librairies spécifiques. Par exemple, les éléments de programmation standard de base fournis avec Optima PLC (plusieurs centaines de POUs) sont regroupées dans des librairies spécialisées.

#### 2.3.1.1 Contenu d'une librairie

Une librairie est constituée de POUs de différents types :

	<a href="#">tâche</a> 	cadencement de l'exécution des programmes.
	<a href="#">variables globales</a> 	déclarations de variables disponibles pour l'ensemble des POUs de programmation.
	<a href="#">types</a> 	déclarations de type de variables disponibles pour l'ensemble des POUs de programmation.
	<a href="#">bloc de programme</a> 	programmation principale.
	<a href="#">bloc de fonction</a> 	sous-programme, avec ou sans valeur de retour, qui mémorise l'état de ses variables internes entre 2 appels. Ainsi pour un même jeu de paramètres d'entrée le résultat renvoyé par ce bloc peut être différent (exemple du PID : pour une même consigne la sortie de ce bloc de régulation évolue jusqu'à ce qu'elle atteigne la consigne).
	<a href="#">fonction</a> 	sous-programme dont la valeur de retour reste identique pour un même jeu de paramètres d'entrée (pas de mémorisation des variables internes).
	Action	sous-programme utilisé dans les diagrammes SFC, permettant de décrire une action à effectuer depuis les étapes.
	<a href="#">sous-groupe</a> 	POU de structuration de la librairie. Son unique objet est de contenir lui-même d'autres POUs.

##### 2.3.1.1.1 Tâche

Conformément à la norme IEC 61131-3, une tâche est un POU qui provoque l'exécution de programmes ou de blocs de fonction soit périodiquement, soit en réponse au changement d'état d'une variable booléenne.

Les paramètres associés à une tâche sont :

- son nom
- le niveau de priorité pour l'exécution des programmes ou blocs de fonction

Les niveaux de priorité des tâches sont définis de 0 à 9 comme suit :

Niveau de priorité	Utilisation	Temps réel
0	Priorité maximale, utilisée en interne par Optima PLC.	oui
1		oui
2		oui
3		oui

<b>4</b>	Niveau de priorité "standard" pour le programme automate. L'actualisation des tables d'images des entrées/sorties est effectuée à ce niveau de priorité, qui doit donc être utilisé pour un fonctionnement "synchrone" avec ces mises à jour.	oui
<b>5</b>		oui
<b>6</b>		oui
<b>7</b>	Priorité temps réel minimale	oui
<b>8</b>	Niveau de priorité non temps réel, à utiliser pour les tâches effectuant des appels brefs à l'API de Windows et devant s'exécuter à une fréquence élevée (réseau, ...).	non
<b>9</b>	Niveau de priorité non temps réel, à utiliser pour les tâches effectuant des appels non prioritaires à l'API de Windows (fichiers, ...).	non

- le trigger (c'est-à-dire le nom de la variable booléenne optionnelle provoquant l'exécution).

Cette variable, si elle est définie, doit être de type BOOL, et être accessible au niveau global. Dans ce cas, la tâche s'exécute lors d'un front montant de la valeur de cette variable.

- la période d'exécution de la tâche

La période doit être définie à l'aide de la syntaxe IEC 61131-3, c'est à dire avec le préfixe T#. Par exemple : T#5ms (5 millisecondes), T#4s (4 secondes), etc.

Eventuellement, la période peut être précisée à l'aide d'une variable globale de type TIME.

Si ni la période ni le trigger ne sont définis, la tâche est exécutée à la fréquence maximale autorisée par le système (actuellement 1000 Hz).

- la liste des programmes ou blocs de fonction à gérer

Cette liste comprend, pour chaque programme, le nom de variable, le type de bloc de programme, et la liste éventuelles des paramètres.

- les phases d'exécution du runtime durant lesquelles la tâche est autorisée.

Le runtime passe par plusieurs phases d'exécution. Les tâches définies par l'utilisateur peuvent être autorisées ou non à s'exécuter durant certaines d'entre elles :

<b>Phases</b>	<b>Interprétation</b>
START PLC	Tâche d'initialisation, permettant d'effectuer certaines opérations uniquement lors du démarrage de l'application. Lorsque la tâche a terminé son

	initialisation, elle doit impérativement passer à 1 la variable booléenne StartPLCPhaseOK
RUN	Phase de fonctionnement normal du runtime.
FAULT	Phase de fonctionnement dégradé, un sous-système matériel au moins n'est pas opérationnel.
STOP PLC	Tâche de finalisation, permettant d'effectuer certaines opérations uniquement lors de la fin de l'exécution de l'application. Lorsque la tâche a terminé son exécution, elle doit impérativement passer à 1 la variable booléenne StopPLCPhaseOK

Dans la plupart des cas, les tâches classiques sont autorisées à fonctionner en phase RUN et FAULT.

#### 2.3.1.1.2 Variables globales

Un POU "Variables globales" permet de déclarer des variables qui pourront ensuite être utilisées par tous les POU de programmation (déclarés à la suite dans le projet).

Comme ce POU est dédié à la déclaration de variables, l'encadrement des déclarations avec les mots clés VAR et END\_VAR est automatiquement inséré à la création du POU. Cependant, d'autres déclarations peuvent être ajoutées, avec VAR CONSTANT ou VAR RETAIN. Le POU peut ainsi contenir plusieurs blocs VAR END\_VAR.

Les variables peuvent être des variables automates ou d'entrées/sorties, dans ce cas la déclaration n'a pas pour but de "créer" la variable mais simplement d'affecter un mnémonique et/ou un commentaire.

#### 2.3.1.1.3 Types

Un POU "Types" permet de déclarer des types de variables qui pourront ensuite être utilisées par tous les POU de programmation (déclarés après dans le projet).

Comme ce POU est dédié à la définition de types, l'encadrement des déclarations avec les mots clés TYPE et END\_TYPE est automatiquement inséré à la création du POU.

#### 2.3.1.1.4 Bloc de programme

Un bloc de programme est un assemblage logique d'instructions et de POU.

Il se compose d'une partie déclaration locale et d'une partie implémentation.

Les variables déclarées peuvent être des variables d'entrée, de sortie, locales ou temporaires, les encadrements de ces différentes déclarations avec les mots clés de la norme IEC 61131-3 sont automatiquement insérés à la création du POU.

A la différence des blocs de fonction, les programmes peuvent contenir des déclarations "VAR\_GLOBAL", qui seront ensuite utilisées dans des blocs de fonction avec la déclaration "VAR\_EXTERNAL".

L'implémentation peut s'effectuer dans tous les langages disponibles dans Optima PLC.

#### 2.3.1.1.5 Bloc de fonction

Un bloc de fonction est un sous-programme, disposant de variables internes rémanentes.

Comme le bloc de programme, il se compose d'une partie déclaration locale (de variables et/ou de types) et d'une partie implémentation.

Les variables déclarées peuvent être des variables d'entrée, de sortie, locales ou temporaires, les encadrements de ces différentes déclarations avec les mots clés de la norme IEC 61131-3 sont automatiquement insérés à la création du POU.

L'implémentation peut s'effectuer en ST, IL, SFC, ladder (texte ou diagramme) ou diagramme de blocs de fonctions.



### 2.3.1.1.6 Fonction

Une fonction est un sous-programme dont les possibilités sont moindre que celles d'un bloc de fonction puisque les variables internes ne sont pas mémorisées entre deux appels et qu'une seule variable de sortie est autorisée.

Comme le bloc de programme, il se compose d'une partie déclaration locale (de variables et/ou de types) et d'une partie implémentation.

Les variables déclarées peuvent être des variables d'entrée ou temporaires, les encadrements de ces différentes déclarations avec les mots clés de la norme IEC 61131-3 sont automatiquement insérés à la création du POU.

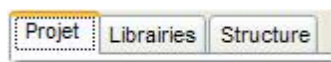
L'implémentation peut s'effectuer dans tous les langages disponibles dans Optima PLC sauf en SFC.

### 2.3.1.1.7 Sous-groupe

Le sous-groupe est un élément de structuration, il ne comporte ni déclaration ni instruction. Son but est de grouper des POU afin que la librairie soit organisée de façon logique.

## 2.3.1.2 Arborescences des POU

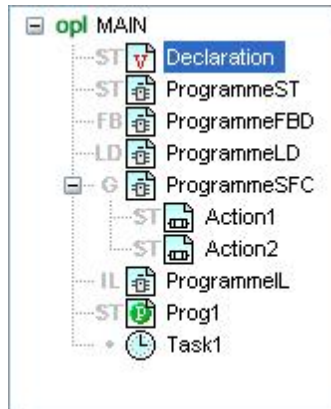
La partie gauche de l'écran d'Optima PLC permet de visualiser les POUs à partir de vues arborescentes décrites ci-après. Le choix s'effectue avec l'onglet suivant :



### 2.3.1.2.1 Arbre de conception "projet"

La composition d'une librairie s'affiche sous une forme arborescente (panneau à onglet de gauche, onglet "projet").

L'arborescence d'une librairie constitue **l'arbre de conception**. Il ne reflète pas les relations de dépendance et d'appartenance qui existent entre les blocs, contrairement à **l'arbre d'exécution affiché dans l'onglet Structure et dans le runtime**, mais seulement la structure définie par le concepteur par commodité d'accès.



Seule exception, les actions sont affichées en dépendance directe du bloc auquel elles appartiennent, et cette relation est identique dans l'arbre de conception et dans l'arbre d'exécution.

L'arbre de conception "projet" peut contenir plusieurs librairies. Il s'agit normalement des librairies conçues spécifiquement pour le projet en cours et situées dans le répertoire de ce projet.

### 2.3.1.2.2 Arbre de conception "librairies"

Cette arborescence, similaire à l'arborescence "projet", est disponible dans l'onglet "librairies". Elle permet d'accéder spécifiquement aux librairies standards d'Optima PLC, c'est à dire celles qui sont installées en même temps que le produit, ou que l'utilisateur a créées et placées dans le répertoire OPL, afin d'en faciliter le partage entre plusieurs application.

Les librairies de cette arborescence sont toutes accessibles depuis les librairies de l'onglet projet.

Les boutons suivants sont disponibles dans cet onglet :



- Autres librairies disponibles

Ce bouton permet d'afficher la liste des librairies standards disponibles, mais non chargées actuellement dans le projet.

- Optimiser les librairies chargées

Lorsque des librairies sont chargées sans être utilisées dans le projet, elles allongent inutilement le temps de compilation. Ce bouton permet de décharger automatiquement les librairies standards inutilisées dans le projet.

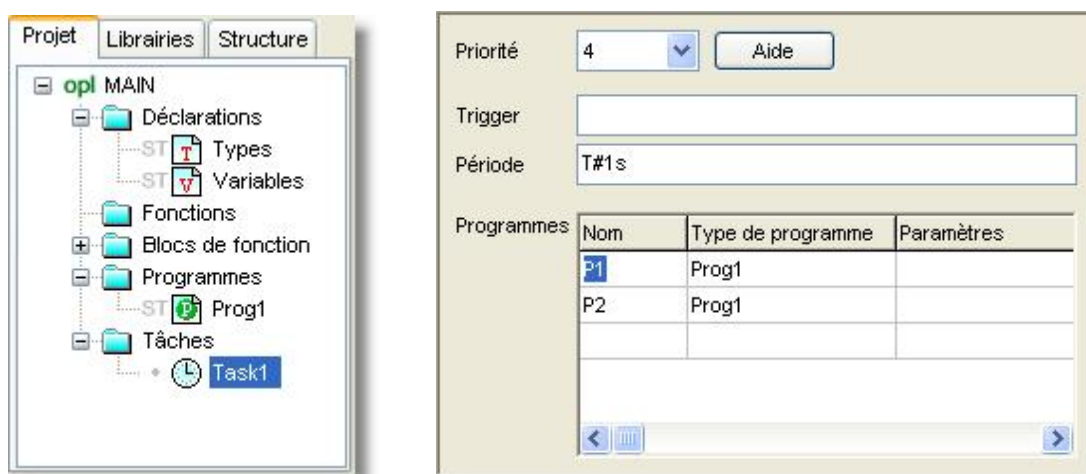
### 2.3.1.2.3 Structure (arbre d'exécution)

La structure des POU de l'application correspond à l'imbrication définie par les déclarations faites dans chaque POU, et donc à la structure des données finale de l'application. Elle s'affiche sous forme arborescente dans le panneau à onglet "Structure".

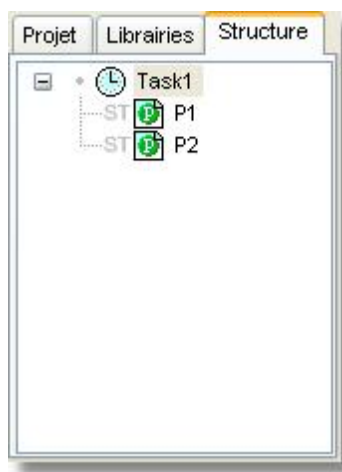
La structure part de chaque tâche définie, qui contient des programmes, eux-même contenant des blocs de fonctions, etc... Cette structure est aussi appelée dans Optima PLC "arbre d'exécution". Elle peut être également visualisée dans le runtime.

Dans cette structure, les noms des instances de POU (c'est à dire les variables) sont affichées, et non pas les noms des types de POU. D'autre part, les librairies et sous-groupes n'apparaissent pas ici (ils servent uniquement à ranger efficacement les POU disponibles et n'ont pas d'incidence sur la structure).

Voici par exemple un projet contenant une tâche et un programme. La tâche appelle le programme 2 fois, sous les noms de variables P1 et P2.



La structure résultante est la suivante :



Dans cette vue de la structure, aucune opération de modification de celle-ci n'est possible dans cette version d'Optima PLC. Cependant la sélection d'un élément de l'arbre provoque l'affichage dans la partie droite de l'écran, des données du POU correspondant.

Les modifications effectuées dans les déclarations des POU provoquent la mise à jour de la structure s'il n'y a pas d'erreur de compilation. En cas d'erreur, après correction, utiliser la fonction de reconstruction de l'application pour mettre à jour la structure.

### Option Afficher toutes les références

Cette option affichée en bas d'onglet permet d'afficher, en plus de la structure des blocs et programmes, l'ensemble des références aux fonctions et définitions de types ou de variables.

#### 2.3.1.3 Librairies "Projet"

Une librairie peut être définie comme étant une librairie "Projet". Il s'agit d'une distinction d'organisation, afin de repérer les librairies vraiment spécifiques à un projet, et de les distinguer de celles d'usage générale.

Un onglet permet de choisir entre l'affichage des librairies "projet" et des librairies "standards".

Les librairies "projets" sont toujours placées après les autres dans [l'arbre de conception](#)<sup>39</sup>, ce qui leur permet de faire référence à tous les POU placés dans les librairies standards.

#### 2.3.1.4 Ordre d'apparition des POU

L'ordre d'apparition des POU dans la librairie est très important : dans Optima PLC, un POU ne peut faire référence qu'aux POU qui sont placés au-dessus de lui dans l'arborescence. Pour cette raison, les déclarations de types et de variables globales précèdent généralement les définitions de blocs. Les tâches sont généralement définies en dernier.

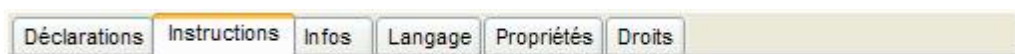
Un POU peut faire référence aux POU placés dans d'autres librairies, à condition que celles-ci soient placées avant dans l'arbre de conception.

Les librairies de l'onglet "Projet" sont placées après celles de l'onglet "Librairies".

#### 2.3.1.5 Propriétés des POU

Pour chaque POU, il est possible de visualiser les données dans la panneau à onglet à droite de l'arborescence, le nombre d'onglets disponibles dépend du type de POU ainsi que des droits de l'utilisateur.

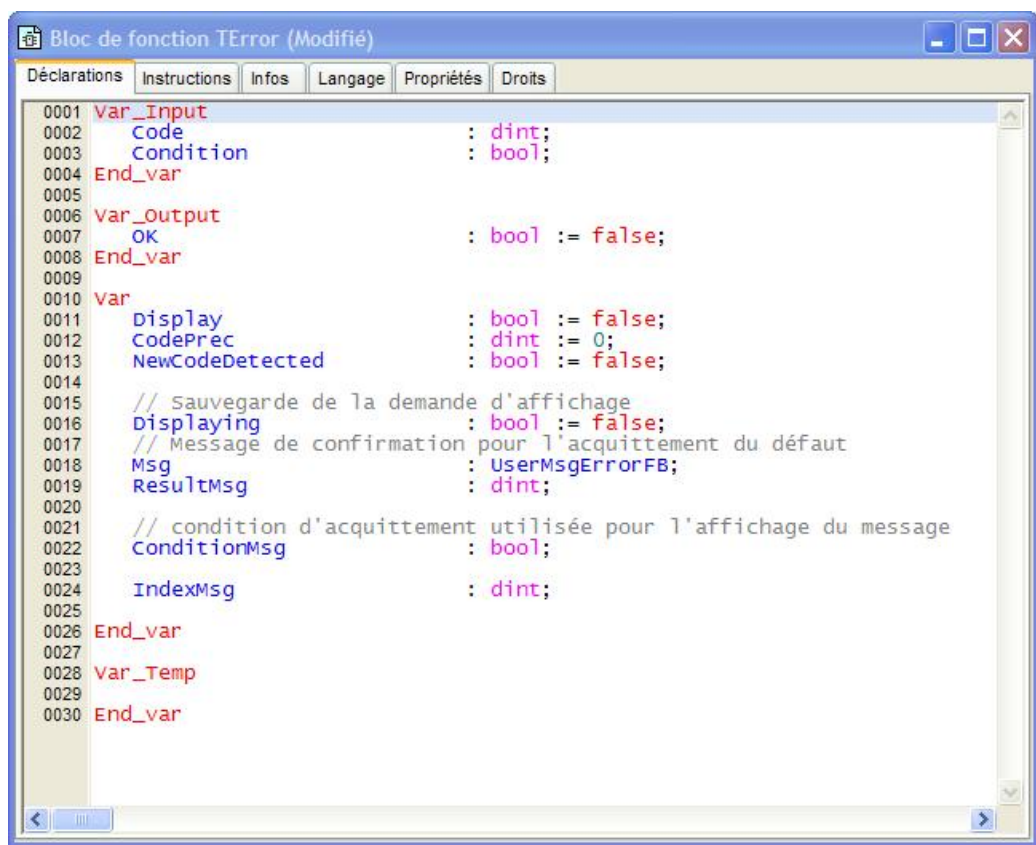
Exemple pour un bloc de fonctions :



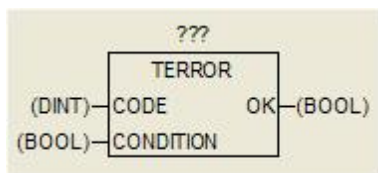
##### 2.3.1.5.1 Page Déclarations

Contrairement aux pages "Variables globales" et "Types", il est indispensable dans cette page d'encadrer les déclarations par les mots clés de la norme IEC 61131.

Les mots clés ainsi que le format de déclarations à respecter sont décrits dans le chapitre "Programmation" de cette documentation.

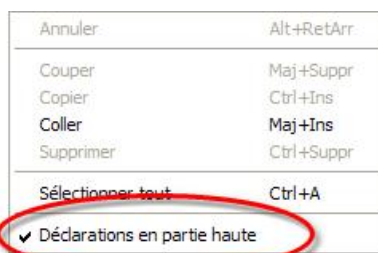


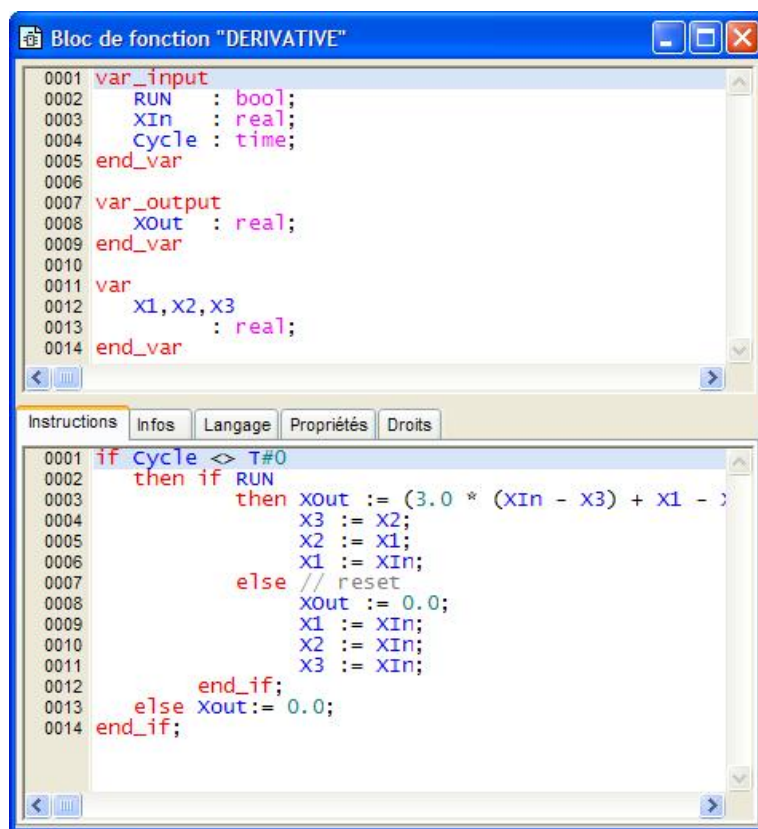
Les déclarations ci-dessus correspondent à un bloc dont voici l'interface :



Les variables "Code" et "Condition" sont définies en entrée, la variable "OK" en sortie. Quant aux autres variables définies dans "var", il s'agit de variables statiques utilisées uniquement en interne dans le bloc et qui ne sont pas accessibles de l'extérieur.

Le menu contextuel de l'éditeur des déclarations contient une option "Déclarations en partie haute", permettant de sortir les déclarations de l'onglet et de les afficher en haut de la fenêtre. Le mode sélectionné est conservé pour les affichages ultérieurs.

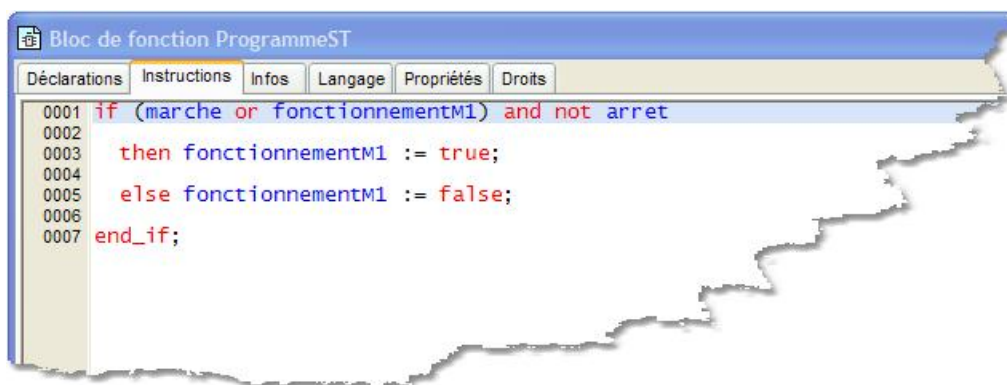




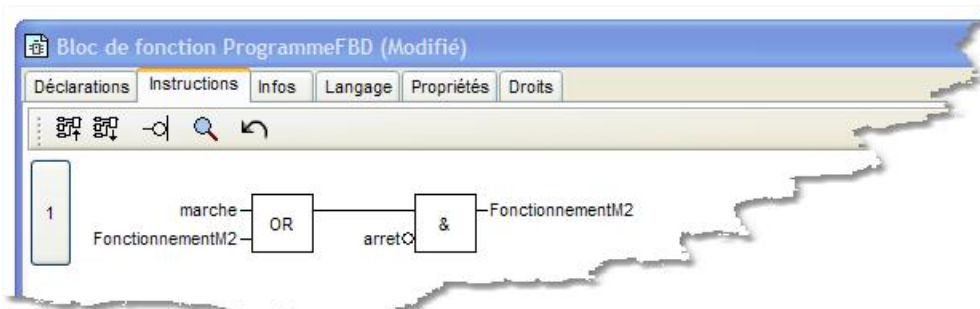
### 2.3.1.5.2 Page Instructions

Cette page prend un aspect différent en fonction du type de langage sélectionné.

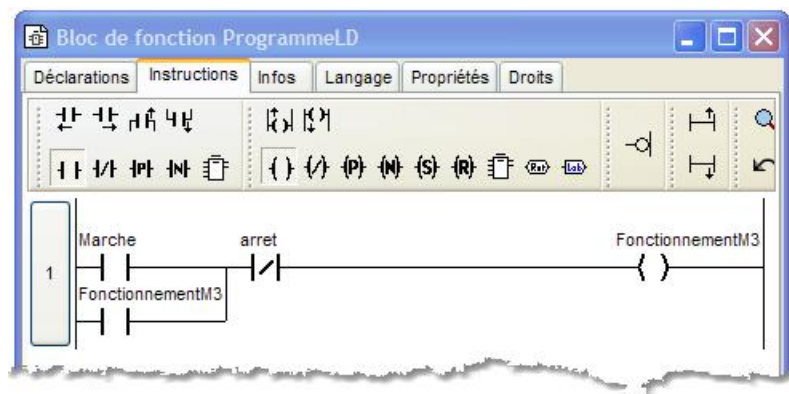
ST, IL et SFC texte : la page se comporte comme un simple éditeur de texte. La syntaxe propre au langage est cependant mise en valeur.



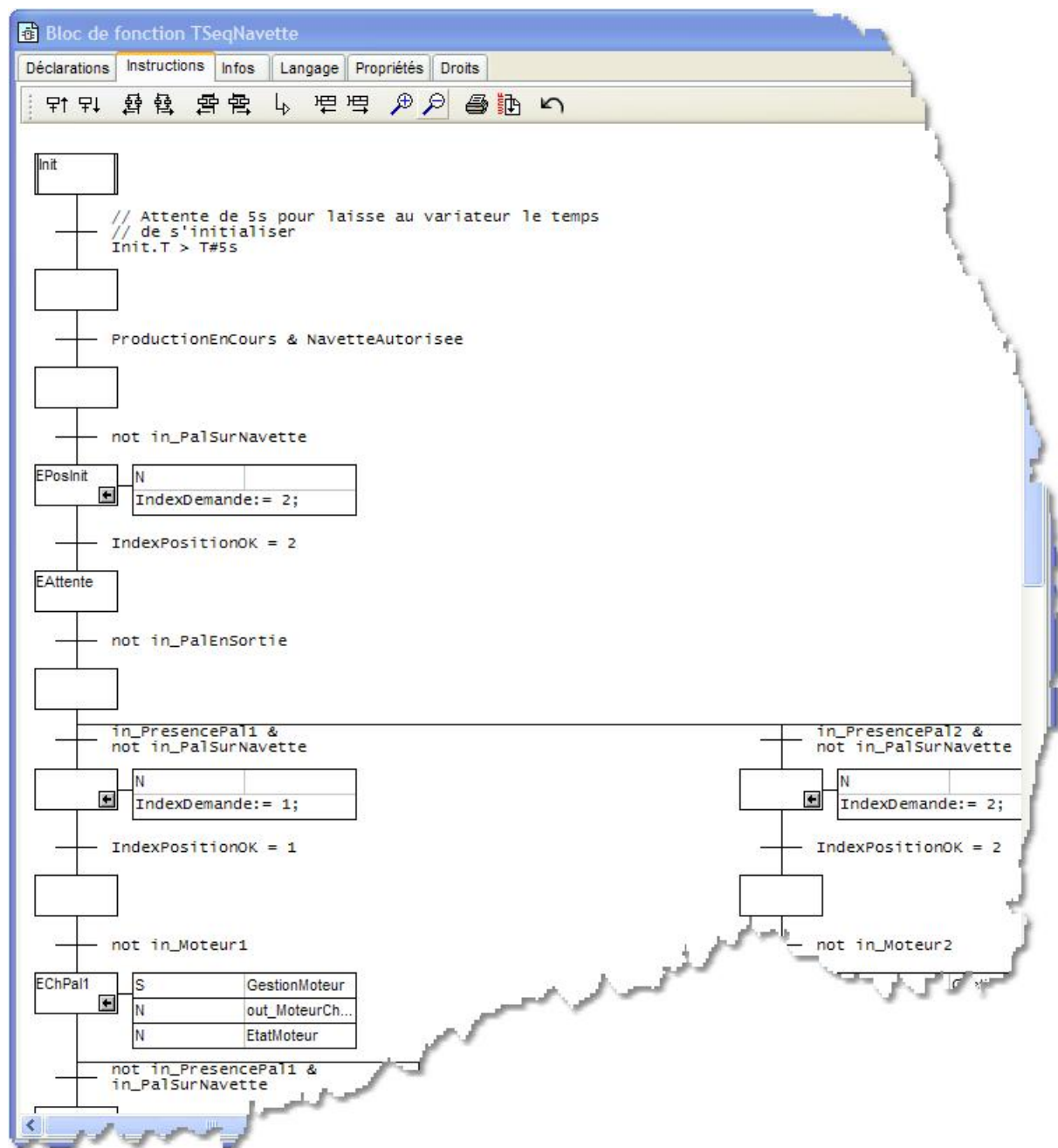
Diagrammes de Bloc de fonctions, Ladder et SFC : la page se compose d'une barre d'icône et d'une zone d'édition pour le diagramme



**Blocs de fonctions**



Ladder



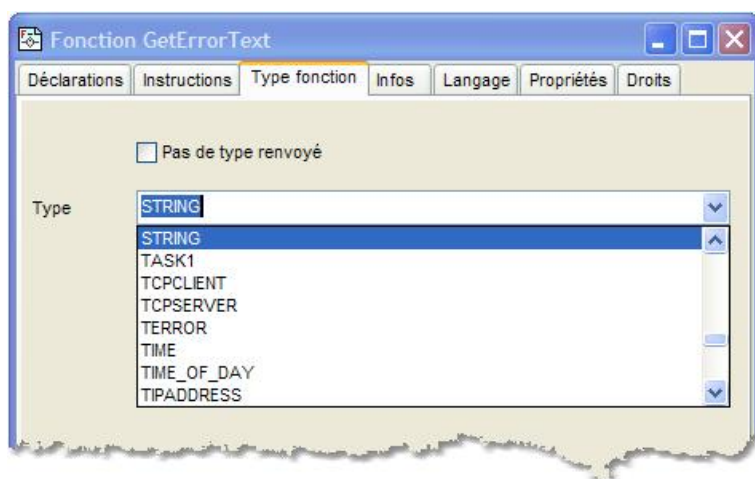
SFC ou GRAFCET

Les mots clés ainsi que le format des instructions sont décrits dans le chapitre "Programmation" de cette documentation.

L'utilisation des icônes dans les langages graphiques est décrite dans le chapitre "[Création d'une librairie](#)" 52".



### 2.3.1.5.3 Page Type fonction



Cette page permet de définir le type de la variable de sortie de la fonction.

Si la case "Pas de type renvoyé" est décochée, la fonction ne renvoie aucune valeur, il s'agit d'une "procédure".

Dans le cas contraire, entrer sur la ligne du type le nom du type de valeur renvoyé. La ligne permet d'afficher la liste de tous les types connus du système.

### 2.3.1.5.4 Page Langage

La page se compose d'une boîte à choix unique parmi les propositions.



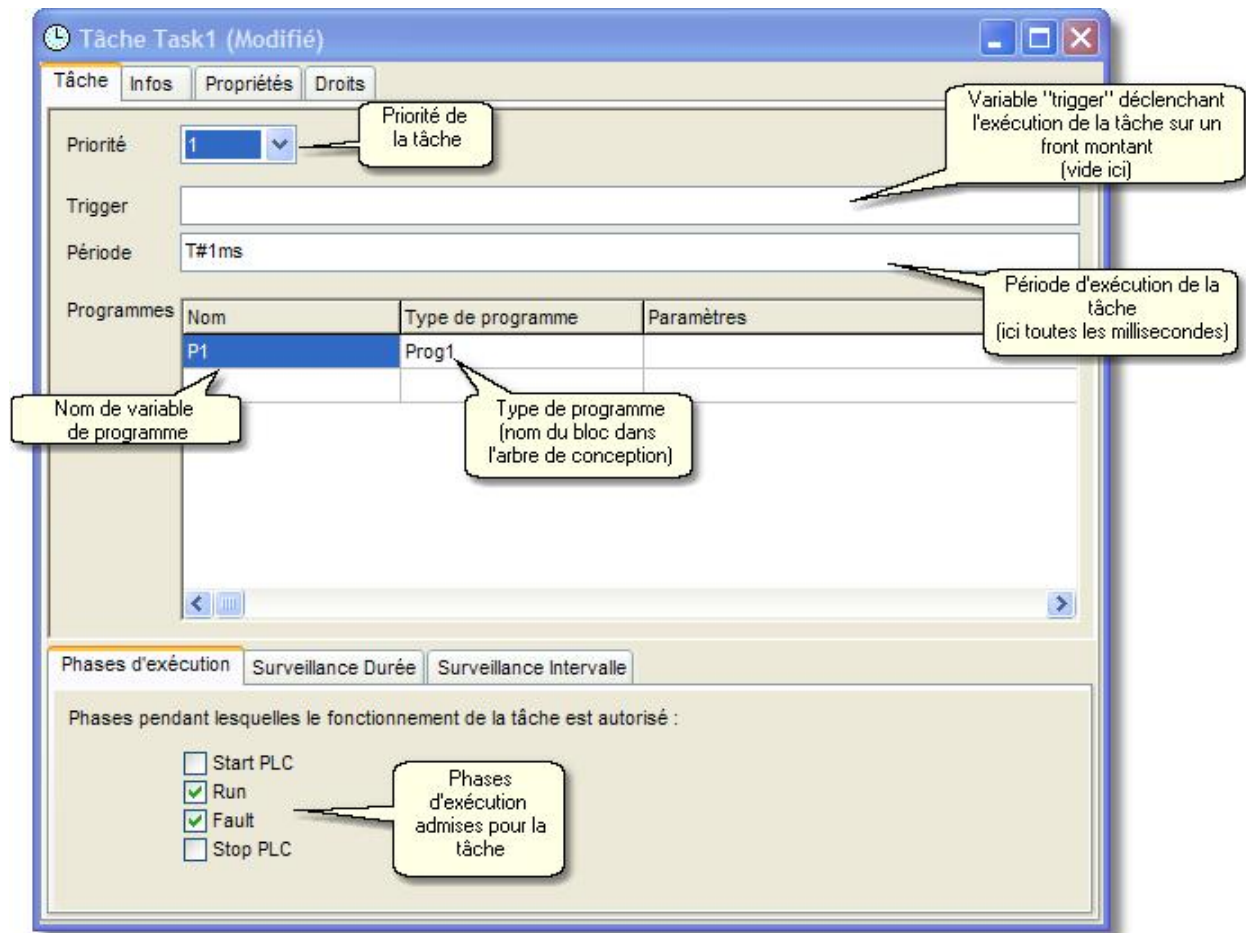
Les langages proposés sont le ST, l'IL, le SFC (sous format texte ou diagramme), le ladder, le diagramme de blocs et le langage Apigraf.

Le langage est généralement choisi à la création du POU, cependant il peut être modifié dans cette page. Aucun outil de conversion d'un langage à l'autre n'est disponible, le changement de langage implique donc la perte de toutes les instructions du POU. Par conséquent, un message de mise en garde s'affiche lors d'une demande de changement de langage, le changement ne sera effectif qu'après une confirmation de l'utilisateur.

Lorsqu'un nouveau langage est choisi, la page "Instructions" est aussitôt sélectionnée.

### 2.3.1.5.5 Page Tâche

Cette page permet d'affecter les paramètres associés à une [tâche](#) <sup>36</sup>.

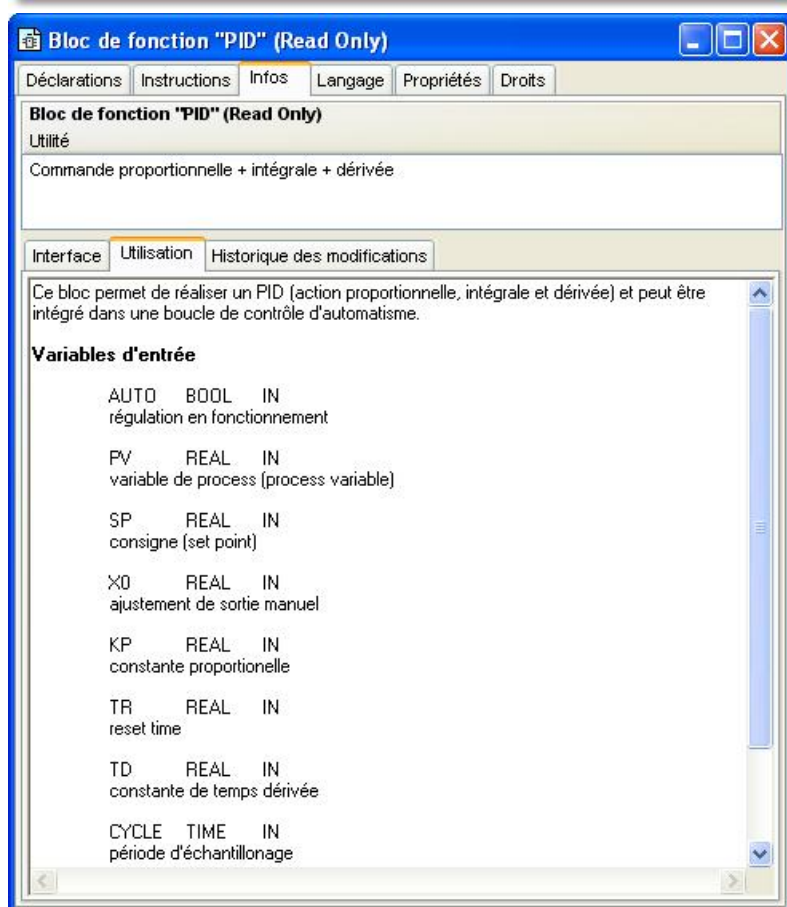
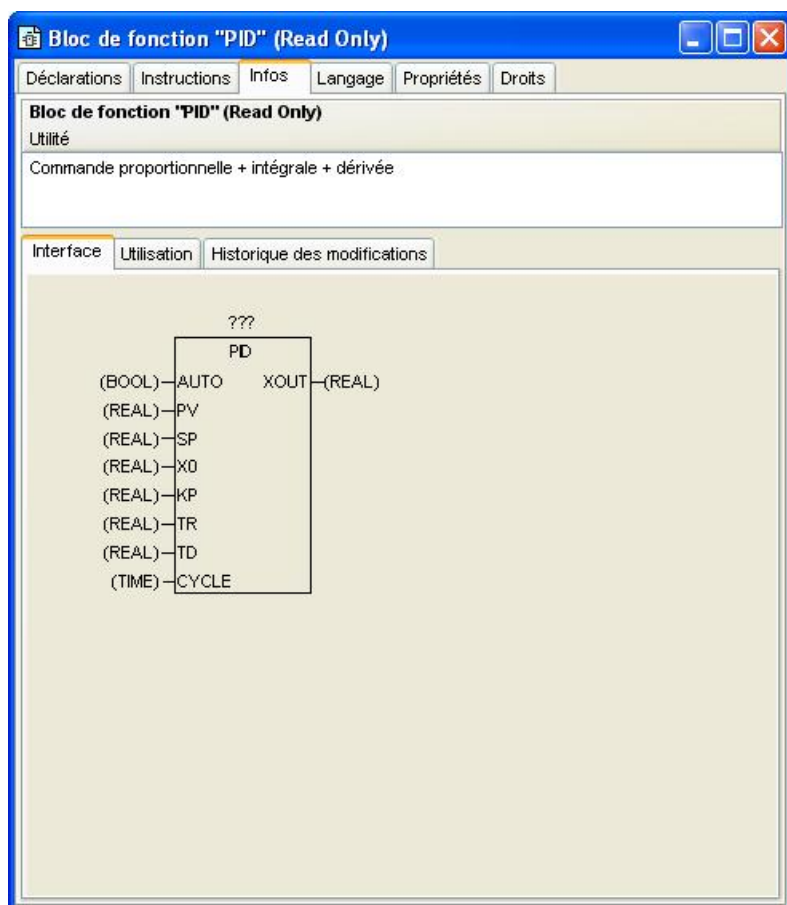


- le niveau de priorité pour l'exécution des programmes ou blocs de fonction
- le trigger
- la période (utiliser la syntaxe de la norme IEC 61131-3 pour les constantes de type Time)
- la liste des programmes ou blocs de fonction à gérer, présentée sous forme d'un tableau

De plus, cette page permet de préciser les phases d'exécution admises.



## 2.3.1.5.6 Page Infos



La première information comporte le nom et le type du POU sélectionné.

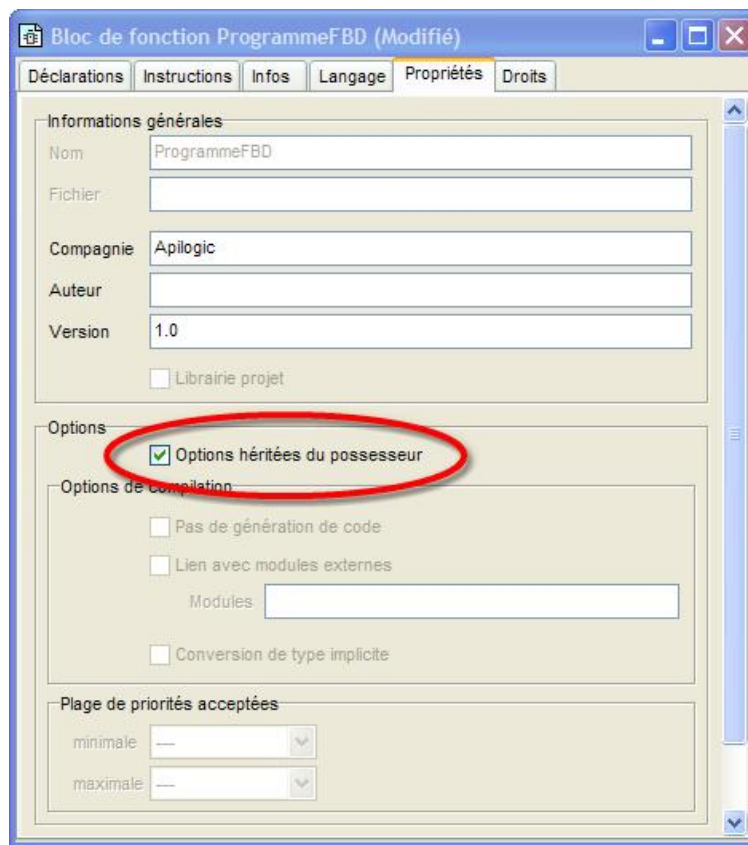
Une zone d'édition permet d'en préciser l'utilité générale.

La page à onglet située en bas de la page permet d'afficher l'interface générale du POU s'il s'agit d'un bloc de fonction ou de programme ou d'une fonction. L'onglet suivant permet d'entrer un descriptif de l'utilisation du POU. Enfin la dernière page est dédiée à l'historique des modifications.

### 2.3.1.5.7 Page Propriétés

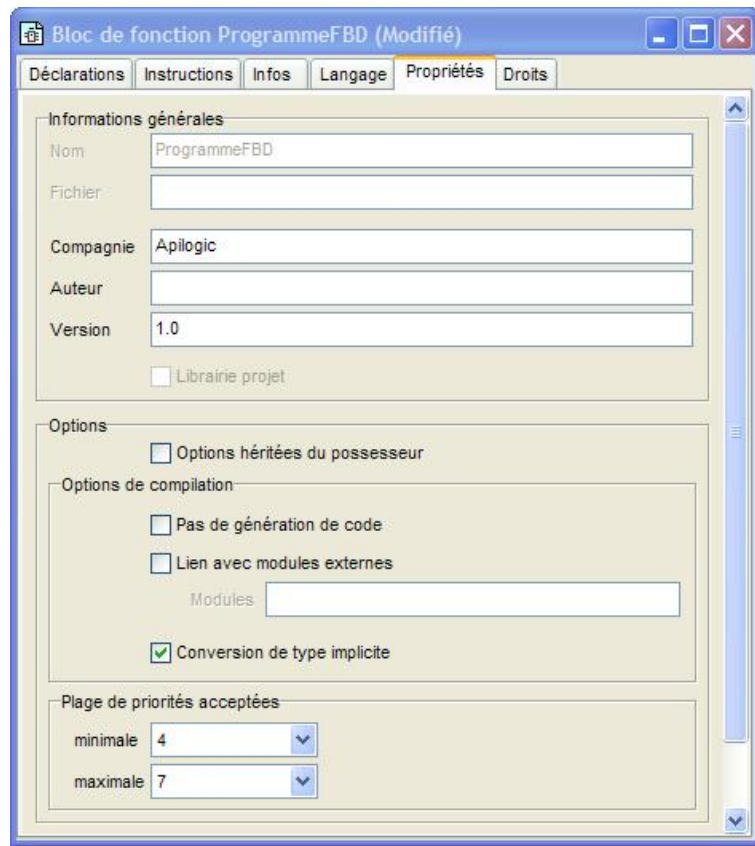
Une partie "Informations générales" permet d'afficher le nom du POU et quelques informations de traçabilité concernant sa conception (auteur, version).

Dans le cas d'une librairie, le nom du fichier correspondant est mentionné, et une case à cocher permet d'indiquer s'il s'agit d'une [librairie "projet"](#) ou non.



Une partie "Options" permet de définir certaines options de compilation et les niveaux de priorités acceptés, par défaut ces options sont héritées du possesseur, comme cela est montré sur la copie d'écran ci-dessus.

Le fait d'hériter les options permet de modifier facilement et simultanément les options concernant l'ensemble des POUs d'une librairie ou d'un sous-groupe. Cependant il est possible de définir les options POU par POU, en décochant la case "options héritées".



Dans ce cas les options suivantes sont disponibles :

- Pas de génération de code

Cette option permet de désactiver toute génération de code pour ce POU, qui est donc simplement ignoré par l'atelier lors de la compilation.

- Lien avec modules externes

Cette option permet l'utilisation par un POU des outils fournis par une librairie de code extérieure, non écrite en langage IEC 61131-3. Cette possibilité est surtout exploitée dans les librairies standard d'Optima PLC.

- Conversion de type implicite

Par défaut, le compilateur interne d'Optima PLC se conforme aux règles volontairement très strictes de contrôle de types du standard IEC 61131-3. Il n'est par exemple pas possible d'affecter un INT à un DINT sans effectuer explicitement une conversion du type INT vers DINT avant l'affectation.

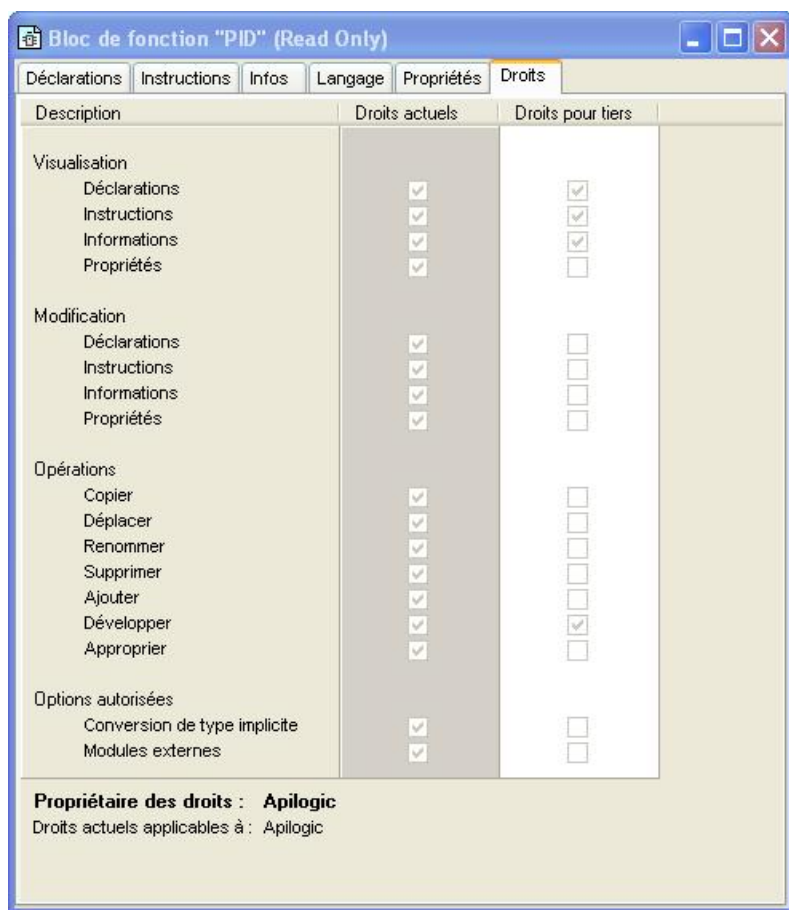
Cependant, il est possible parfois d'alléger ces contraintes sans risque particulier, comme dans le cas cité ci-dessus ; c'est ce que permet de faire cette option, qui, si elle est cochée, réduit la nécessité de procéder explicitement à des conversions de type.

- Plage de priorités acceptées

Certains POU ne peuvent pas être utilisés à tous les degrés de priorité du système. Par exemple, il est aisé de comprendre qu'un bloc réalisant des accès à un fichier sur disque ne puisse être exécuté aux priorités "temps réel".

La plage de priorités acceptées permet de définir les priorités auxquelles le système accepte l'exécution du POU.

## 2.3.1.5.8 Page Droits



Les droits d'accès aux POU sont gérés dans cette page.

Les droits sont classés suivant 4 critères principaux : visualisation, modification, opérations et options autorisées.

Pour la visualisation et la modification, les droits viennent s'appliquer aux propriétés, informations, déclarations et instructions.

Les droits d'opérations concernent les actions qui s'appliquent sur l'intégralité du POU (ajout, suppression...).

Les droits sur les options autorisées viennent limiter les possibilités de compilation.

La page se présente sous la forme d'un tableau qui précise quels sont les droits actuels (c'est-à-dire ceux de l'utilisateur actuel) et les droits autorisés pour un tiers. Il est possible dans ces conditions de conserver la propriété d'un POU tout en permettant à un autre utilisateur de l'utiliser dans ses bibliothèques en lui restreignant les possibilités de consultation ou de modification.

## 2.3.1.5.9 Tableau des pages visibles en fonction des POU

Type/ Page	Infos	Propriétés	Droits	Tâche	Déclarations	Instructions	Type fonction
Tâche	X	X	X	X			
Variables globales	X	X	X		X		
Types	X	X	X		X		
Bloc de programme	X	X	X		X	X	
Bloc de fonction	X	X	X		X	X	

<b>Fonction</b>	X	X	X		X	X	X
<b>Sous-groupe</b>	X	X	X				

### 2.3.1.6 Librairies Optima PLC

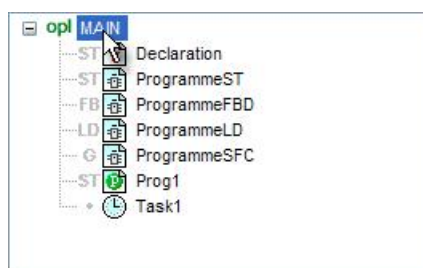
Le progiciel Optima PLC comprend des librairies par défaut, composée uniquement de fonction ou de blocs de fonctions. En voici un aperçu, une description exhaustive de chaque librairie étant disponible dans cette documentation.

<b>SYSTEM</b>	Outils systèmes. Donne la possibilité de quitter ou de générer une exception dans le runtime et également d'insérer des informations dans le fichier d'événements d'Optima PLC.
<b>STANDARD</b>	Blocs et fonctions standards de la norme IEC 61131-3.
<b>MESSAGES</b>	Affichage de messages d'information ou d'erreur
<b>IODRIVERS</b>	Gestion des drivers d'entrées/sorties
<b>TEXTOUT</b>	Gestion des fichiers de sortie texte d'Optima PLC
<b>FILES</b>	Gestion des fichiers
<b>SERIALCOM</b>	Gestion des ports séries
<b>JACKS</b>	Pilotage de vérins simple ou double effet
<b>TABSAMPLES</b>	Utilisation de tableaux de mesures
<b>TCP</b>	Communication en TCP

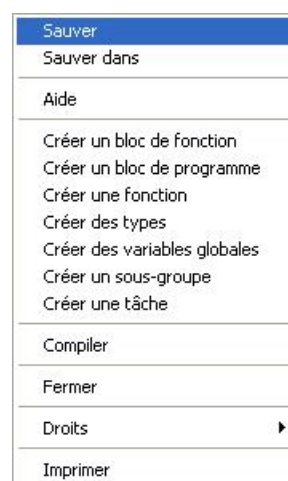
## 2.3.2 Création d'une librairie

### 2.3.2.1 Menu de gestion des librairies

Pour faire apparaître ce menu, sélectionner le dossier librairie dans l'arborescence (1) puis cliquer sur le bouton droit de la souris (pour faire apparaître le menu popup associé aux librairies) (2).



(1)

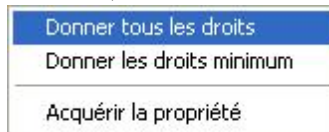


(2)

Le menu se compose de 7 parties :

- Pour sauver ou renommer une librairie (Sauver et Sauver dans)
- Pour visualiser l'aide en ligne

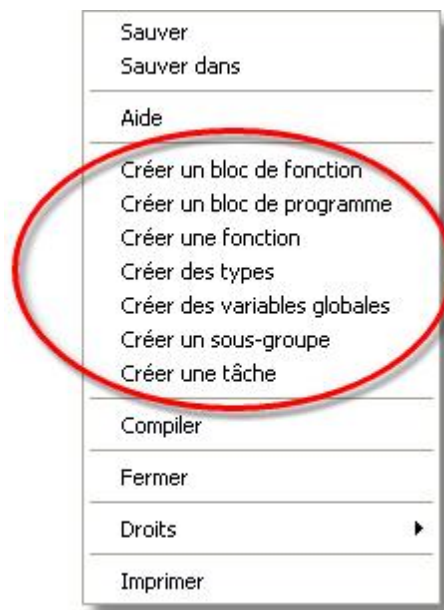
- Pour créer un POU
- Pour compiler la librairie sélectionnée
- Pour fermer la librairie sélectionnée
- Pour affecter des droits d'accès maximum ou minimum aux tiers sur le POU et ceux qu'il contient (si vous êtes le détenteur actuel des droits sur le POU).



- Pour imprimer le dossier de programmation

### 2.3.2.2 Créer un POU

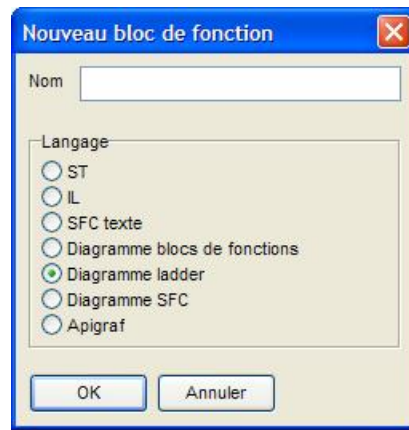
Le menu associé aux librairies permet entre autres de créer un POU parmi les 7 types suivants :



- Bloc de fonction
- Bloc de programme
- Fonction
- Types
- Variables globales
- Sous-groupe
- Tâche

Si le POU à créer est du type "[Variables globale](#)<sup>[38]</sup>s", "[Types](#)<sup>[38]</sup>" ou [sous-groupe](#)<sup>[39]</sup> alors l'icône associée à ce type est immédiatement créée dans l'arborescence, le curseur est positionné sur le nom du POU afin qu'il soit nommé (par défaut, il est nommé "Nouveau").

Si le POU est d'un autre type alors une fenêtre de dialogue apparaît pour d'une part nommer le POU, et d'autre part choisir le langage de programmation de ce POU.



L'icône associée à ce nouveau POU n'est créée dans l'arborescence que si un nom a effectivement été donné à ce POU et si la boîte de dialogue a été validée par le bouton "OK".

### 2.3.2.3 Menu de gestion des POUs

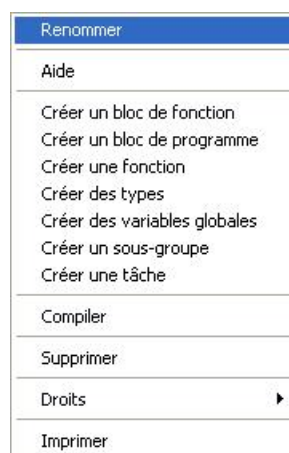
Le menu popup associé au POU sélectionné est fonction du type du POU cependant dans tous les cas, les options suivantes sont disponibles :

- Renommer
- Afficher l'aide contextuelle
- Compiler
- Supprimer
- Affecter les droits
- Imprimer le POU



Des options de menu supplémentaires sont disponibles lorsque le POU sélectionné est :

- Un sous-groupe : il est alors possible de créer un nouveau POU



- Un bloc de fonction : il est alors possible de créer une action ; dans la version actuelle cette option n'est disponible que lorsque le langage de programmation est de type SFC diagramme



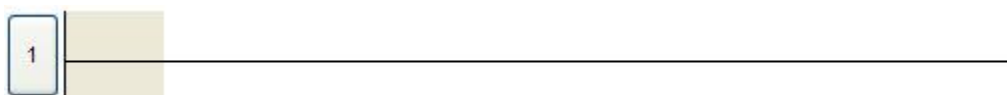
Remarque : certaines options peuvent être inhibées en fonction des droits de l'utilisateur.

### 2.3.2.4 Création de diagrammes Ladder

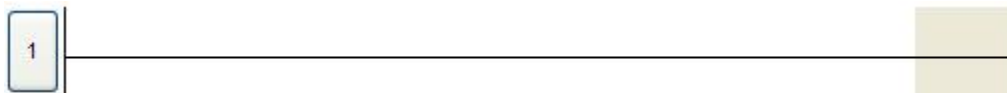
La partie basse de la page à onglet " Instructions " contient les diagrammes d'instructions. Une colonne d'en-tête indique le numéro de diagramme (ou numéro de réseau). Lorsqu'un nouveau diagramme est créé, il apparaît sous la forme d'une ligne cependant il est composé de 3 parties : un élément de début de réseau (sélectionnable), un élément de fin de réseau (sélectionnable) et un lien entre ces 2 éléments.



Pour modifier l'équation du réseau, l'élément de début de réseau ou un contact doit être sélectionné.



Pour modifier la sortie, l'élément de fin de réseau ou une bobine doit être sélectionné.



Chaque bobine ou contact est nommé, ce nom correspond à la variable associée à l'élément.

Pour sélectionner plusieurs éléments en même temps, appuyer sur la touche Control du clavier en même temps que la sélection effectuée avec la souris (par simple clic). Il est possible également de sélectionner les éléments contenus dans une zone de l'affichage en étirant un rectangle englobant de sélection (cliquer un coin du rectangle en laissant le bouton de la souris enfoncé, déplacer la souris jusqu'au coin opposé, et relâcher).

La barre d'outil disponible en haut de la page "Instructions" permet d'ajouter des éléments ou de modifier les caractéristiques de l'élément sélectionné.

Un menu popup est associé à chaque élément sélectionné, les options qu'il propose sont différentes en fonction du type d'élément.

#### 2.3.2.4.1 Barre d'outil des diagrammes ladder

La barre d'outil comprend 6 parties :

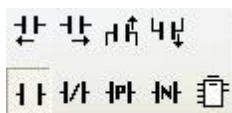


- Gestion des contacts
- Gestion des sorties
- Négation
- Gestion des diagrammes
- Opérations diverses
- Zoom et impression (dans un fichier WMF ou vers une imprimante)



### 2.3.2.4.1.1 Gestion des contacts

Neuf boutons permettent d'ajouter/modifier les contacts :



Les cinq boutons inférieurs peuvent être utilisés de deux manières différentes : si un contact est sélectionné, cliquer un de ces boutons transforme le contact selon l'icône du bouton ; et si une opération de glisser/déposer est initiée à partir du bouton (cliquer, laisser le bouton de la souris enfoncé, puis relâcher sur un contact déjà existant), un nouveau contact correspondant à l'icône est inséré.












	permet d'insérer un contact avant le contact ou les contacts sélectionnés (ou en début de ligne si aucun contact n'est présent dans le diagramme)
	permet d'insérer un contact après le contact ou les contacts sélectionnés (ou en début de ligne si aucun contact n'est présent dans le diagramme)
	permet d'insérer un contact en parallèle et au-dessus du contact ou des contacts sélectionnés
	permet d'insérer un contact en parallèle et au-dessous du contact ou des contacts sélectionnés
	<ul style="list-style-type: none"> <li>• permet de transformer le type du contact sélectionné en contact négatif.</li> <li>• un glisser/déposer avec ce bouton permet d'insérer un contact négatif à gauche de l'élément sur lequel il est lâché.</li> </ul>
	<ul style="list-style-type: none"> <li>• permet de transformer le type du contact sélectionné en contact normal.</li> <li>• un glisser/déposer avec ce bouton permet d'insérer un contact normal à gauche de l'élément sur lequel il est lâché.</li> </ul>
	<ul style="list-style-type: none"> <li>• permet de transformer le type du contact sélectionné en contact actif sur front montant.</li> <li>• un glisser/déposer avec ce bouton permet d'insérer un contact actif sur front montant à gauche de l'élément sur lequel il est lâché.</li> </ul>
	<ul style="list-style-type: none"> <li>• permet de transformer le type du contact sélectionné en contact actif sur front descendant.</li> <li>• un glisser/déposer avec ce bouton permet d'insérer un contact actif sur front descendant à gauche de l'élément sur lequel il est lâché.</li> </ul>
	<ul style="list-style-type: none"> <li>• permet de transformer le contact sélectionné en bloc de fonction, par défaut le raccordement se fait sur la première des entrées et la première des sorties du bloc.</li> <li>• un glisser/déposer avec ce bouton permet d'insérer un bloc par défaut &amp; à gauche de l'élément sur lequel il est lâché.</li> </ul>

### 2.3.2.4.1.2 Gestion des sorties


Les boutons suivants permettent d'ajouter/modifier les sorties :



Les boutons inférieurs peuvent être utilisés de deux manières différentes : si une bobine est sélectionnée, cliquer un de ces boutons transforme la bobine selon l'icône du bouton ; et si une opération de glisser/déposer est initiée à partir du bouton (cliquer, laisser le bouton de la souris enfoncé, puis relâcher sur une bobine déjà existante), une nouvelle bobine correspondant à l'icône est insérée.



	permet d'insérer une bobine en parallèle et au-dessus de celle(s) sélectionnée(s)
	permet d'insérer une bobine en parallèle et au-dessous de celle(s) sélectionnée(s)
	<ul style="list-style-type: none"> <li>• permet d'insérer une bobine dont l'état de la variable associée est le résultat de l'équation de contacts qui la précède.</li> <li>• Un glisser/déposer avec ce bouton sur une bobine du diagramme insère ce type de bobine juste avant.</li> </ul>
	<ul style="list-style-type: none"> <li>• permet d'insérer une bobine dont l'état de la variable associée est l'opposé du résultat de l'équation de contacts qui la précède.</li> <li>• Un glisser/déposer avec ce bouton sur une bobine du diagramme insère ce type de bobine juste avant.</li> </ul>
	<ul style="list-style-type: none"> <li>• permet d'insérer une bobine, l'état de la variable associée devient " vrai " (état mémorisé) lorsqu'il y a un front montant sur le résultat de l'équation de contacts qui la précède.</li> <li>• Un glisser/déposer avec ce bouton sur une bobine du diagramme insère ce type de bobine juste avant.</li> </ul>
	<ul style="list-style-type: none"> <li>• permet d'insérer une bobine dont l'état de la variable associée devient " vrai " (état mémorisé) lorsqu'il y a un front descendant sur le résultat de l'équation de contacts qui la précède.</li> <li>• Un glisser/déposer avec ce bouton sur une bobine du diagramme insère ce type de bobine juste avant.</li> </ul>
	<ul style="list-style-type: none"> <li>• permet d'insérer une bobine dont l'état de la variable associée devient " vrai " (état mémorisé) lorsque le résultat de l'équation de contacts qui la précède est vrai.</li> <li>• Un glisser/déposer avec ce bouton sur une bobine du diagramme insère ce type de bobine juste avant.</li> </ul>
	<ul style="list-style-type: none"> <li>• permet d'insérer une bobine dont l'état de la variable associée devient " faux " (état mémorisé) lorsque le résultat de l'équation de contacts qui la précède est vrai.</li> <li>• Un glisser/déposer avec ce bouton sur une bobine du diagramme insère ce type de bobine juste avant.</li> </ul>
	<ul style="list-style-type: none"> <li>• permet de transformer la bobine sélectionnée en bloc de fonction, par défaut le raccordement se fait sur l'entrée et la sortie de validation du bloc.</li> <li>• Un glisser/déposer avec ce bouton sur une bobine du diagramme insère ce type de bobine juste avant.</li> </ul>
	<ul style="list-style-type: none"> <li>• permet de transformer la bobine sélectionnée en bloc de "retour" (sortie du bloc)</li> <li>• Un glisser/déposer avec ce bouton sur une bobine du diagramme insère un retour juste avant.</li> </ul>
	<ul style="list-style-type: none"> <li>• permet de transformer la bobine sélectionnée en bloc de saut ("goto") vers un label défini dans le présent bloc.</li> <li>• Un glisser/déposer avec ce bouton sur une bobine du diagramme insère un saut juste avant.</li> </ul>

### 2.3.2.4.1.3 Négation

Lorsque l'entrée ou la sortie d'un bloc de fonction est sélectionnée, il est possible de lui affecter une négation grâce au bouton .






### 2.3.2.4.1.4 Gestion des diagrammes

Il est possible de définir plusieurs diagrammes dans un POU :

-  permet d'ajouter un diagramme avant celui qui est sélectionné
-  permet d'ajouter un diagramme après celui qui est sélectionné



### 2.3.2.4.1.5 Opérations diverses

Les boutons placés à la fin de la barre d'outils permettent de :



-  permet de faire un zoom sur le bloc de fonction sélectionné c'est-à-dire d'en visualiser les instructions lorsque les droits d'utilisateur le permettent.
-  permet d'annuler la dernière modification effectuée dans les diagrammes.
-    : ces trois icônes permettent respectivement de couper, copier et coller la sélection.

### 2.3.2.4.1.6 Zoom et impression

Il est possible de faire un zoom avant ou arrière pour mieux visualiser les diagrammes :

-  : rapetisse l'affichage du diagramme
-  : agrandit l'affichage du diagramme

Les instructions peuvent être enregistrées dans un fichier ou imprimées :

-  : enregistrement dans un fichier WMF
-  : impression des diagrammes

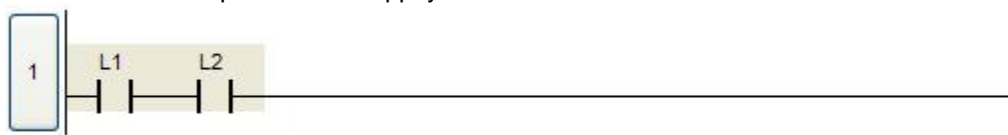
### 2.3.2.4.2 Utilisation de la sélection


L'insertion de contact en parallèle avec d'autres contacts (fonction "ou") nécessite que ces autres contacts soient auparavant sélectionnés.

Exemple : réaliser un diagramme testant la condition (L1 & L2) or L3.  
Initialement les contacts L1 et L2 ont été insérés comme suit.



Sélectionner L1 et L2 en les cliquant tout en appuyant sur la touche Control






Cliquer sur le bouton  : un nouveau contact est inséré en parallèle avec la sélection. Il reste à frapper le nom de la variable associée.



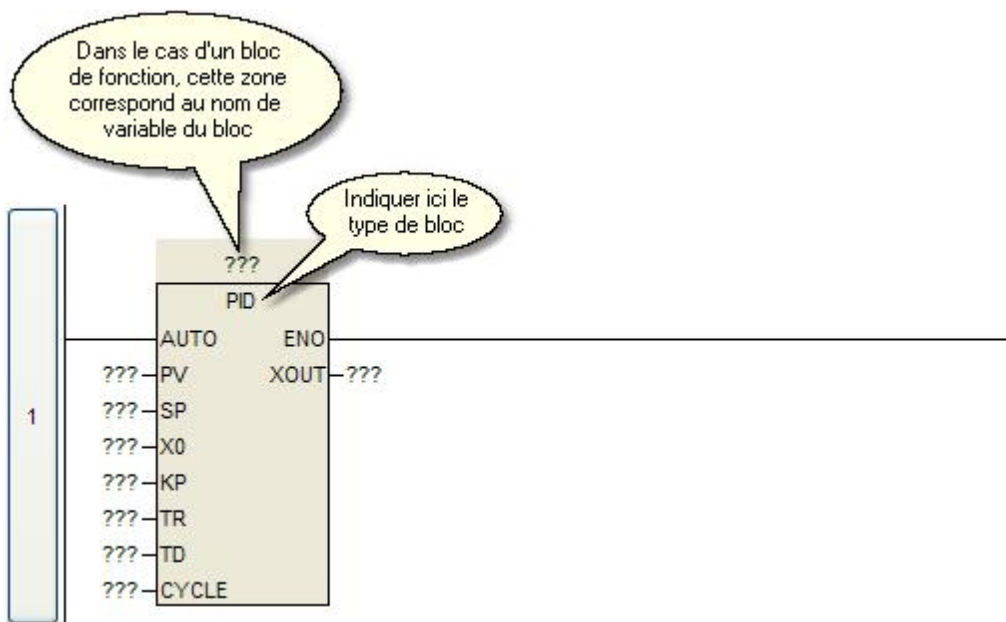
### 2.3.2.4.3 Utilisation de blocs dans un diagramme ladder

Des blocs (fonctions et blocs de fonction) peuvent être utilisés dans un diagramme ladder, en respectant certaines règles.

#### 1ère méthode : utilisation des icônes

Pour insérer un bloc en contact, insérer tout d'abord un contact normal, avec par exemple le bouton , puis transformer ce contact en bloc en cliquant sur le bouton  (il est aussi possible de faire directement glisser le bouton  sur la position d'insertion du bloc). Enfin, entrez le type de bloc souhaité.

Dans l'exemple suivant, un bloc par défaut & est inséré, puis remplacé par un PID (ce type est frappé à la place de &).



Dans le cas d'une fonction, seul le type de bloc doit être saisi. S'il s'agit d'un bloc de fonction, une zone de saisie supplémentaire située au dessus du bloc permet de saisir le nom de la variable correspondante, qui doit être déclarée avec ce type de bloc.

Depuis la version 2.9 d'Optima PLC, la déclaration de la variable correspondant à un bloc de fonction n'est plus obligatoire. Celle-ci est faite automatiquement avec un nom par défaut.

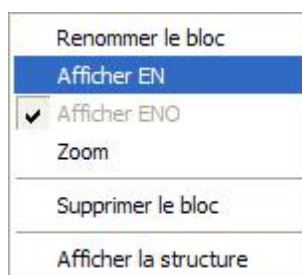
#### 2ème méthode : utilisation de l'arbre des POU

Cliquer sur l'onglet Bibliothèques, et développer la partie de l'arbre depuis laquelle vous souhaitez insérer un bloc. Cliquer sur le bloc en maintenant le bouton de la souris enfoncé, puis le relâcher à l'emplacement d'insertion souhaité. De cette manière, un bloc PID par exemple peut être directement inséré depuis la bibliothèque Standard.

#### Usage de EN et ENO (Enable, EnableOut)

Lors de l'insertion d'un bloc, la première entrée et la première sortie sont normalement connectées au réseau ladder. Cependant, ces entrées et sorties ne sont pas forcément de type BOOL, ainsi que le réseau ladder le voudrait. Dans ce cas, une entrée supplémentaire EN et/ou une sortie supplémentaire ENO sont ajoutées d'office.

L'utilisation de EN ou ENO peut également être forcée (menu contextuel du bloc) :

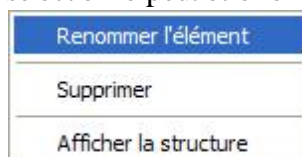


Dans le cas illustré ci-dessus, le bloc PID comporte une première entrée Auto de type BOOL, qui est connectée au réseau ladder. Par contre sa première sortie XOut est de type REAL : une sortie ENO est automatiquement ajoutée.

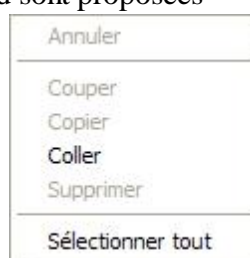
#### 2.3.2.4.4 Menus contextuels des diagrammes ladder

Les menus sont fonction de l'élément sélectionné qui peut être :

- un contact ou une bobine : l'élément sélectionné peut être renommé ou supprimé



- un texte : les options d'édition standard sont proposées



- un bloc, une entrée (raccordée ou non) ou une sortie d'un bloc :



- un diagramme : il est possible d'ajouter un label ou un commentaire à un diagramme, ou de supprimer le diagramme sélectionné



### 2.3.2.4.5 Exemple de création d'un diagramme ladder

L'équation à représenter est la suivante :  $MiseSousPuissance = (Start1 \text{ or } Start2) \text{ and not } AU$

Sélectionner (ou créer) un POU de type Bloc de fonction en langage ladder.


La première étape à effectuer est de créer les variables de l'équation.


Sélectionner l'onglet "Déclarations" et insérer les déclarations suivantes :

```
Var_Input
    Start1      : bool;
    Start2      : bool;
    AU           : bool;
End_Var


Var_Ouput
    MiseSousPuissance : bool;
End_Var
```


Sélectionner l'onglet "Instructions".

Sélectionner l'élément d'entrée du diagramme, cliquer sur  et nommer le contact "Start1".

Sélectionner le contact Start1, cliquer sur  et nommer le contact "AU".

Sélectionner le contact AU puis cliquer sur .

Sélectionner le contact Start1, cliquer sur  et nommer le contact "Start2".

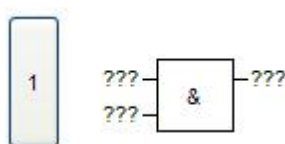
Sélectionner l'élément de sortie du diagramme, cliquer sur  et nommer la bobine "MiseSousPuissance".



Le diagramme est complet. Pour vérifier qu'aucune erreur n'a été commise pendant la création, il suffit alors de compiler.

### 2.3.2.5 Création de diagrammes FBD

La partie basse de la page à onglet "Instructions" contient les diagrammes d'instructions. Une colonne d'en-tête indique le numéro de diagramme. Lorsqu'un nouveau diagramme est créé, il est composé d'un seul bloc (de type ET).



Il est dès lors possible de modifier ce premier bloc ou d'ajouter des blocs en amont et en aval (connectés aux entrées et sorties du bloc) dans le diagramme.

Pour sélectionner plusieurs blocs en même temps, appuyer sur la touche Control du clavier en même temps que la

sélection effectuée avec la souris (par simple clic).








La barre d'outil disponible en haut de la page "Instructions" permet d'ajouter des diagrammes, d'appliquer une négation sur une entrée d'un bloc et de visualiser les instructions internes au bloc sélectionné.

Un menu popup est associé à chaque élément sélectionné, les options qu'il propose sont différentes en fonction du type d'élément.

### 2.3.2.5.1 Barre d'outil d'un diagramme Blocs de fonction

Les boutons de la barre d'outils sont actifs en fonction de l'élément sélectionné :

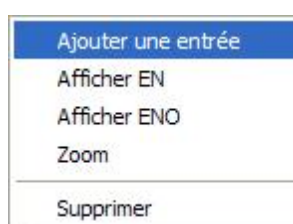


	lorsqu'un diagramme ou l'un de ses éléments (bloc, entrée ou sortie d'un bloc) est sélectionné, il est possible d'ajouter un diagramme avant le diagramme concerné par la sélection
	lorsqu'un diagramme ou l'un de ses éléments (bloc, entrée ou sortie d'un bloc) est sélectionné, il est possible d'ajouter un diagramme après le diagramme concerné par la sélection
	lorsqu'un bloc est sélectionné, il est possible d'en visualiser les instructions internes
	lorsqu'une entrée ou une sortie de bloc est sélectionnée, il est possible d'appliquer une négation à celle-ci
	ce bouton permet d'annuler la dernière modification effectuée dans le diagramme
	copie la sélection dans le presse-papier et la supprime du diagramme
	copie la sélection dans le presse-papier
	colle le presse-papier avant la sélection actuelle
	rapetisse l'affichage du diagramme
	agrandit l'affichage du diagramme
	enregistrement du diagramme dans un fichier WMF
	impression du diagramme

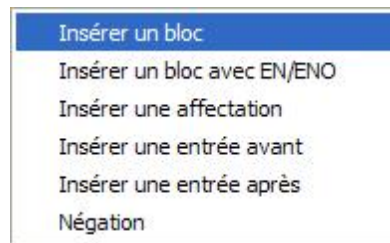
### 2.3.2.5.2 Menus contextuels d'un diagramme blocs de fonction

Les menus sont fonction de l'élément sélectionné qui peut être :

- un bloc : pour afficher les entrée/sortie de validation du bloc (EN et ENO dans la norme IEC 61131-3), faire un zoom sur les instructions internes au bloc, supprimer le bloc, lui ajouter une entrée si le nombre d'entrées du bloc est variable (exemple : un ET peut être effectué sur 2 entrées, 3 ou plus), renommer le bloc lorsqu'il a un nom (exemple : PID).



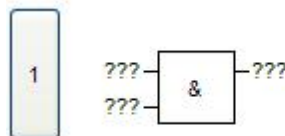
- une entrée de bloc : pour ajouter un bloc (avec ou sans les entrée/sortie EN et ENO), insérer une branche de dérivation (affectation), lui appliquer une négation ou, dans le cas où le nombre d'entrées est variable, ajouter une entrée ou supprimer celle qui est sélectionnée.



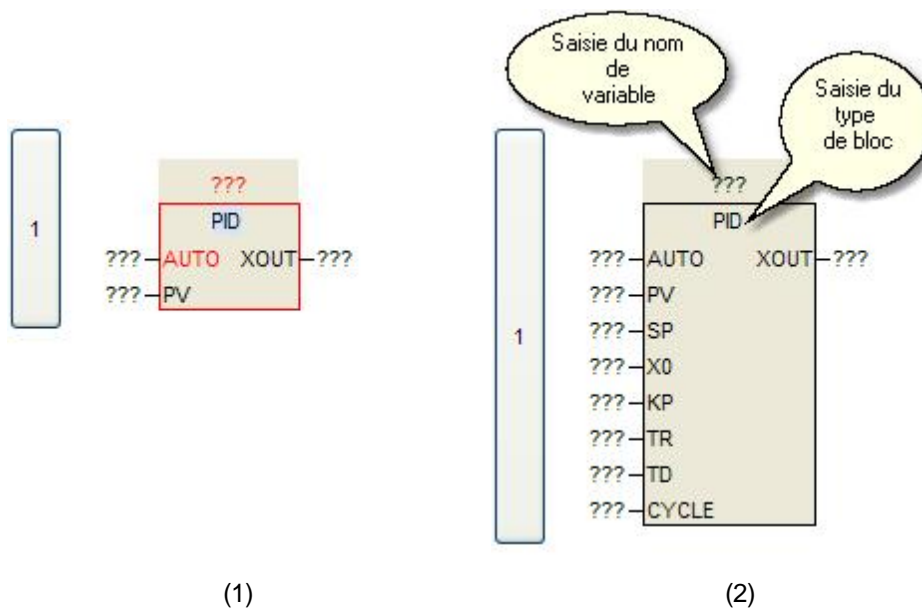
- une sortie de bloc ou une sortie d'une branche de dérivation: pour ajouter un bloc (avec ou sans les entrée/sortie EN et ENO), insérer une branche de dérivation (affectation) ou lui appliquer une négation.
- une branche de dérivation : pour ajouter une sortie sur cette dérivation.
- un diagramme : il est possible d'ajouter un commentaire à un diagramme, ou de supprimer le diagramme sélectionné.
- un texte : les options d'édition standard sont proposées, il est à noter que modifier le texte correspondant à la fonction du bloc entraîne une mise à jour complète du bloc.

### 2.3.2.5.3 Définition du type de bloc

Lors de son insertion, le bloc prend le type par défaut "&" correspondant à la fonction AND.



Pour modifier le type par défaut, éditer le texte correspondant au type de bloc (ici &) en cliquant dessus, puis en tapant le nouveau type souhaité.



- (1) : le nom de bloc PID est frappé, le bloc se met partiellement à jour au cours de la frappe.  
 (2) : le nom est validé, le bloc se met à jour (ajout ou suppression d'entrées/sorties en fonction de son interface).

Si le bloc est un bloc de fonction ou de programme, il faut en plus entrer le nom de la variable correspondante dans



la zone de saisie qui se trouve juste au-dessus. La variable correspondante doit être définie dans la partie déclarations, ou dans une déclaration de variables globales. Depuis la version 2.9, cette déclaration n'est plus obligatoire : laisser alors cette zone de saisie vide, la déclaration étant effectuée automatiquement avec un nom par défaut.

Si le bloc est une fonction, cette zone de saisie située au dessus du bloc n'existe pas.

#### 2.3.2.5.4 Utilisation de variables en entrée et en sortie de bloc

Une variable peut être utilisée pour chaque entrée ou sortie de bloc. Cette variable doit être du même type que l'entrée ou la sortie.

Pour les entrées, il est également possible d'utiliser une expression ST dont le résultat correspond au type de l'entrée.

Pour définir une variable, cliquer sur la zone de saisie et entrer le nom souhaité.

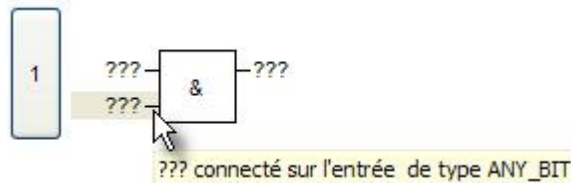
#### 2.3.2.5.5 Chaînage de blocs

Les blocs peuvent être chaînés en respectant les règles suivantes :

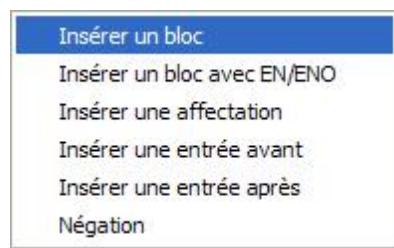
- seule la première sortie d'un bloc peut être chaînée à l'entrée d'un autre bloc.
- toutes les entrées d'un bloc peuvent être chaînées à la première sortie d'un autre bloc.

Pour chaîner un nouveau bloc sur une entrée ou la première sortie d'un bloc, procéder de la manière suivante :

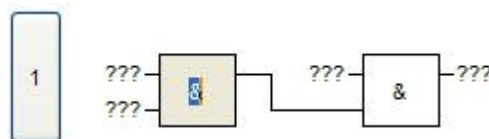
faire un clic droit sur l'entrée ou sortie concernée.



Le menu contextuel suivant s'affiche :



Cliquer sur "insérer un bloc", le bloc par défaut est alors ajouté :



Ce bloc peut être modifié en frappant le type de bloc désiré.

#### 2.3.2.5.6 Exemple de diagrammes FBD

Les équations à représenter sont les suivantes :

DemandeStart = Start1 or Start2

MiseSousPuissance = DemandeStart and not AU

Sélectionner (ou créer) un POU de type Bloc de fonction en langage diagramme Blocs de fonction.

La première étape à effectuer est de créer variables de l'équation.

Sélectionner l'onglet "Déclarations" et insérer les déclarations suivantes :

```

Var_Input
  Start1      : bool;
  Start2      : bool;
  AU          : bool;
End_Var

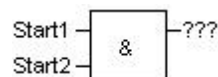
Var_Output
  MiseSousPuissance : bool;
  DemandeStart      : bool;
End_Var

```

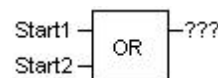
Sélectionner l'onglet "Instructions".

Sélectionner le bloc "&" créé en même temps que le diagramme, cliquer sur une des entrées du bloc et, en utilisant le menu contextuel, affecter la variable "Start1" (en modifiant le texte "???" de l'entrée).

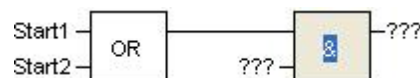
Sélectionner la seconde entrée et affecter la variable "Start2".



Sélectionner le bloc "&" et modifier sa fonction en cliquant sur le texte & et en le remplaçant par "or" .

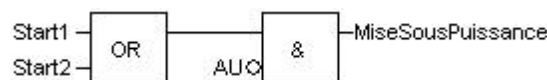


Sélectionner la sortie du bloc et insérer un bloc.



Sélectionner l'entrée de ce second bloc qui n'est pas reliée au précédent et lui affecter une négation (en utilisant le menu contextuel) puis la variable AU.

Sélectionner la sortie du second bloc et lui affecter la variable "MiseSousPuissance".

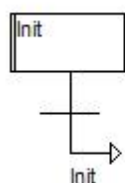


Sélectionner la sortie du premier bloc et à l'aide du menu contextuel "Affectation", ajouter une sortie parallèle pour affecter la variable "DemandeStart".

Le diagramme est complet. Pour vérifier qu'aucune erreur n'a été commise pendant la création, il suffit alors de compiler. Il est dès lors possible d'utiliser le diagramme comme un seul bloc.

### 2.3.2.6 Création de diagrammes SFC (Grafcet)

La partie basse de la page à onglet "Instructions" contient un diagramme d'instructions (et un seul, pour créer un autre diagramme SFC, il faut créer un autre POU).



Lorsqu'un nouveau diagramme est créé, il se compose d'un pas initial (nommé par défaut Init), d'une transition et d'une branche de retour qui renvoie vers le pas initial.

La barre d'outil disponible en haut de la page "Instructions" permet d'ajouter des éléments dans le diagramme.

La suppression d'éléments doit être logique et laisser un diagramme cohérent. Ainsi, si une étape est sélectionnée

seule, elle ne peut pas être supprimée, la suppression n'est possible que si pour chaque étape sélectionnée, il y a au moins une transition sélectionnée en même temps.

La sélection multiple s'effectue en appuyant sur la touche Control du clavier au moment où l'élément est sélectionné avec la souris. Lors de l'insertion d'une divergence en "ET" ou en "OU", la divergence est mise en parallèle avec l'ensemble des étapes sélectionnées.

### Transitions



Les conditions de transition doivent être définies sous la forme d'une condition à la syntaxe ST. L'édition se fait en cliquant à droite du trait de transition.

Une transition peut être laissée vide, dans ce cas elle est considérée comme étant toujours vraie.

Des commentaires peuvent être placés dans une transition, avec la syntaxe classique : //, { } ou ( \* ). Attention cependant, si la transition ne contient qu'un commentaire, elle n'est pas considérée comme étant vraie par défaut, et une erreur de compilation est signalée. Pour éviter cela, mettre la mention True en dehors des commentaires.

### Etapes

L'édition du nom de l'étape se fait en cliquant le contenu de la boîte (en haut à gauche). Par défaut, une nouvelle étape ne comporte pas de nom, celui-ci n'est en effet pas obligatoire, sauf pour les sauts.

Une étape peut avoir une ou plusieurs actions associées, cependant l'ajout d'une action n'est pas obligatoire, une étape sans action est dite "étape vide". L'affichage de l'action peut être masqué/démasqué ou non grâce aux boutons  et  situés dans le rectangle d'étape.

La définition d'une action peut prendre trois formes :





- Action booléenne : une variable booléenne est précisée, l'action consiste alors à mettre à 1 cette variable.
- Action "in situ" : l'action est programmée directement dans le diagramme, elle doit être définie dans l'un des langages textuels.
- Action externe : l'action mentionnée dans le diagramme fait référence à une action définie en dehors du diagramme, sous la dépendance du bloc en cours, et se trouve donc au niveau du dessous dans l'arbre de conception. Ces actions peuvent être définies en n'importe quel langage.











Un menu popup est associé à chaque élément sélectionné, les options qu'il propose sont différentes en fonction du type d'élément.

#### 2.3.2.6.1 Barre d'outil d'un diagramme SFC

Les boutons de la barre d'outils sont actifs en fonction de l'élément sélectionné :



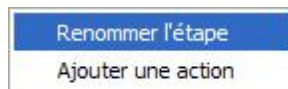
	lorsqu'une étape, une transition ou une branche de retour est sélectionnée, il est possible d'ajouter un ensemble étape/transition avant l'élément sélectionné
	lorsqu'une étape ou une transition est sélectionnée, il est possible d'ajouter un ensemble étape/transition après l'élément sélectionné
	lorsqu'une transition est sélectionnée, il est possible d'ajouter une branche parallèle (fonction OU) à gauche de la transition sélectionnée
	lorsqu'une transition est sélectionnée, il est possible d'ajouter une branche parallèle (fonction OU) à droite de la transition sélectionnée

	lorsqu'une étape est sélectionnée, il est possible d'ajouter une branche parallèle (fonction ET) à gauche de l'étape sélectionnée
	lorsqu'une étape est sélectionnée, il est possible d'ajouter une branche parallèle (fonction ET) à gauche de l'étape sélectionnée
	lorsqu'une transition appartenant à une branche parallèle est sélectionnée, il est possible de modifier le renvoi, c'est-à-dire de le rediriger vers une autre étape
	masque toutes les actions de toutes les étapes
	démasque toutes les actions de toutes les étapes
	agrandit l'affichage du diagramme
	rapetisse l'affichage du diagramme
	imprime le diagramme
	exporte le graphique du diagramme dans le répertoire du projet sous la forme d'un fichier WMF (Windows Meta File) dont le nom est le nom du bloc.
	ce bouton permet d'annuler la dernière modification effectuée dans le diagramme

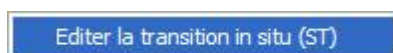
### 2.3.2.6.2 Menus contextuels des diagrammes SFC

Les menus sont fonction de l'élément sélectionné qui peut être :

- une étape : l'étape peut être renommée ou une action peut être ajoutée



- une transition : la condition de passage de la transition peut être éditée "in situ"



- une branche de retour : pour renommer l'étape vers laquelle la branche retourne
- une action : pour paramétrer ou supprimer l'action sélectionnée
- un texte : les options d'édition standard sont proposées

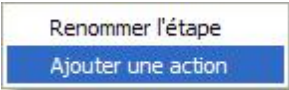
### 2.3.2.6.3 Création d'une action

La création d'une action s'effectue en 2 étapes :

- paramétrage de l'action
- implémentation de l'action

Une action est représentée par un ensemble de 2 ou 3 rectangles placés à droite de l'étape, le premier pour afficher le qualificatif, le second (à sa droite) pour le nom de l'action, et éventuellement un troisième en dessous des 2 premiers pour les instructions définies in situ.

Pour créer une action, faire un clic droit sur l'étape ; le menu contextuel suivant s'affiche :



Cliquer sur Ajouter une action. La fenêtre suivante est affichée :

Paramètres d'action

Qualificateur

N

Temps

Variable d'action booléenne

Nom d'action

Variable indicatrice

Action éditée in situ

☐ En langage ST

☐ En langage IL

☐ En langage Apigraf

☐ Appel d'action inhibé

☐ Simple commentaire

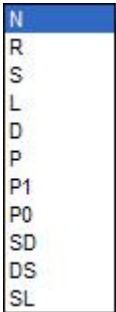
OK

Annuler

Aide

2.3.2.6.3.1 Qualificateur d'action

Le champ "qualificateur" permet de définir comment l'action sera exécutée. Ceci est valable pour tous les types de définition d'action.



Les différents qualificateurs sont définis par la norme IEC 61131-3. Optima PLC vous donne la possibilité de tous les utiliser.

Attention : l'utilisation de certains qualificateurs, en particulier ceux permettant de temporiser les actions, peuvent compliquer le fonctionnement du diagramme SFC et en rendre la mise au point difficile. Les qualificateurs N, R, S, P1 et P0 sont généralement suffisants.

Qualificateur	Usage	Paramètre
N	Non mémorisé	
R	Remise à 0	
S	Positionné (mémorisé)	
L	Limité dans le temps	temps
D	Différé	temps
P	Impulsion	

<b>P1</b>	Impulsion à l'entrée dans l'étape	
<b>P0</b>	Impulsion à la sortie de l'étape	
<b>SD</b>	Mémorisé et différé	temps
<b>DS</b>	Différé et mémorisé	temps
<b>SL</b>	Mémorisé et limité dans le temps	temps

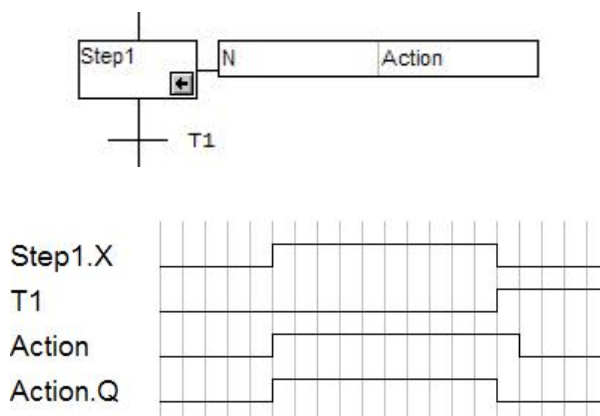
Attention, Optima PLC respecte strictement la définition des qualificateurs de la norme IEC 61131-3. Ceci implique que les actions sont systématiquement exécutées deux fois au minimum : tant que la condition d'exécution est vraie, et une fois encore lorsqu'elle devient fausse. Le but est de permettre à l'action de faire un traitement particulier lors de sa désactivation. L'action dispose d'une variable X indiquant son activité. Cette variable passe à false lorsque l'action est désactivée.

Ceci pose surtout problème avec le qualificateur P (impulsion). L'action ainsi qualifiée est exécutée lors de l'entrée dans l'étape, mais aussi une deuxième fois. La première fois avec la variable d'activité X à true, la seconde fois à false. Une erreur fréquente est de considérer que l'action ne sera exécutée qu'une fois, et d'utiliser ce qualificateur pour effectuer un comptage, par exemple, sans tester la variable X pour vérifier que l'action est bien active.

Pour pallier ce problème, la norme a introduit deux exceptions : les qualificatifs P0 et P1. Les actions utilisant ces qualificatifs sont exécutées une seule fois, et sont donc le moyen idéal de traiter une impulsion à l'entrée ou à la sortie d'une étape.

#### **N : Action Non mémorisée.**

Ce type d'appel d'action est le plus simple : l'action est exécutée tant que l'étape d'appel est active, puis une dernière fois après sa désactivation.



#### **S : Action mémorisée (Set).**

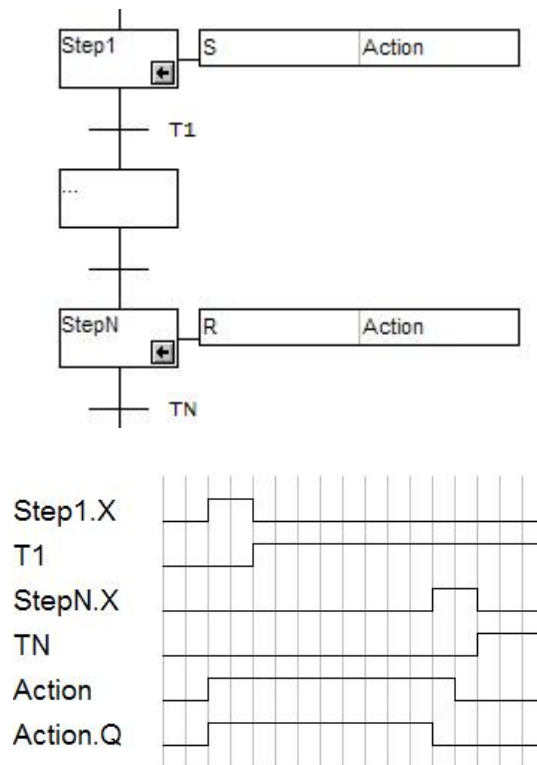
#### **R : Action remise à 0 (Reset).**

Une action appelée avec le qualificateur S est exécutée dès que l'étape devient active, et reste mémorisée. Elle est exécutée jusqu'à un appel ultérieur avec le qualificateur R, même si l'étape d'appel initiale n'est plus active.

Si une action appelée avec le qualificateur S n'est jamais arrêtée avec un appel en R, elle est exécutée indéfiniment à chaque appel du bloc contenant le diagramme SFC.

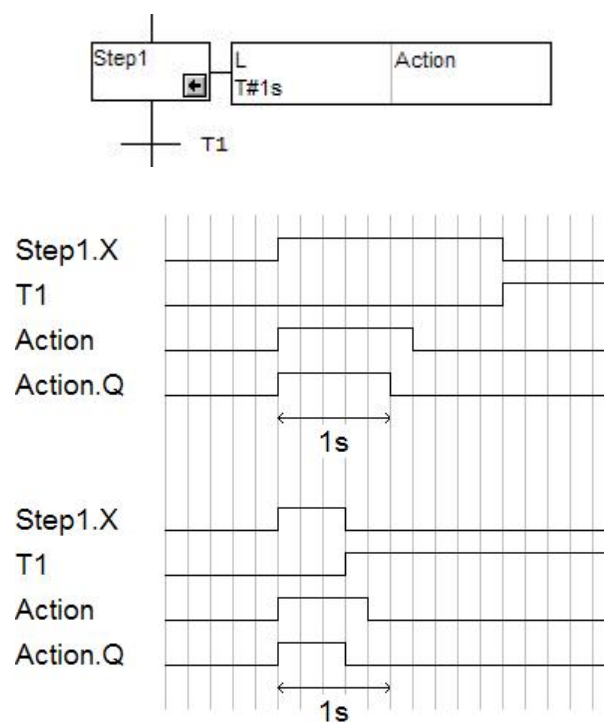
Lorsque l'action est appelée avec le qualificateur R, elle est exécutée une fois encore avec la variable Q à 0.

Le qualificateur R permet d'arrêter une action appelée avec un autre qualificateur que S et mémorisant une action (SD, DS ou SL).



### L : Action Limitée dans le temps

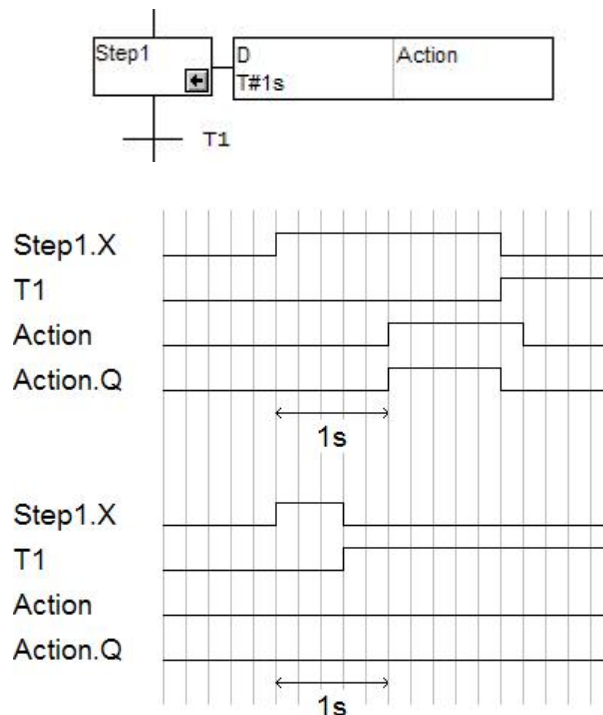
Une action appelée avec le qualificateur L commence son exécution dès que l'étape devient active. Elle continue à s'exécuter pendant un laps de temps précisé. Si l'étape devient inactive avant la fin du délai, l'action est arrêtée.



### D : Action Différée

Une action appelée avec le qualificateur D commence à s'exécuter après l'écoulement d'un délai spécifié. Elle continue tant que l'étape reste active, puis une dernière fois après la désactivation de l'étape.

Si l'étape devient inactive avant l'écoulement du délai, l'action n'est pas exécutée.

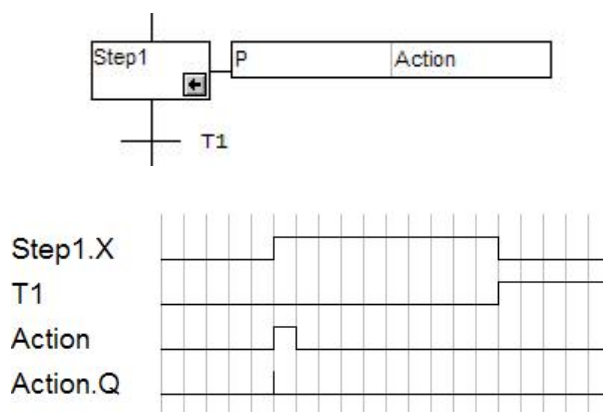


### P : Action "impulsion" (Pulse)

L'action appelée avec le qualificateur P est exécutée une première fois lors de l'activation de l'étape, puis une seconde, avec la variable Q à 0.

Ce type d'appel d'action doit être utilisé avec précaution : une erreur fréquente consiste à l'utiliser en pensant à tort que l'exécution n'a lieu qu'une fois, pour un comptage par exemple : il faut dans ce cas utiliser P1.

Pour effectuer une impulsion sur une variable booléenne (action booléenne), l'appel d'action P est l'idéal : lors de la première exécution, la variable est mise à 1, puis remise à zéro lors de la seconde exécution.



### P1 : impulsion à l'entrée dans l'étape

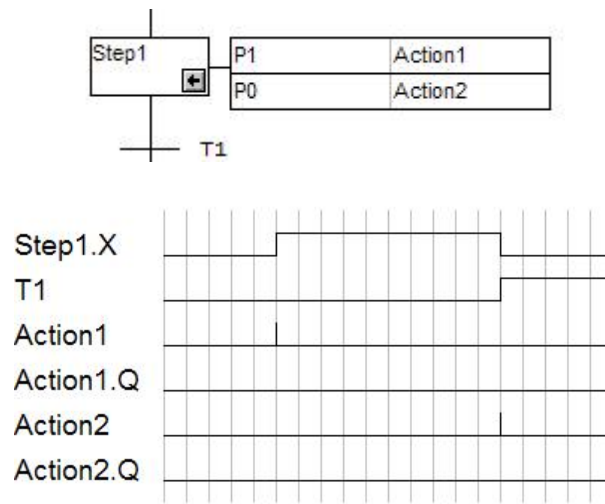
### P0 : impulsion à la sortie de l'étape

Les actions appelées avec ces qualificatifs sont appelées une fois et une seule, respectivement lors de l'activation et de la désactivation de l'étape.

L'appel n'ayant lieu qu'une fois, la variable Q reste à 0. Cela signifie également que ces qualificatifs ne fonctionnent pas avec les actions booléennes (la variable booléenne recopie l'état de Q).

Ces appels d'action sont particulièrement adaptés aux actions comportant du comptage par exemple, ou tout traitement à exécuter une seule fois. P1 constitue un bon substitut à P.

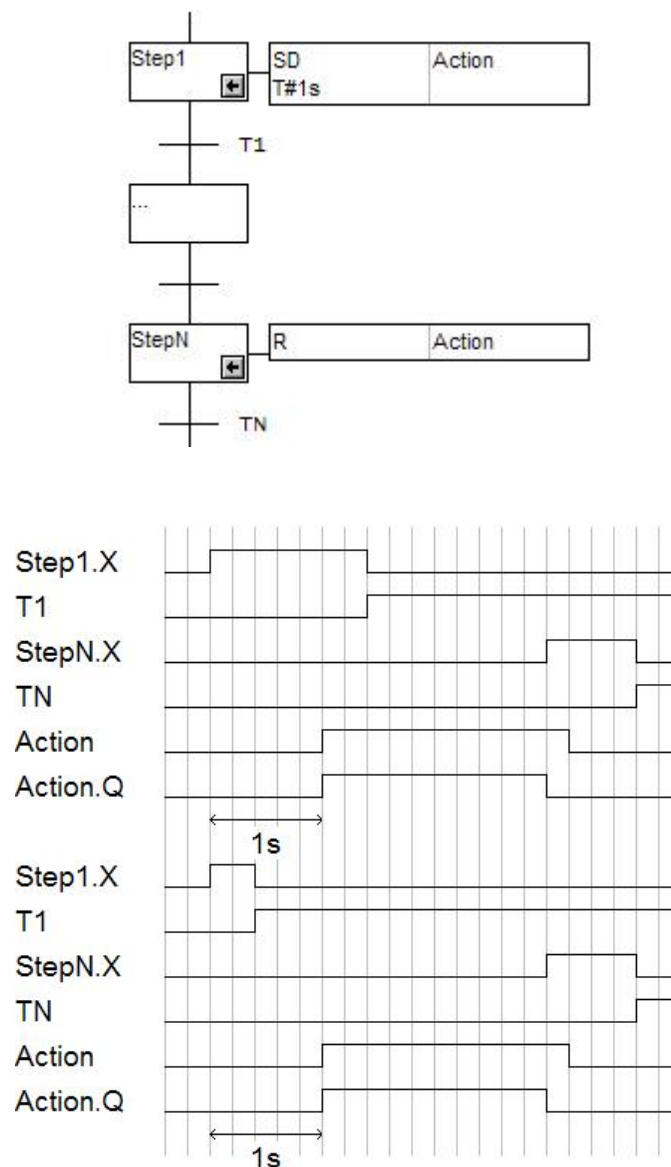




### SD : Action mémorisée et Différée

Avec ce type d'appel, une action est mémorisée et commence à s'exécuter après un délai précisé, même si l'étape n'est plus active.

L'exécution se termine lors d'un appel ultérieur de l'action avec le qualificateur R (Reset). Si l'appel avec R est effectué avant la fin du délai, l'action n'est pas exécutée.

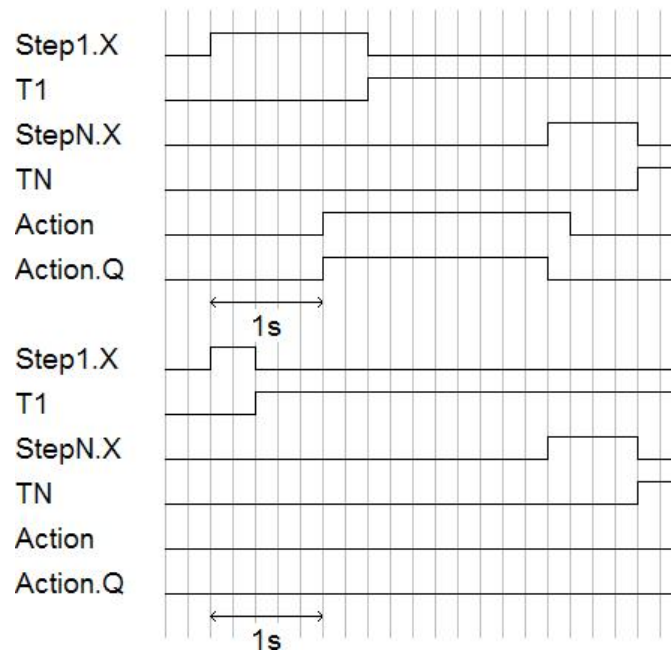
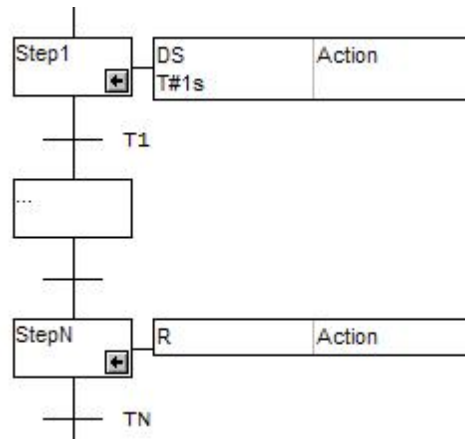


**DS : Action Différée et mémorisée**

L'action appelée avec le qualificateur DS est différée selon un délai précisé. Au bout de ce délai, l'action commence à s'exécuter, et est mémorisée : l'exécution continue même si l'étape est désactivée.

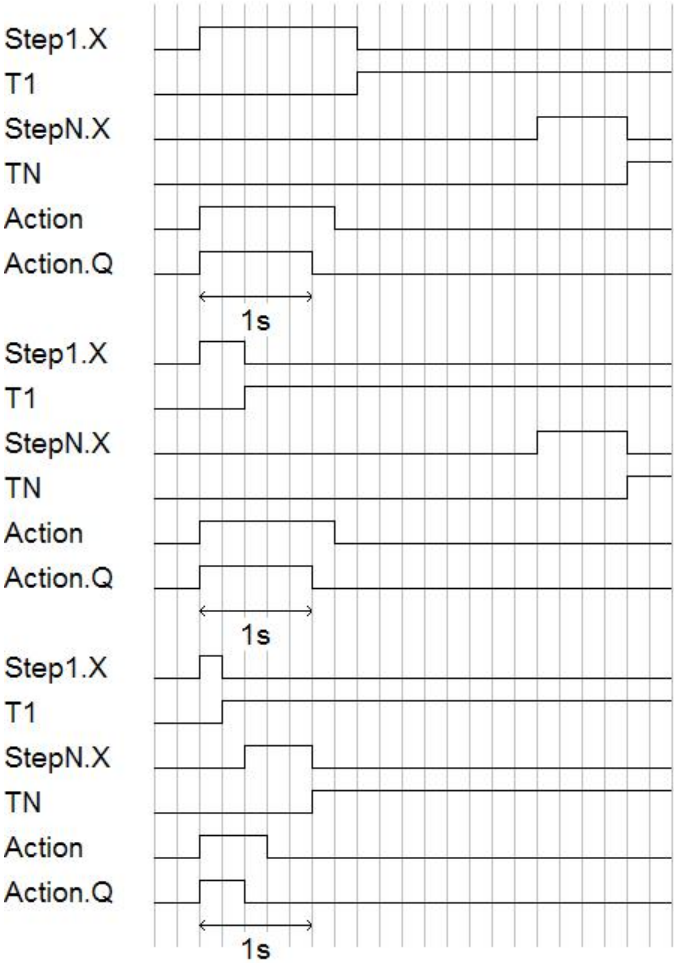
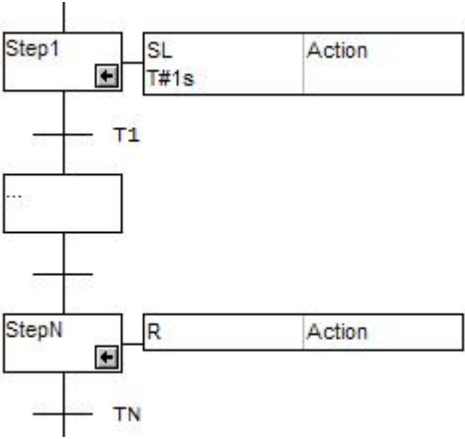
L'exécution se termine lors d'un appel ultérieur de l'action avec le qualificateur R (Reset).

Si l'étape est désactivée avant la fin du délai, l'action n'est pas exécutée et n'est pas mémorisée.

**SL : Action mémorisée et Limitée dans le temps**

Avec ce qualificateur, une action est immédiatement mémorisée et commence son exécution lors de l'activation de l'étape. L'action s'exécute pendant le délai indiqué, même si l'étape est désactivée.

L'exécution se termine également lors d'un appel ultérieur de l'action avec le qualificateur R (Reset).



2.3.2.6.3.2 Paramétrage d'une action booléenne

Dans le cas d'une action booléenne, le nom de la variable concernée doit être entré. Le qualificateur, bien qu'il puisse prendre toutes les valeurs possibles, sera généralement l'un de ceux-ci :

N	La variable est mise à 1 pendant la durée d'activation de l'étape, puis remise à 0 dès que l'étape n'est plus active.
S	La variable est mise à 1 dès l'entrée dans l'étape.
R	La variable est mise à 0 dès l'entrée dans l'étape.

Si par exemple la variable SA1 doit être mise à 1 pendant l'activation de l'étape :

**Paramètres d'action**

Qualificateur: N

Temps:

Variable d'action booléenne: SA1

Nom d'action:

Variable indicatrice:

Action éditée in situ:

☒ En langage ST ☐ En langage L

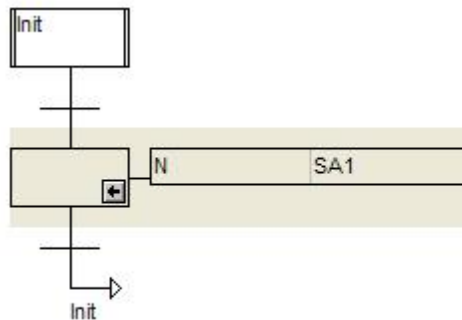
☐ En langage Apigraf

☐ Appel d'action inhibé

☐ Simple commentaire

OK Annuler Aide

Voici le résultat obtenu :



#### 2.3.2.6.3.3 Paramétrage d'une action "in situ"

Pour définir une action "in situ", c'est à dire écrite directement dans le diagramme, cocher la case correspondant au langage textuel souhaité :

**Paramètres d'action**

Qualificateur: N

Temps:

Variable d'action booléenne:

Nom d'action: Test

Variable indicatrice:

Action éditée in situ:

☒ En langage ST ☐ En langage L

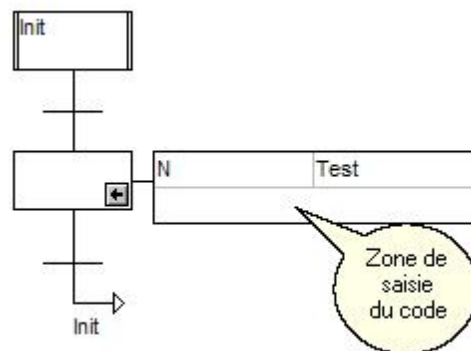
☐ En langage Apigraf

☐ Appel d'action inhibé

☐ Simple commentaire

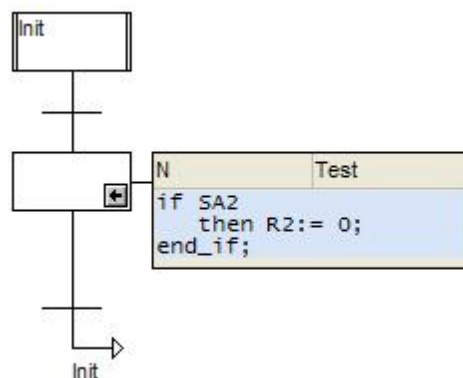
OK Annuler Aide

Le champ "nom d'action" peut être laissé vide, l'action ne sera dans ce cas pas nommée. Cependant, une action, même définie "in situ", peut être appelée en plusieurs endroits du diagramme. Dans ce cas, elle doit être nommée.



Lorsque la fenêtre est validée par OK, l'action vide est insérée dans le diagramme. Pour entrer du code, cliquer sur la zone de saisie du code. Le code entré ici doit être en langage ST, car c'est le langage coché plus haut.

Exemple :



#### 2.3.2.6.3.4 Paramétrage d'une action "externe"

Pour appeler une action "externe" (c'est à dire non définie dans le diagramme directement), entrer uniquement le nom désiré dans le champ "nom d'action".

**Paramètres d'action**

Qualificateur: N

Temps:

Variable d'action booléenne:

Nom d'action: Test

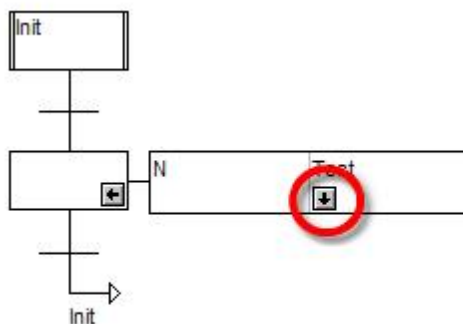
Variable indicatrice:

Action éditée in situ:

- ☐ En langage ST
- ☐ En langage IL
- ☐ En langage Apigra
- ☐ Appel d'action inhibé
- ☐ Simple commentaire

OK Annuler Aide

L'appel de l'action est ajouté au diagramme comme suit.

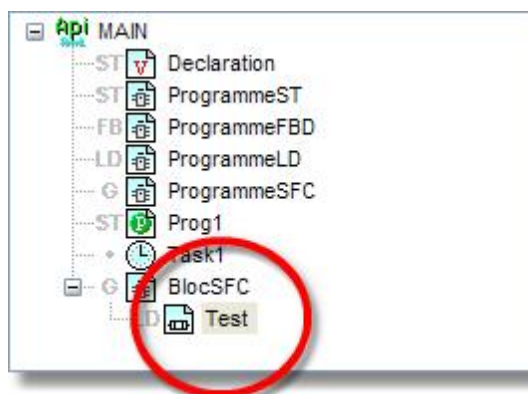


Le bouton entouré (📄) permet de sauter à la définition de l'action externe. Si celle-ci existe déjà, elle est directement affichée. A noter que cet appel d'action peut faire référence à une action nommée définie "in situ", auquel cas la partie correspondante du diagramme est affichée.

Si l'action n'existe pas encore, un clic sur le bouton 📄 provoque sa création. La fenêtre suivante est tout d'abord affichée, permettant de définir le langage de définition de l'action :



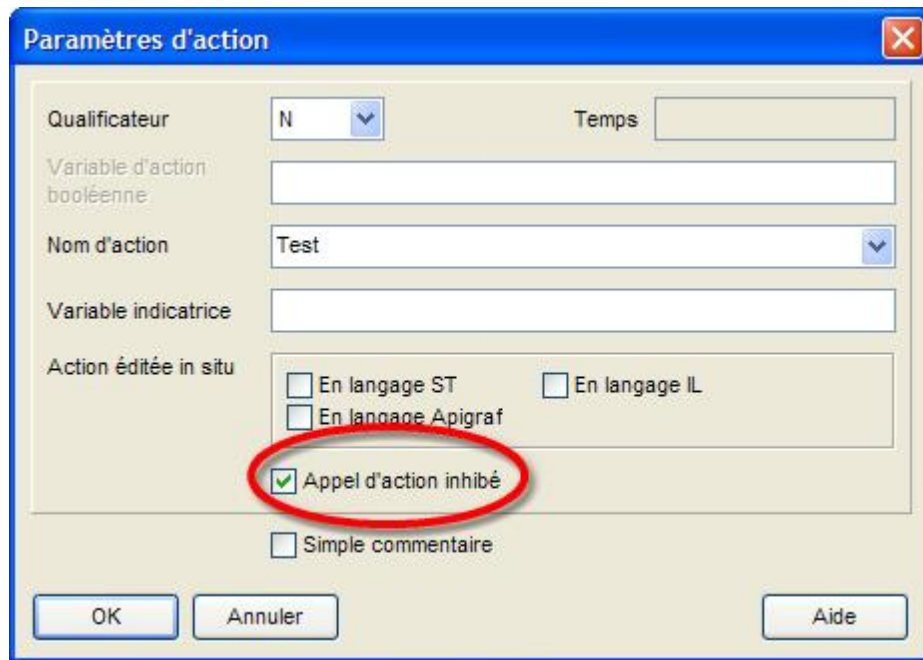
Une fois la fenêtre validée, l'action est ajoutée aux actions du POU en cours (ici BlocSFC) :



#### 2.3.2.6.3.5 Inhibition d'un appel d'action

Il est possible de désactiver l'appel à une action, sans pour autant le supprimer totalement. Cette possibilité permet d'effectuer des essais facilement en phase de test.

Dans la fenêtre de paramétrage d'action, cocher la case Appel d'action inhibé. L'action n'est pas compilée et apparaît en grisé dans le graphe.

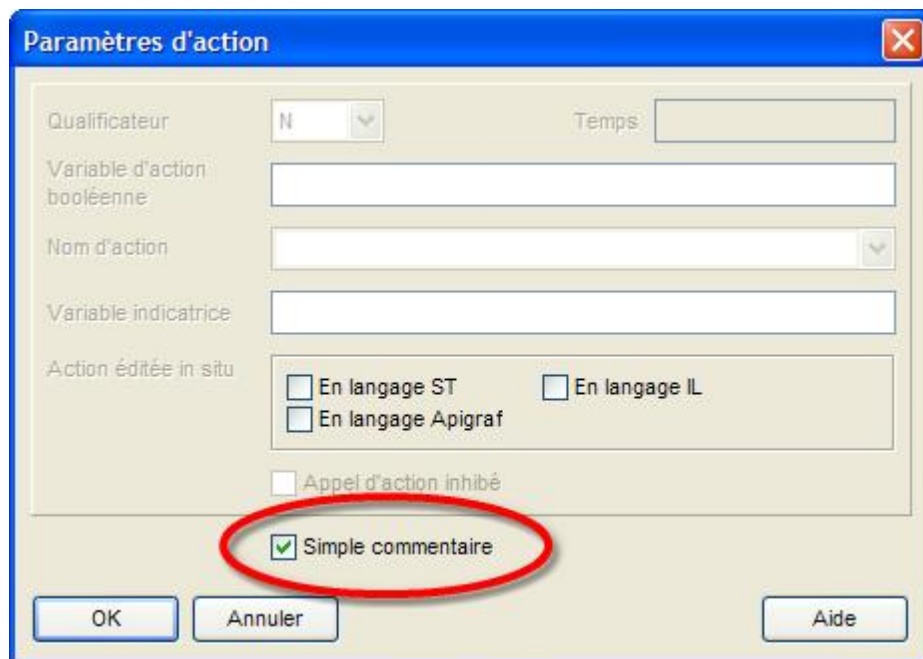


The screenshot shows the 'Paramètres d'action' (Action Parameters) dialog box. It has a blue title bar with a close button. The main area is light beige. Fields include: 'Qualificateur' (dropdown set to 'N'), 'Temps' (text box), 'Variable d'action booléenne' (text box), 'Nom d'action' (dropdown set to 'Test'), 'Variable indicatrice' (text box), and 'Action éditée in situ' (checkboxes for 'En langage ST', 'En langage IL', 'En langage Apigraf', and 'Appel d'action inhibé' which is checked and circled in red). There is also a 'Simple commentaire' checkbox which is unchecked. At the bottom are 'OK', 'Annuler', and 'Aide' buttons.

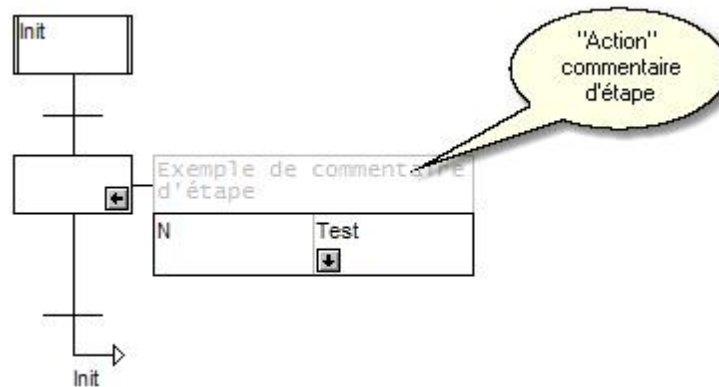
#### 2.3.2.6.3.6 Créer un commentaire d'étape

Pour commenter l'étape ou les actions de celle-ci, un commentaire peut être défini au moyen d'une action paramétrée en tant que commentaire.

Pour cela, cocher l'option Simple commentaire dans la fenêtre de paramétrage d'action. Le commentaire peut ensuite être saisi directement dans le graphe à la manière des instructions in situ.



This screenshot shows the same 'Paramètres d'action' dialog box, but with the 'Simple commentaire' checkbox checked and circled in red. The 'Appel d'action inhibé' checkbox is now unchecked. All other fields and the bottom buttons ('OK', 'Annuler', 'Aide') remain the same as in the previous screenshot.



#### 2.3.2.6.3.7 Manipulation des actions


Les manipulations suivantes sont possibles sur les actions


Ajouter une action	Clic droit sur l'étape puis "Ajouter une action"
Supprimer une action	Clic droit sur l'action puis "Supprimer l'action"
Monter une action	Clic droit sur l'action puis "Monter l'action"
Descendre une action	Clic droit sur l'action puis "Descendre action"
Paramétrer une action	Double clic sur l'action, ou clic droit sur l'action puis "Paramétrer l'action"
Copier une action	Boutons copier/coller, ou glisser/relâcher avec la souris
Déplacer une action	Boutons couper/coller

#### 2.3.2.6.4 Copie des éléments d'un diagramme SFC

Les étapes, les actions et les transitions peuvent être copiés puis collés dans un même diagramme SFC ou dans un autre diagramme SFC.

La sélection du ou des éléments à copier s'effectue par un simple clic sur l'objet ou par une fenêtre de sélection (définie avec la souris).

La copie peut alors être faite en cliquant sur le bouton "Copier" (  ) de la barre d'outils.

Pour recopier la sélection, il suffit de sélectionner une étape, une action ou une transition (du même diagramme SFC ou d'un autre) et de cliquer sur le bouton "Coller" (  ).

Il est possible également d'utiliser la méthode du glisser-relâcher : sélectionner les objets, placer la souris sur la sélection, appuyer sur la touche gauche de la souris et la maintenir tout en faisant glisser la position de la souris jusqu'à l'objet avant lequel la sélection doit être recopiée.

Afin de respecter la syntaxe du diagramme, la copie est effectuée de façon intelligente dans les cas suivants :


- copie d'une étape avant une étape (ou une action) : une transition est automatiquement ajoutée entre les deux étapes
- copie d'une transition avant une autre transition : une étape vide est automatiquement ajoutée entre les deux transitions
- coller une action n'est possible que si l'élément sélectionné pour la recopie est une étape
- la recopie d'une étape initiale ou d'un renvoi vers une étape est interdit



- la recopie d'étapes ou de transitions avant l'étape initiale est interdit

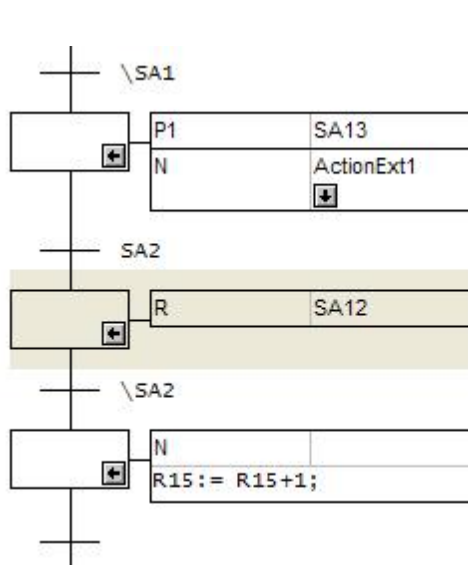
### 2.3.2.6.5 Suppression d'éléments d'un diagramme SFC

Pour supprimer des éléments du diagramme, il suffit de les sélectionner puis de frapper la touche Suppr (Del).

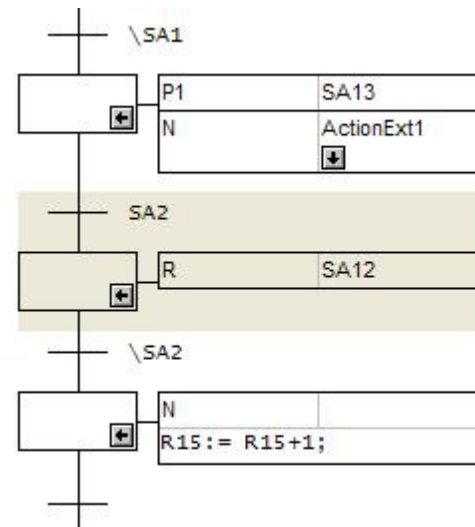
Il est possible également de faire une suppression avec copie des éléments supprimés dans le presse-papier, au moyen du bouton "couper" (  ).

Attention cependant, la suppression n'est pas toujours possible, car le principe suivant doit toujours être respecté : après chaque opération, qu'il s'agisse d'une création, copie ou suppression, le diagramme doit toujours être dans un état valide, c'est à dire que les règles de disposition des étapes et transitions doivent être observées.

Par exemple, il n'est pas possible de supprimer seulement une étape. Il faut également supprimer une transition, avant ou après, de manière à ce que le diagramme reste correct.



Suppression impossible



Suppression possible

### 2.3.2.6.6 Exemple de création d'un diagramme SFC

L'exemple est un chenillard comprenant 3 lampes représentées par les variables Lampe1, Lampe2 et Lampe3. Le démarrage du chenillard s'effectue grâce à la variable StartChenillard et s'arrête lorsque la variable StopChenillard est à "vrai".

Sélectionner (ou créer) un POU de type Bloc de fonction "BlocSFC" en langage SFC.

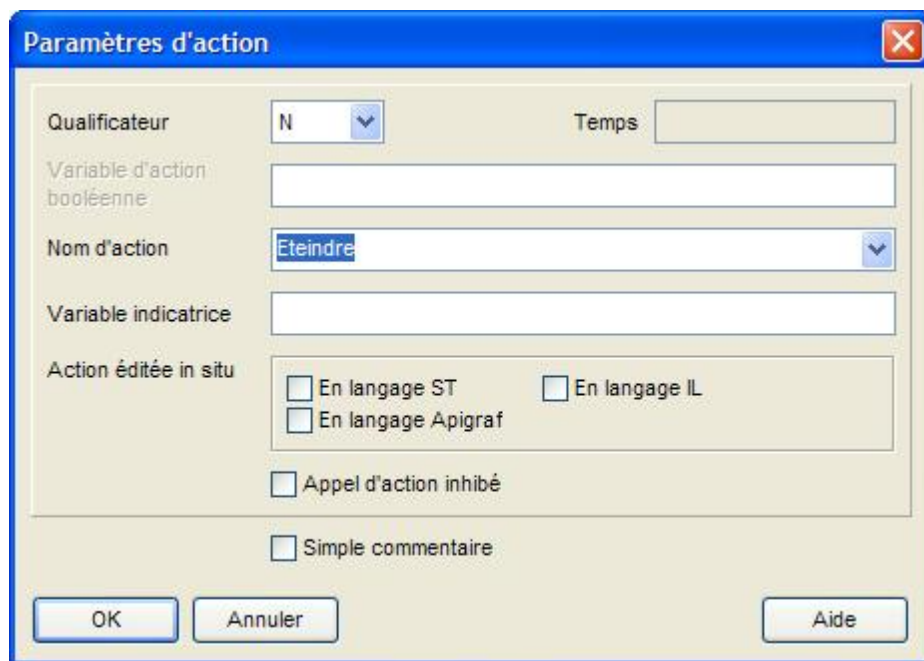
La première étape à effectuer est de créer les variables de l'équation. Sélectionner l'onglet "Déclarations" et insérer les déclarations suivantes :

```
Var_Input
  StartChenillard : bool;
  StopChenillard  : bool;
End_Var

Var_Output
  Lampe1 : bool;
  Lampe2 : bool;
  Lampe3 : bool;
End_Var
```

Sélectionner l'onglet "Instructions".

Sélectionner l'étape "Init" puis utiliser le menu contextuel pour ajouter une action avec le qualificateur "N" (action exécutée tant que l'étape est active) et la nommer "Eteindre".



Cliquer sur le bouton  et choisir le langage ST.

Un POU nommé "Eteindre" est alors créé dans l'arborescence de la librairie, la page à onglet "Instructions" qui lui est associée est automatiquement sélectionnée.

Ecrire les instructions suivantes :

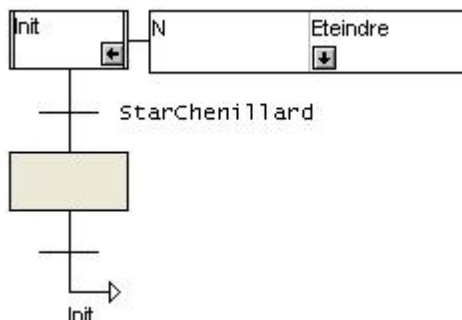
```
Lampe1:= false;
Lampe2:= false;
Lampe3:= false;
```

Resélectionner le POU "BlocSFC" dans l'arborescence de la librairie.



Sélectionner la transition du diagramme, et utiliser le menu contextuel (clic droit de la souris) pour éditer la transition : "StartChenillard".

Resélectionner la transition puis cliquer sur  pour créer une nouvelle étape.



Utiliser le menu contextuel pour renommer l'étape en "Allumer1" puis ajouter une action avec le qualificateur "D", un temps "T#1s" (l'action s'exécute 1 seconde après l'activation de l'étape) et la nommer "ActionAllumer1".

**Paramètres d'action**

Qualificateur: D Temps: T#1s

Variable d'action booléenne:

Nom d'action: ActionAllumer1

Variable indicatrice:

Action éditée in situ:

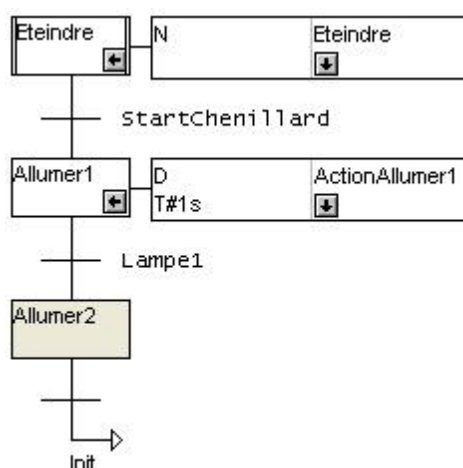
- ☒ En langage ST
- ☐ En langage IL
- ☐ En langage Apigraf
- ☐ Appel d'action inhibé
- ☐ Simple commentaire

OK Annuler Aide

Cliquer sur le bouton et choisir le langage ST.  
 Un POU nommé "ActionAllumer1" est alors créé.  
 Ecrire les instructions suivantes :

```
Lampe1:= true;
Lampe2:= false;
Lampe3:= false;
```

Resélectionner le POU "BlocSFC" dans l'arborescence de la librairie.  
 Sélectionner la transition suivante, y ajouter la condition "Lampe1", la resélectionner puis cliquer sur pour créer une nouvelle étape.

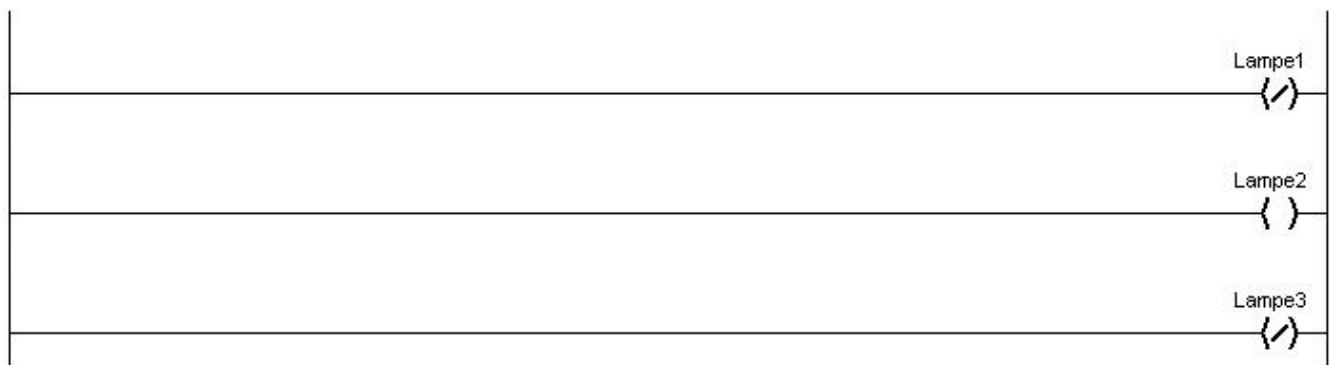


Utiliser le menu contextuel pour renommer l'étape en "Allumer2" puis ajouter une action avec le qualificateur "D", un temps "T#1s" (l'action s'exécute 1 seconde après l'activation de l'étape) et la nommer "ActionAllumer2".

Cliquer sur le bouton et choisir le langage ladder.

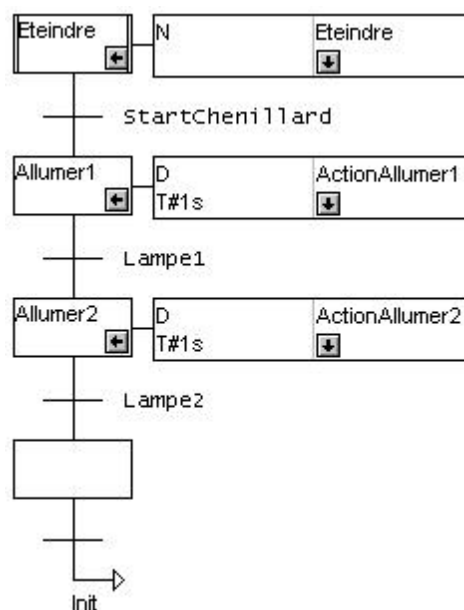
Un POU nommé "ActionAllumer2" est alors créé.

Ajouter les diagrammes pour réaliser les équations suivantes :



Resélectionner le POU "BlocSFC" dans l'arborescence de la librairie.

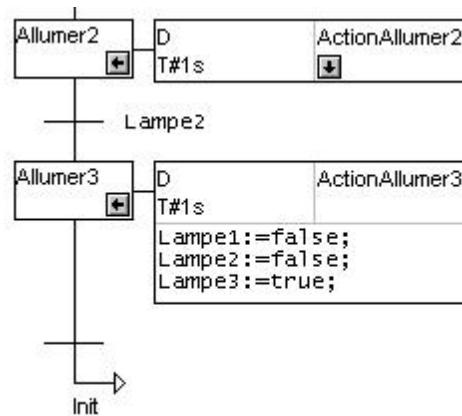
Sélectionner la transition suivante, y ajouter la condition "Lampe2", la resélectionner puis cliquer sur pour créer une nouvelle étape.





Nommer la nouvelle étape "Allumer3" et créer une action "ActionAllumer3" (qualificateur D, Temps de 1 seconde) en cochant l'option "Edition de l'action in situ".

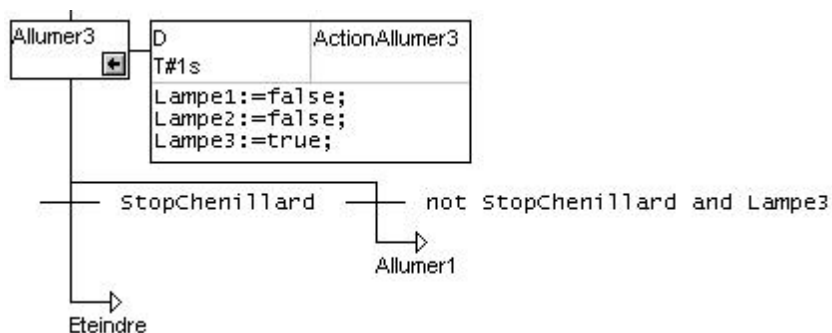
Sélectionner le rectangle dédié aux instructions puis écrire les instructions suivantes :


```
Lampe1:= false;
Lampe2:= false;
Lampe3:= true;
```



Sélectionner la transition suivante, y ajouter la condition "StopChenillard", la resélectionner puis cliquer sur  pour créer une divergence en OU.

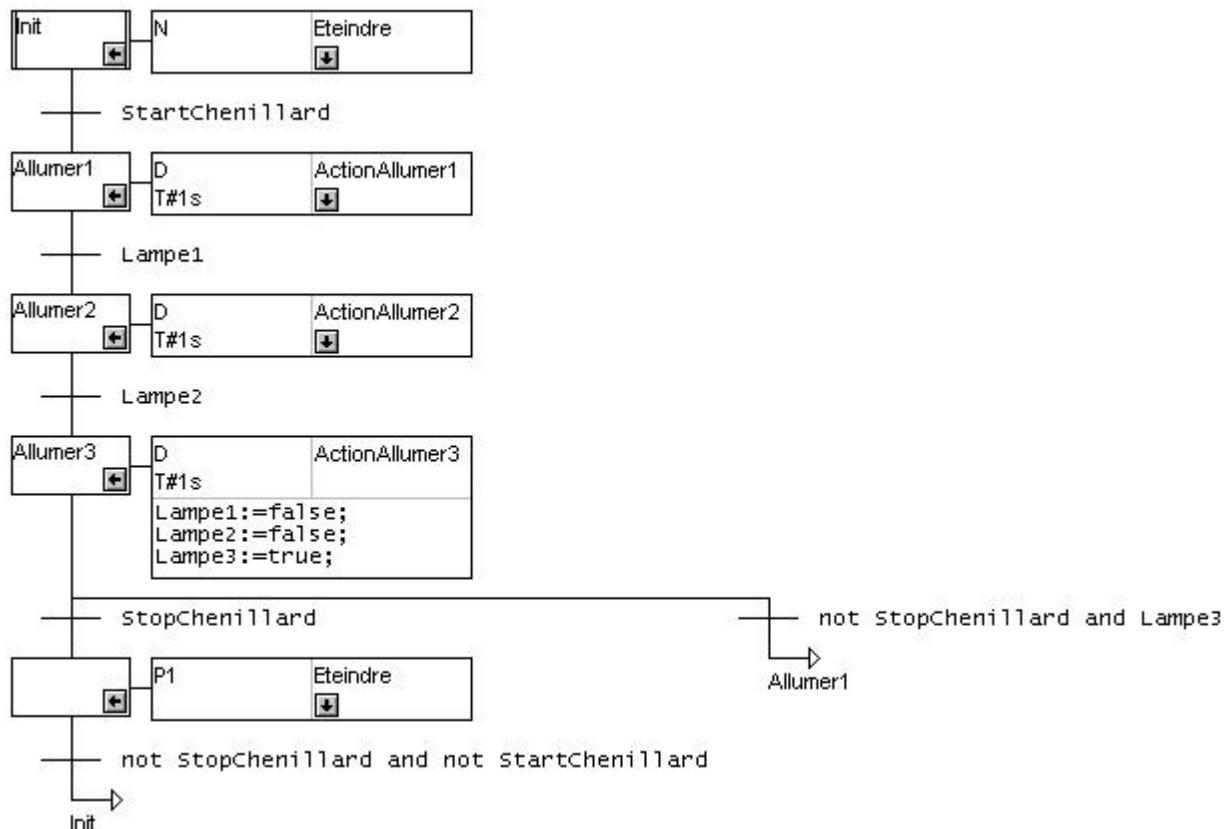
Ajouter la condition "not StopChenillard and Lampe3", resélectionner cette transition parallèle puis cliquer sur  en précisant l'étape de retour "Allumer1".



Sélectionner la transition "StopChenillard" puis cliquer sur  pour créer une nouvelle étape.

Ajouter l'action "Eteindre" (pour appeler l'action précédemment créée) avec le qualificateur P1(exécution à l'activation de l'étape).

Sélectionner la dernière transition et ajouter la condition "not StopChenillard and not StartChenillard".



**Le diagramme est complet.**

Pour vérifier qu'aucune erreur n'a été commise pendant la création, il suffit alors de compiler.

Pour le tester, sélectionner un bloc de programme (Prog1 dans l'exemple), déclarer une variable de type "BlocSFC" et appeler le bloc dans la partie instructions :

#### Déclaration :

VarTestSFC : BlocSFC;

#### Instruction :

VarTestSFC(StartChenillard:= SA0, StopChenillard:= SA1, Lampe1 => L2, Lampe2 => L3, Lampe3 => L4);

Compiler puis exécuter.

### 2.3.2.6.7 Réinitialisation d'un diagramme SFC

Il peut être nécessaire de forcer un diagramme SFC à repartir de son étape initiale.

Cette opération ne pose pas de problème dans le diagramme lui-même si ce forçage est nécessaire dans certaines étapes particulières, il s'agit alors de simples retour à l'étape initiale après une transition..

Si par contre le forçage doit être effectué sur une condition quelle que soit l'étape active, il est préférable de le programmer depuis l'extérieur du diagramme, par exemple dans le bloc depuis lequel le diagramme est appelé. Le forçage est effectué avec la fonction "ResetSFC".

Si par exemple le diagramme est programmé dans un bloc appelé SequencePrincipale, le reset est obtenu en faisant :

```
ResetSFC(SequencePrincipale)
```

Attention, si ce diagramme contient également d'autres séquences, dans des sous-blocs ou des actions, toutes les séquences SFC sont également forcées à leur étape initiale.

### 2.3.2.7 Création de bloc ST, IL et SFC textuel

La création de blocs programmés dans l'un de ces langages textuels s'effectue comme pour les langages graphiques. Par contre l'édition du code se fait dans un éditeur de texte classique.

### 2.3.2.8 Manipulation des POU d'une librairie

Les POU peuvent être déplacé et copiés d'un groupe à l'autre, d'une librairie à une autre, en les faisant glisser au moyen de la souris.

Ils peuvent être "montés" ou "descendus" dans l'arborescence de conception au moyen des boutons  et .


Pour renommer un POU, faire un clic droit et utiliser l'option "renommer". L'atelier vérifie que le nom que vous donnez à un POU n'est pas déjà utilisé dans les librairies chargées.

## 2.3.3 Impression

Plusieurs niveaux d'impression sont proposés :

- impression des diagrammes seuls pour les langages graphiques (SFC, LD et FBD)
- impression d'un POU (informations, déclarations et instructions)
- impression de la librairie utilisateur
- impression d'un projet (toutes les librairies utilisateur)

### 2.3.3.1 Impression de diagramme

Dans le cas des langages graphiques, il est possible d'imprimer directement les diagrammes dans la page à onglets nommée "Instructions" en cliquant sur l'icône  de la barre d'outil.

### 2.3.3.2 Impression de POU

L'option "Imprimer" est disponible dans le menu surgissant des POU.

L'impression du contenu d'un POU se présente sur une ou plusieurs pages et comporte :

- les informations disponibles sur le POU parmi les suivantes :
  - les propriétés
  - le symbole interface
  - le descriptif d'utilisation
  - les déclarations
  - les instructions
- les informations sur les POU appartenant au POU sélectionné (cas des POU appartenant à des sous-groupes ou encore des actions définies dans un POU en langage SFC)

### 2.3.3.3 Impression d'une librairie

L'option "Imprimer" est disponible dans le menu surgissant des librairies.

Elle provoque l'impression de tous les POU contenus dans la librairie.

### 2.3.3.4 Impression d'un projet

L'option "Imprimer" est disponible dans le menu Projet de l'atelier de programmation.

Elle provoque l'impression de toutes les librairies ouvertes dans le projet.

## 3 Atelier de supervision

Pour un premier démarrage, aller consulter le chapitre :

[Création de pages de supervision - Principes généraux](#)  99

## **3.1 Menu principal de l'application**

### **3.1.1 Menu fichier**

#### **3.1.1.1 Menu fichier - Nouveau**

Le menu « nouveau » est utilisé pour créer une nouvelle page de supervision. Une fois créée, la page de supervision doit être sauvegardée.

#### **3.1.1.2 Menu fichier - Sélection projet**

Le menu « projets » appelle la page de sélection de projets. Ce menu permet donc de sélectionner un projet, de créer un nouveau projet ou de supprimer un projet.

Ce menu n'est pas disponible si l'atelier de supervision a été démarré depuis Optima PLC.

#### **3.1.1.3 Menu fichier - Enregistrer**

Le menu « sauver » permet de sauvegarder la page de supervision courante dans le projet. Si la page de supervision courante vient d'être créée, une boîte de dialogue permettra de saisir un nom de sauvegarde pour la page.

#### **3.1.1.4 Menu fichier - Ouvrir**

Le menu « ouvrir » permet de charger une page de supervision du projet en cours dans l'atelier de supervision.

#### **3.1.1.5 Menu fichier - Enregistrer sous**

Le menu « Sauver sous » permet de sauvegarder une page de supervision existante sous un autre nom de fichier. La boîte de dialogue permet de définir le nouveau nom du fichier ainsi que sa place dans l'arborescence des répertoires.

#### **3.1.1.6 Menu fichier - Quitter**

Ce menu permet de quitter l'atelier de supervision.

### **3.1.2 Menu Edition**

#### **3.1.2.1 Menu édition - Annuler**

Permet d'annuler la dernière action effectuée.

#### **3.1.2.2 Menu édition - Couper**

Efface tous les objets sélectionnés et les place dans un presse-papiers. Ces objets pourront être « collés » par la suite.

#### **3.1.2.3 Menu édition - Copier**

Copie tous les objets sélectionnés dans un presse-papiers.

#### **3.1.2.4 Menu édition - Coller**

Recopier tous les objets présents dans le presse-papiers sur la page en cours.

#### **3.1.2.5 Menu édition - Supprimer**

Supprime tous les objets sélectionnés.

#### **3.1.2.6 Menu édition - Tout sélectionner**

Sélectionne tous les objets présents sur la page.

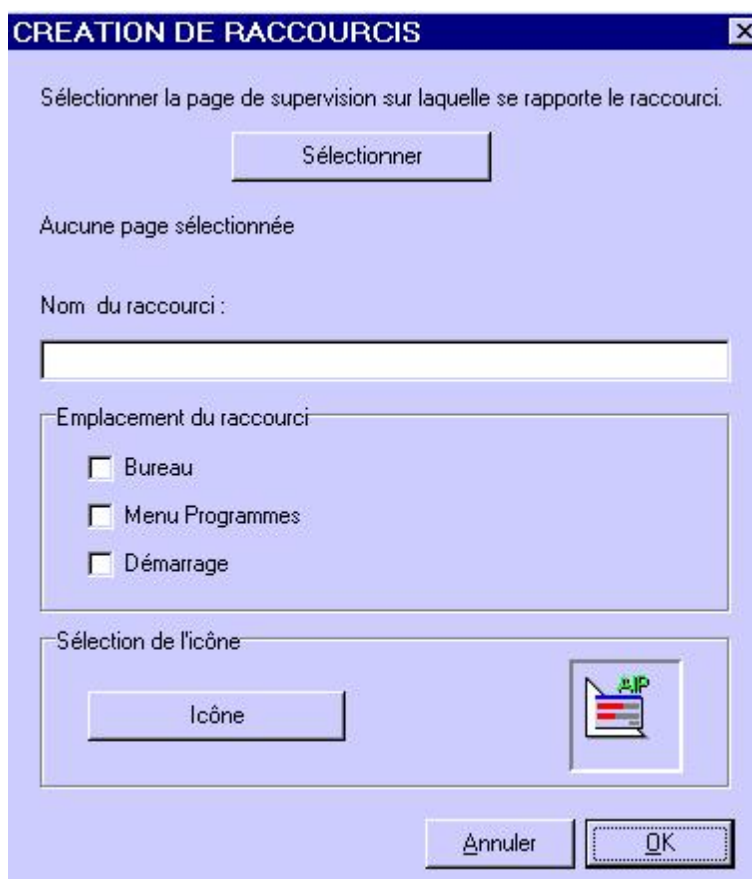
### **3.1.3 Menu Outils**

#### **3.1.3.1 Menu Création de raccourcis**

Ce menu permet à l'utilisateur de créer des raccourcis vers des pages de supervision et de placer ces raccourcis sur le bureau, dans le menu programmes ou encore dans le démarrage automatique de WINDOWS®



Le menu ouvre la fenêtre suivante :



Les différents contrôles de cette page sont :

**Bouton sélectionner**

Ce bouton permet de sélectionner la page de supervision sur laquelle se rapportera le ou les raccourcis créés.

**Nom du raccourci**

Permet de personnaliser le nom du raccourci. Le nom du raccourci par défaut est : « Raccourci vers » + le nom de la page de supervision.

**Emplacement du raccourci**

Ce contrôle est composé de 3 cases à cocher, il permet de spécifier la destination du raccourci créé (Bureau, menu programmes ou démarrage). Il est possible de sélectionner plusieurs destinations simultanément.

**Sélection de l'icône**

Il est possible de personnaliser l'image de l'icône en cliquant sur le bouton « Icône » et en choisissant un fichier icône valide (\*.ICO).

L'icône par défaut est celui de l'atelier de supervision.

Un aperçu de l'icône sélectionné est également disponible dans la fenêtre.

### 3.1.3.2 Menu Traductions utilisateur - Ouvrir un fichier langage

L'utilisateur a la possibilité de gérer plusieurs langages dans les pages de supervision (voir gestion des traductions ). Ce menu ouvre un fichier langage existant dans le projet (le nom du fichier étant du type : langage.1 langage.2 langage.3 ...)

### 3.1.3.3 Menu Traductions utilisateur - Nouveau fichier langage

Crée un nouveau fichier langage. Si aucun fichier langage n'existe dans le projet en cours, un fichier LANGAGE.1 est alors créé. Si un fichier de langage existe déjà (LANGAGE.1), alors un nouveau fichier (LANGAGE.2) va être créé, etc..

### 3.1.3.4 Menu Traductions utilisateur - Extraction Auto textes de la page

Ce menu permet d'automatiser l'écriture du fichier de langage pour une page de supervision contenant déjà des textes. Ce menu ouvre la fenêtre suivante :

Le nom du fichier spécifie le fichier langage qui va être utilisé pour effectuer l'opération. Si celui-ci n'existe pas, il sera automatiquement créé.

Les opérations possibles sont :

#### Remplacer Texte par @LGx :Texte

Avec cette opération tous les champs texte de tous les objets de la page de supervision seront remplacés par « @LGx : » suivi du texte initial du champ, x représentant un nombre donné pour référencer la traduction.

Exemple : Si une page contient un bouton windows dont le texte est « Quitter » et si le fichier LANGAGE.1 contient :

@LG0 :  
Nouveau  
@LG1 :  
Page2

Le texte du bouton windows sera remplacé par « @LG2 :Quitter » et le fichier LANGAGE.1 deviendra :

@LG0 :  
Nouveau  
@LG1 :  
Page2  
@LG2 :  
Quitter

#### Remplacer Texte par @LGx

Cette opération est la même que la précédente à la différence près que les textes des objets ne contiendront pas leur ancienne valeur en commentaire.

« Quitter » serait remplacé par « @LG2 » dans l'exemple du dessus.

#### Remplacer @LGx par les chaînes du fichier

Cette opération est l'opération inverse des deux opérations précédentes.

Si les textes d'une page de supervision ont été extraits vers un fichier langage, cette opération permet de récupérer les anciens textes des objets dans le fichier langage mentionné et de les réinsérer dans les objets de la page.

#### Option « Réutiliser les textes déjà référencés »

Cette option permet de rechercher d'abord si le texte à extraire est présent dans le fichier langage avant de créer une nouvelle référence. Si c'est le cas, la référence existante sera utilisée.

### 3.1.3.5 Menu Explorateur Apigraf IP

Permet de lancer l'explorateur Apigraf IP. Cet utilitaire offre la possibilité de gérer ou de transférer des fichiers sur des postes PLC ou des postes WIN pourvus de l'application APIFTPServer.

### 3.1.3.6 Menu Gestionnaire d'alarmes

Lance le gestionnaire d'alarmes centralisées : OPTIM'ALARM. Si le gestionnaire est déjà lancé alors il est simplement affiché en premier plan.

## 3.1.4 Menu Affichage

### 3.1.4.1 Menu Affichage - Barres d'outils

Ce menu référence les quatre barres d'outils disponibles.

**Gestion :**

Si ce menu est coché la barre d'outils Gestion sera visible.

**Edition :**

Si ce menu est coché la barre d'outils Edition sera visible.

**Fonction :**

Si ce menu est coché la barre d'outils Fonction sera visible. Le barre d'outils fonction permet de lancer une page de supervision en exécution.

**Palette :**

Si ce menu est coché la Palette qui contient tous les objets usuels pour la construction des pages de supervision sera visible.

### 3.1.4.2 Menu Affichage - Fenêtre Fonction

Si ce menu est coché la fenêtre Fonctions sera visible. La fenêtre "Fonctions" apparait en bas de l'écran. Cette fenêtre permet de lancer une page de supervision en exécution ou encore d'exécuter des zooms sur les objets.

### 3.1.4.3 Menu Affichage - Explorateur d'objets

Si ce menu est coché la fenêtre de l'explorateur d'objets sera visible.

L'explorateur d'objets liste dans une fenêtre arborescente tous les objets présents dans la page de supervision courante. Il est possible depuis cette page de forcer un objet à être invisible pendant le phase de conception de la page de supervision.

Plus de détails dans le chapitre :

[Explorateur d'objets](#) <sup>300</sup>

### 3.1.4.4 Menu Affichage - Fenêtre Alignement

Si ce menu est coché la fenêtre Alignement sera visible. La fenêtre "Alignement" apparait en bas de l'écran. Cette fenêtre permet d'avoir accès à des fonctions d'alignement des objets de supervision.



Les boutons de fonction de la palette alignement ne sont pas actifs si aucun objet de la page de supervision n'est sélectionné.

Pour les quatres fonctions du haut, il faut au moins deux objets sélectionnés sur la page et il faut que ces objets aient le même propriétaire : (exemple : 2 objets dans le même panneau ou 2 objets directement sur la même page)

Pour les quatres fonctions du bas, il faut au moins deux objets sélectionnés sur la page (peut importe les propriétaires des objets).

**Pour sélectionner plusieurs objets :**

Cliquer sur le premier objet à sélectionner. Appuyer ensuite sur la touche "SHIFT" du clavier et maintenir la touche enfoncée.

Cliquer ensuite sur les autres objets sélectionnés. Si un objet à été sélectionné par erreur, cliquer à nouveau sur celui-ci pour le désélectionner.

Relâcher la touche "SHIFT" une fois la sélection effectuée.

### 3.1.5 Menu Objet

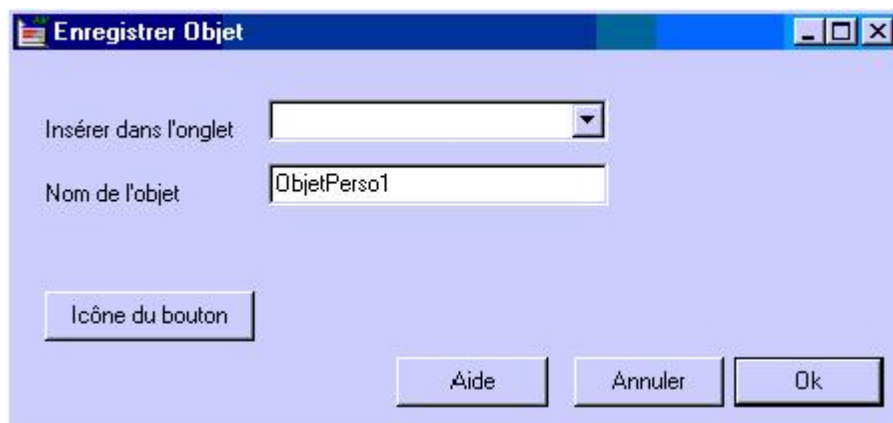
#### 3.1.5.1 Menu Objet - Ajouter

Ce menu permet d'enregistrer un objet sélectionné avec tous ses paramètres.

Il est aussi accessible en faisant un clic droit sur l'objet pour faire apparaître le menu contextuel et en cliquant sur «enregistrer objet ».

Un bouton sera créé dans un onglet existant ou non qui permettra de rappeler cet objet.

Le menu ouvre la fenêtre suivante :



Les différents paramètres de cette fenêtre sont:

##### Insérer dans l'onglet:

Cette case relate tous les onglets existants en cliquant sur la flèche à droite de la case pour insérer le bouton de l'objet à enregistrer. Il est possible de sélectionner un onglet dans la liste ou de taper le nom de l'onglet que vous voulez créer.

##### Nom de l'objet:

Cette case permet de rentrer le nom du bouton qui s'affichera dans le Hint quand le curseur passera au-dessus. Il définit aussi le nom du fichier ou sera enregistré l'objet:

(ex: User0ObjetPerso0.bib)

##### Icône du bouton :

Le bouton aura une icône par défaut:

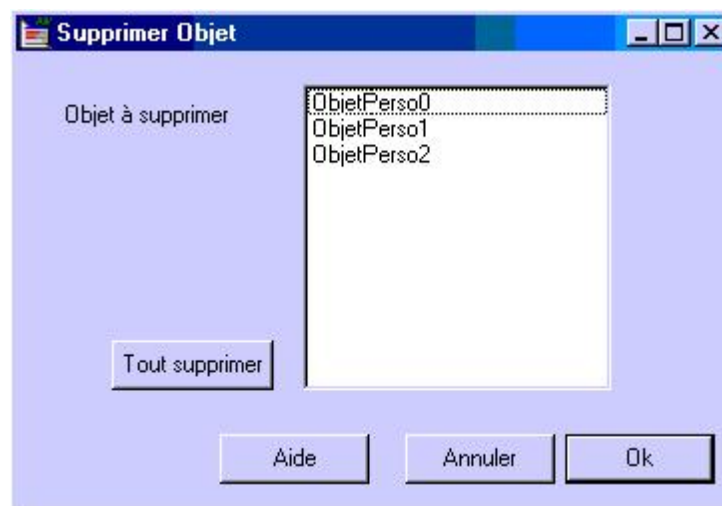


Il est possible de personnaliser l'image de l'icône en cliquant sur le bouton et en choisissant un fichier image valide. Un aperçu de l'image est disponible dans cette boîte de dialogue.

#### 3.1.5.2 Menu Objet - Supprimer

Ce menu permet d'effacer des objets personnalisés. Il effacera le bouton, l'onglet créé s'il n'a plus de bouton dedans ainsi que le fichier de l'objet enregistré.

Ce menu ouvre la fenêtre suivante :



Les paramètres sont les suivant:

**Objet à supprimer:**

Cette liste relate tous les objets qui ont été enregistrés (objets personnalisés).

Il suffit de sélectionner le ou les objets à supprimer. Pour sélectionner plusieurs objets, cliquer sur un objet et faire glisser la souris jusqu'au dernier objet que vous voulez supprimer ou cliquer sur ceux que vous voulez en restant appuyer sur la touche "Control".

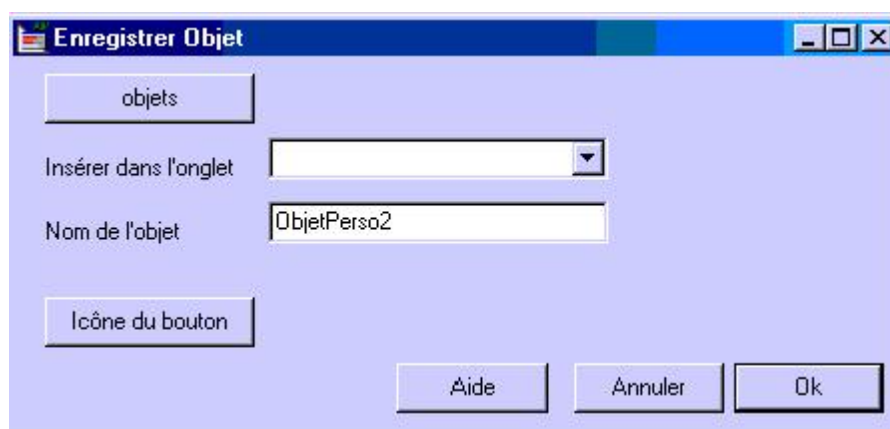
**Tout sélectionner:**

Permet de sélectionner tous les objets de la liste.

### 3.1.5.3 Menu Objet - Importer

Ce menu permet d'importer des objets personnalisés grâce au fichier d'extension « bib ». Un bouton sera placé dans l'onglet désiré.

Ce menu ouvre la fenêtre suivante :



Les différents paramètres de cette fenêtre sont:

**Objets:**

Ce bouton permet de choisir le fichier correspondant à l'objet à importer. Ce fichier doit avoir l'extension "bib".

**Insérer dans l'onglet:**

Cette case relate tous les onglets existants en cliquant sur la flèche à droite de la case pour insérer le bouton de l'objet à enregistrer. Il est possible de sélectionner un onglet dans la liste ou de taper le nom de l'onglet que vous voulez créer.

**Nom de l'objet:**

Cette case permet de rentrer le nom du bouton qui s'affichera dans le Hint quand le curseur passera au-dessus. Il définit aussi le nom du fichier ou sera enregistré l'objet:

(ex: User0ObjetPerso0.bib)

**Icône du bouton :**

Le bouton aura une icône par défaut:



Il est possible de personnaliser l'image de l'icône en cliquant sur le bouton et en choisissant un fichier image valide. Un aperçu de l'image est disponible dans cette boîte de dialogue.

### 3.1.6 Menu Macros

#### 3.1.6.1 Menu Macros - Macros locales

Ce menu affiche tout d'abord la liste des macros locales disponibles.

On peut exécuter une macro à partir de la liste comme suit :

1. Sélectionner dans la liste la macro à exécuter à l'aide de la souris ou des flèches directionnelles du clavier. Celle-ci apparaît alors en surbrillance.
2. Cliquer sur le bouton «exécuter »
3. Une fenêtre contenant le texte de la macro apparaît à l'écran.
4. Compléter le texte de la macro ( Ajouter les paramètres entre parenthèses )
5. Valider avec le bouton « OK »

#### 3.1.6.2 Menu Macros - Macros PLC

Ce menu permet d'exécuter des macros sur le poste APIGRAF PLC Distant.

La fenêtre qui s'affiche à l'écran permet de saisir l'adresse IP du poste APIGRAF PLC ainsi que le texte de la macro à exécuter.

Cliquer sur le bouton «executer » pour exécuter la macro sur le poste distant.

Note : Le champ de l'adresse IP du poste APIGRAF PLC est automatiquement renseigné avec la valeur par défaut de l'adresse IP distante du projet.

#### 3.1.6.3 Menu Macros - Macros Utilisateur

Les macros utilisateur sont des macros contenues dans des DLL développées en dehors d'APIGRAF WIN. .  
Ce menu propose 3 sous menus :

**Liste des DLL Utilisateur :**

Ce sous-menu permet de visualiser les DLL utilisateur qui ont été chargées avec le projet.

**Charger DLL Utilisateur :**

Ce sous-menu permet de charger une DLL utilisateur avec le projet. La DLL sera rechargée à chaque nouvelle ouverture ou exécution du projet.

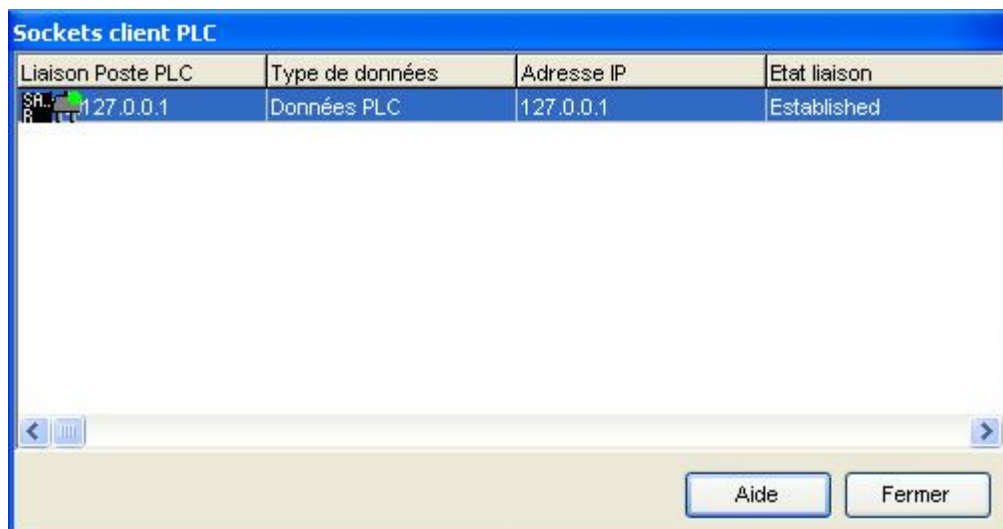
**Décharger DLL Utilisateur :**

Ce sous-menu permet de décharger une DLL utilisateur d'un projet. Une fois déchargée, la DLL ne sera plus chargée lors d'une nouvelle ouverture ou exécution du projet.

### 3.1.7 Menu Diagnostics

#### 3.1.7.1 Clients TCP/IP données PLC

Ce menu ouvre la fenêtre de diagnostic des sockets d'échange de données avec OPTIMA PLC ou APIGRAF.



Cette fenêtre liste tous les sockets connectés ou non à des runtimes OPTIMA PLC ou APIGRAF.

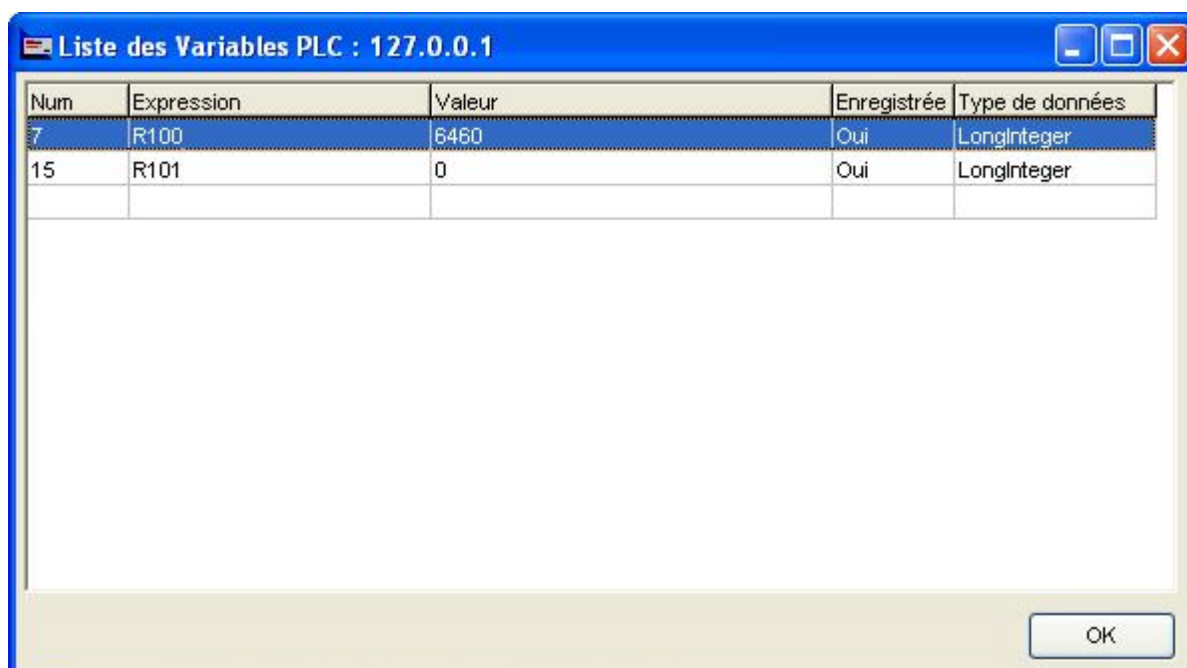
L'adresse IP 127.0.0.1 correspond toujours à l'adresse locale du PC.

Pour chaque socket est mentionné l'adresse IP distante ainsi que l'état de liaison.

La liaison est effective quand l'état est "Established"

Dans le cas ou des tableaux de mesures distants sont également supervisés, les sockets de liaisons correspondant aux différents tableaux de mesures supervisés figurent dans cette liste.

Dans le cas des sockets de données PLC ( voir le champ type de données dans la liste) il est possible d'avoir des détails en effectuant un double-click sur le socket en question. La fenêtre suivante s'ouvre alors :



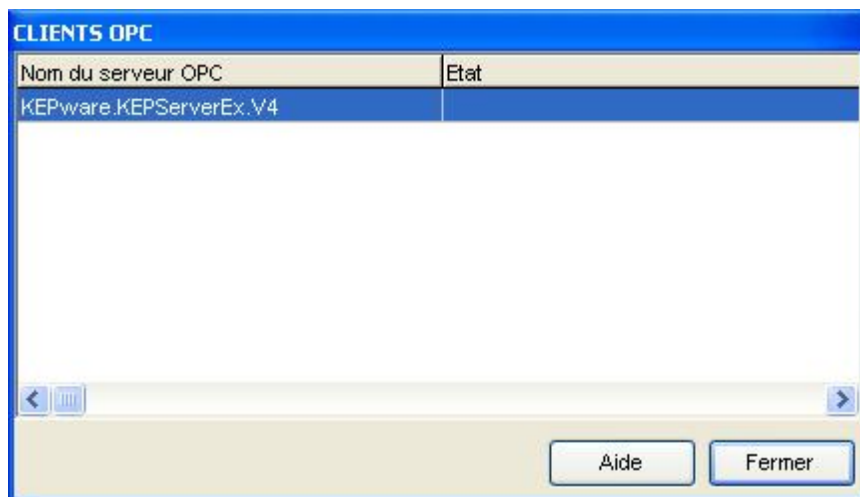
Cette fenêtre liste toutes les variables supervisées sur le socket (ici R100 et R101) ainsi que :

- La valeur de la variable (si celle si est enregistrée donc connue sur le poste distant)
- Si la valeur est enregistrée
- Le type de données de la variable si il est connu.

### 3.1.7.2 Clients OPC

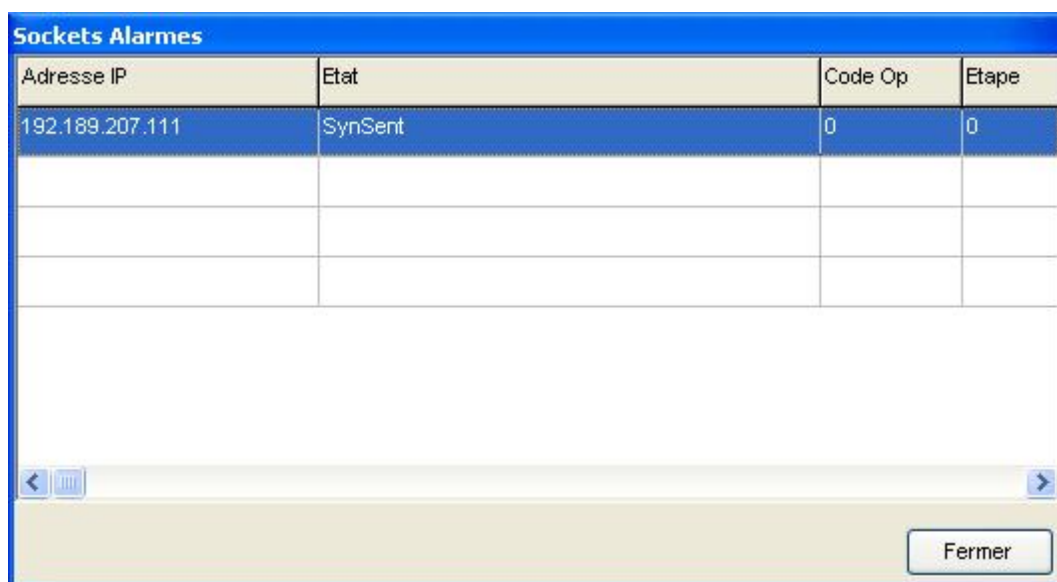
Ce menu ouvre la fenêtre de diagnostic des liaisons avec des serveurs OPC.

Cette fenêtre liste simplement les serveurs OPC avec lesquels une liaison est demandée.



### 3.1.7.3 Clients TCP/IP données Alarmes

Ce menu ouvre la fenêtre de diagnostic des sockets d'échanges de données des alarmes centralisées par Optim'Alarm.



La fenêtre liste les liaisons demandées pour la mise à jour des journaux d'alarmes.

### 3.1.7.4 Tableaux de mesure liés en mémoire

Ce menu ouvre la fenêtre de diagnostic des fichiers de données mappées en mémoire qui contiennent des tableaux de mesures locaux provenant d'OPTIMA PLC.

Lorsqu'un tableau de mesure supervisé dans un objet Graphe provient d'Optima PLC sur le poste local, les données sont échangées en mémoire et non via un socket comme c'est le cas pour des tableaux provenant d'Optima PLC sur poste distant ( sur le réseau ethernet )



TABLEAUX DE MESURES LIES EN MEMOIRE					
Nom Tableau	Nom Fichier mappé	Taille	Nb Acquisitions	Version données	Nb User
TABCANAL1	_TSOPLC_TABCANAL1	4000	3999	22699	1
TABCANAL0	_TSOPLC_TABCANAL0	4000	3999	22699	1

Réinitialisation des liens tableaux

Fermer

La fenêtre liste les liaisons de tableaux en mémoire avec :

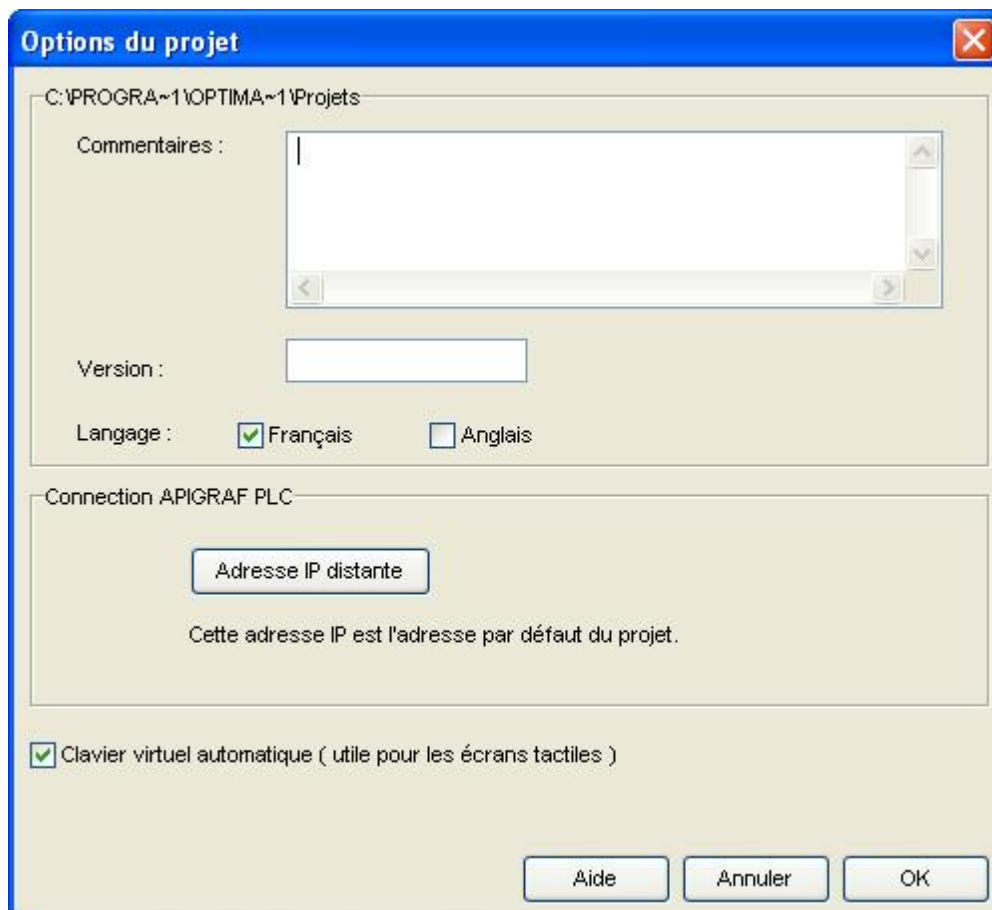
- Le nom du tableau : celui qui est renseigné dans l'objet de supervision graphe
- Le nom du fichier mappé en mémoire : Ce nom est construit automatiquement à partir du nom de tableau
- La taille : nombre maximum d'éléments dans le tableau
- Nb Acquisitions: nombre d'éléments présents dans le tableau
- Version données : version des dernières données mises à jour
- Nb user : nombre de demande de supervision de ce tableau. Ce nombre est incrémenté à chaque fois qu'un objet graphe se rapportant à ce tableau est ouvert.

Si la liaison n'a pas été effectuée au démarrage, certaines données sont marquées "NC" comme le champ Taille, NbAcquisitions, etc...

### 3.1.8 Menu Options

#### 3.1.8.1 Menu Options - Options générales

Le menu « Options générales » ouvre la page de dialogue suivante :

**Boîte de saisie des commentaires :**

Cette fenêtre permet d'associer des commentaires au projet en cours.

**Boîte de saisie de la version :**

Cette fenêtre permet de d'associer un numéro de version au projet en cours

**Choix du langage de l'application :**

Les langages français et anglais sont actuellement disponibles. Sélectionner le langage en activant la case correspondante.

Les informations saisies dans cette page seront prises en compte en cliquant sur le bouton « OK » de la page.

**Remarque** : Ces informations apparaissent dans la fenêtre de sélection de projet pour le projet courant.

**Bouton « Adresse IP distante »**

Ce bouton ouvre une fenêtre de saisie de l'adresse IP par défaut du poste à superviser. C'est cette adresse IP qui sera prise en compte pour rafraîchir les expressions si aucune autre adresse n'est spécifiée au niveau des paramètres de la page de supervision ou au niveau de l'expression. Cette adresse est sauvegardée dans le fichier de configuration du projet (APIGRAF.INI).

Dans le cas d'une utilisation de l'atelier de supervision conjointe avec Optima PLC, l'adresse IP à renseigner est : Local\_Host ce qui correspond au poste local ( adresse IP : 127.0.0.1 )

**Option "Clavier virtuel automatique"**

Le clavier virtuel est utilisé sur les écrans tactiles qui ne sont pas munis de clavier. C'est un clavier qui s'ouvre à l'écran à coté de la fenêtre de saisie et qui permet à l'utilisateur de saisir des caractères alphanumériques en cliquant sur les boutons de ce clavier.

Cette option permet d'ouvrir le clavier virtuel de manière automatique dès que l'utilisateur clique sur une fenêtre de saisie.

Exemple : Clavier virtuel alphanumérique



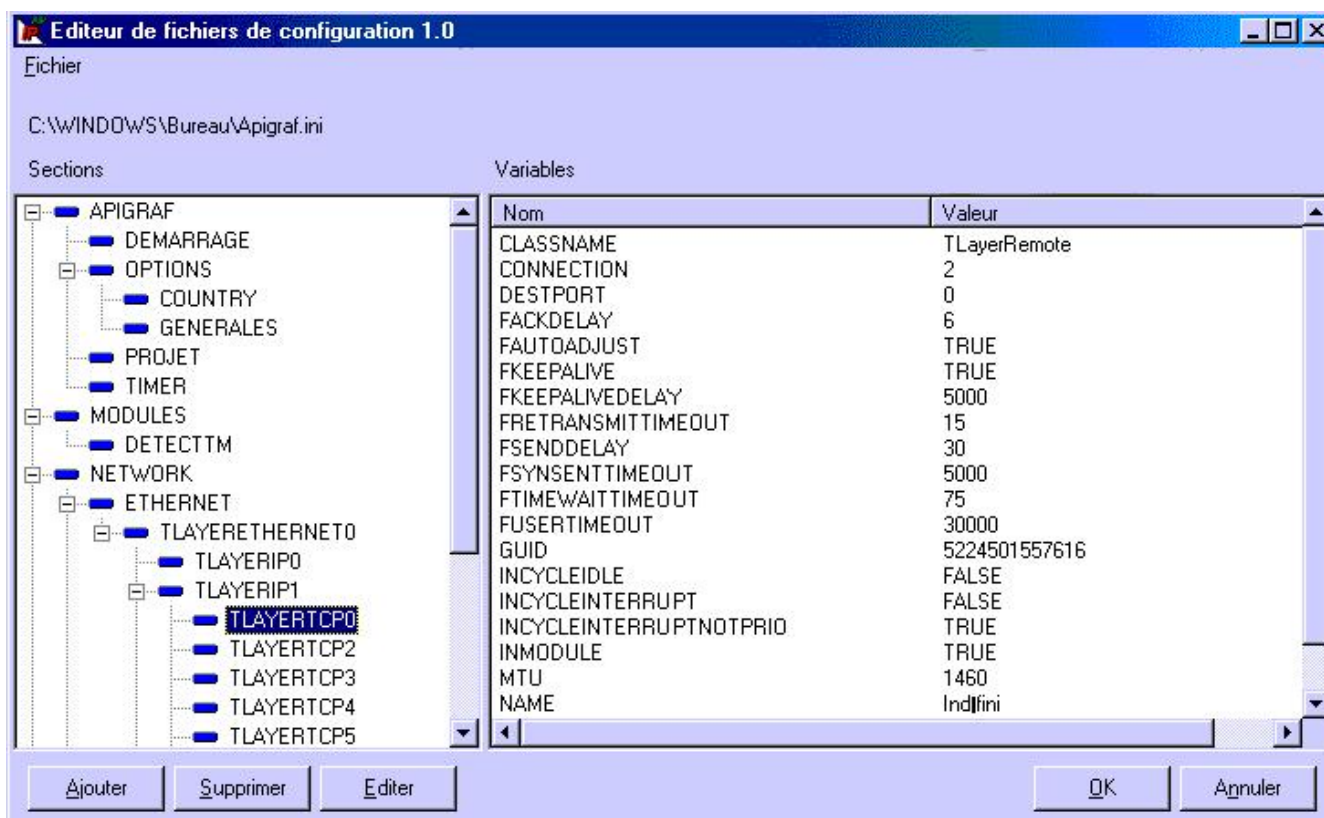
Pour chaque fenêtre de saisie, il est ensuite possible de définir quel type de clavier virtuel sera ouvert : alphanumérique ou numérique et également de spécifier si une fenêtre de saisie doit ou non ouvrir ce clavier virtuel.

### 3.1.8.2 Menu Options - Edition configuration

Ce menu lance l'éditeur de fichiers de configuration « APINIVIEWER ». Cet outil permet de visualiser et de gérer les fichiers de configuration APIGRAF Win et APIGRAF PLC.

Ce menu n'est pas disponible si l'atelier de supervision a été démarré depuis Optima PLC.

La page principale de l'éditeur a la forme suivante :



L'éditeur permet d'ajouter ou de supprimer des sections pour les fichiers de configuration provenant d'APIGRAF WIN ou d'APIGRAF PLC. Pour les fichiers provenant d'APIGRAF PLC, il est possible d'ajouter ou de supprimer des items dans les différentes sections.

### 3.1.8.3 Menu Options - Gestion des utilisateurs

Permet d'activer ou de désactiver la gestion des utilisateurs. Si la gestion des utilisateurs est activée alors le menu apparaît avec un symbole de validation.

La gestion des utilisateurs permet de restreindre l'ouverture de certaines pages de supervision en fonction de

l'utilisateur.

Voir [Gestion des utilisateurs](#) <sup>315</sup>

### 3.1.9 Menu Utilisateurs

#### 3.1.9.1 Changer d'utilisateur

Permet de se définir comme utilisateur courant à condition de posséder un nom d'utilisateur et un mot de passe associé.



Renseigner le nom de l'utilisateur et le mot de passe et valider avec le bouton "OK".

#### 3.1.9.2 Changer le mot de passe

Permet de se changer son mot de passe. L'ancien mot de passe est demandé pour définir le nouveau mot de passe.



Attention : le nouveau mot de passe doit être confirmé et entré dans les 2 derniers champs de saisie de la boîte de dialogue.

#### 3.1.9.3 Liste des utilisateurs

Permet de consulter la liste des utilisateurs et en fonction des droits de créer ou de supprimer des utilisateurs.



### 3.1.10 Menu Version

Affiche les informations concernant la version du logiciel.

Ce menu n'est pas disponible si l'atelier de supervision est démarré depuis Optima PLC.

### 3.1.11 Menu Aide

Permet d'ouvrir l'aide en ligne.

## 3.2 Création de pages de supervision

### 3.2.1 Principes généraux

Une page de supervision graphique est stockée dans un fichier dont l'extension par défaut est .PSGW ( Page de supervision graphique WINDOWS).

Au travers de divers objets graphiques, une page permet de superviser des expressions qui sont évaluées sur une application OPTIMA PLC, une application APIGRAF ou sur n'importe quel serveur OPC.

Dans le cas d'un serveur OPC, on spécifie le nom du serveur OPC sur lequel se connecter.

Dans le cas d'un poste OPTIMA PLC ou APIGRAF, on spécifie l'adresse IP du poste.

Une expression peut être une variable automate, une combinaison de variables automates, une expression logique ou encore une variable utilisateur définie sur le poste distant à l'aide de la commande « EnregistreMot »

#### **Exemples d'expressions valides :**

R2

R2+R3

R3 + R4 > R5

SA2#SA3

(SA2#SA3)&L10

...

...

Pour chaque variable, on peut définir le nom du serveur OPC ou l'adresse IP du poste OPTIMA PLC ou APIGRAF sur lequel sera évaluée la variable.

Si pour une variable, on a défini à la fois une adresse IP et à la fois un nom de serveur OPC alors c'est l'adresse IP qui va être prise en compte.

#### **Définition de l'adresse IP de l'expression :**

Ces variables sont supervisées sur des postes dont l'adresse IP peut être précisée à plusieurs niveaux :

##### Adresse IP du projet par défaut :

L'adresse IP par défaut du projet est saisie dans le menu « OPTIONS » (Voir H.3.8 )

Cette adresse qui est prise en compte si aucune adresse IP n'est précisée au niveau de la page ou au niveau de l'expression.

##### Adresse IP par défaut de la page de supervision :

Si l'adresse IP par défaut de la page est non vide ( VOIR champ de saisie de cette adresse (H.4.3.6) ), c'est cette adresse qui sera prise en compte si aucune adresse IP n'est définie au niveau de l'expression.

##### Adresse IP définie dans un panneau, une page d'un classeur à onglets, d'une zone de défilement.

Certains objets comme le panneau peuvent contenir d'autres objets. Dans ce cas, on peut définir dans cet objet une adresse IP à utiliser par défaut pour tous les objets qu'il contient. L'adresse IP ainsi définie prévaudra alors celle qui est définie dans la page de supervision ou dans le projet.

Adresse IP définie dans l'expression :

Si une adresse IP est définie dans l'expression, c'est celle-ci qui sera prise en compte.

**Exemple :**

R12            Expression de R12  
!193.0.1.1 !R12    Expression de R12 supervisé à l'adresse 193.0.1.1

Si l'utilisation de l'atelier de supervision est conjointe avec Optima PLC, il n'y a généralement pas de poste distant à superviser, l'adresse IP à renseigner par défaut dans le projet et dans les pages de supervision est : Local\_Host (ce qui correspond à l'adresse du PC local à l'adresse 127.0.0.1)

A chaque création de page de supervision depuis Optima PLC, l'adresse Local\_Host est renseignée automatiquement dans le champ "Adresse IP par défaut" des pages de supervision.

**Définition du serveur OPC de l'expression :**

Ces variables sont supervisées sur des serveurs OPC dont le nom peut être précisé à plusieurs niveaux :

Serveur OPC par défaut de la page de supervision :

Si le nom du serveur OPC par défaut de la page est non vide ( VOIR champ de saisie dans l'onglet "Options" de la fenêtre de paramétrage de la page de supervision ), c'est ce nom de serveur OPC qui sera pris en compte à moins qu'un autre nom de serveur soit précisé dans l'expression (ou bien une adresse IP).

Serveur OPC défini dans un panneau, une page d'un classeur à onglets, d'une zone de défilement.

Certains objets comme le panneau peuvent contenir d'autres objets. Dans ce cas, on peut définir dans cet objet le nom du serveur OPC à utiliser par défaut pour tous les objets qu'il contient. Le nom du serveur OPC ainsi défini prévaudra alors sur le nom de serveur OPC défini dans la page de supervision.

Serveur OPC défini dans l'expression :

Si le nom d'un serveur OPC est défini dans l'expression, c'est celui-ci qui sera pris en compte.

**Exemple :**

COMPTEUR1            Expression de Compteur1  
|SERVOPC.PLC|COMPTEUR1    Expression de Compteur1 supervisé sur le serveur OPC : SERVOPC.PLC

### 3.2.2 Création d'une page de supervision

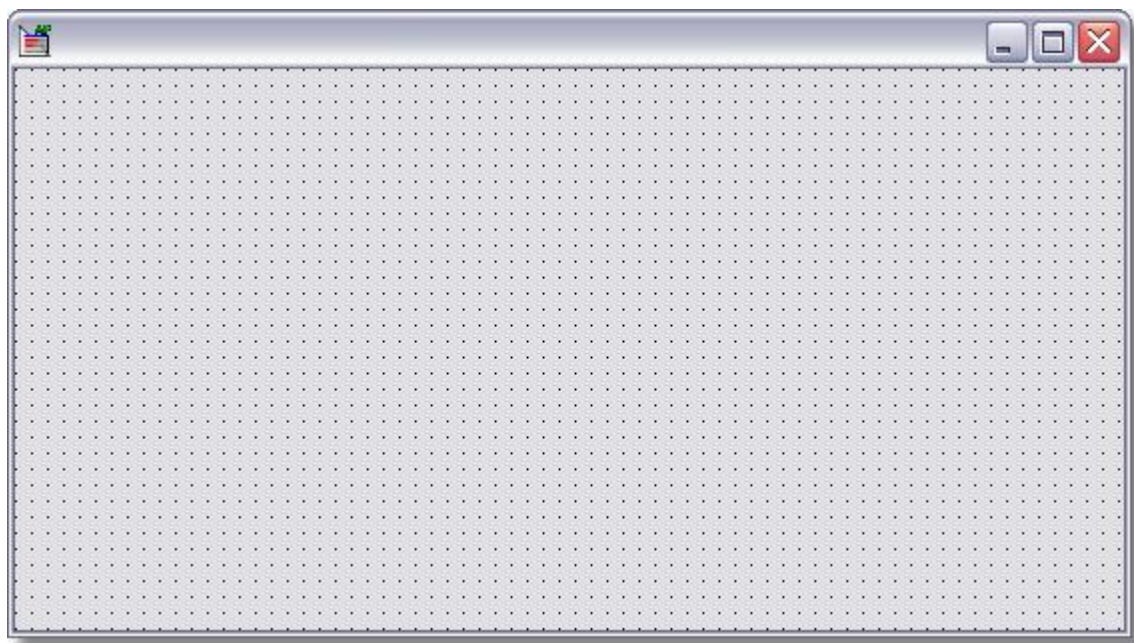
Pour créer une nouvelle page de supervision, utiliser le menu « NOUVEAU » de l'atelier de supervision.

Une nouvelle fenêtre s'affiche, initialement vide. Deux barres de menu « flottantes » s'affichent à l'écran. Ces barres peuvent être déplacées indépendamment de la fenêtre de supervision.

La première barre correspond aux objets disponibles pour la création des pages (Bouton, fenêtres de saisie,...).

La deuxième barre de menu, appelée barre de fonctions, comporte certaines fonctions accessibles depuis l'atelier. Cette barre comporte également l'indication de la position en X et en Y de la souris par rapport à la fenêtre de supervision.

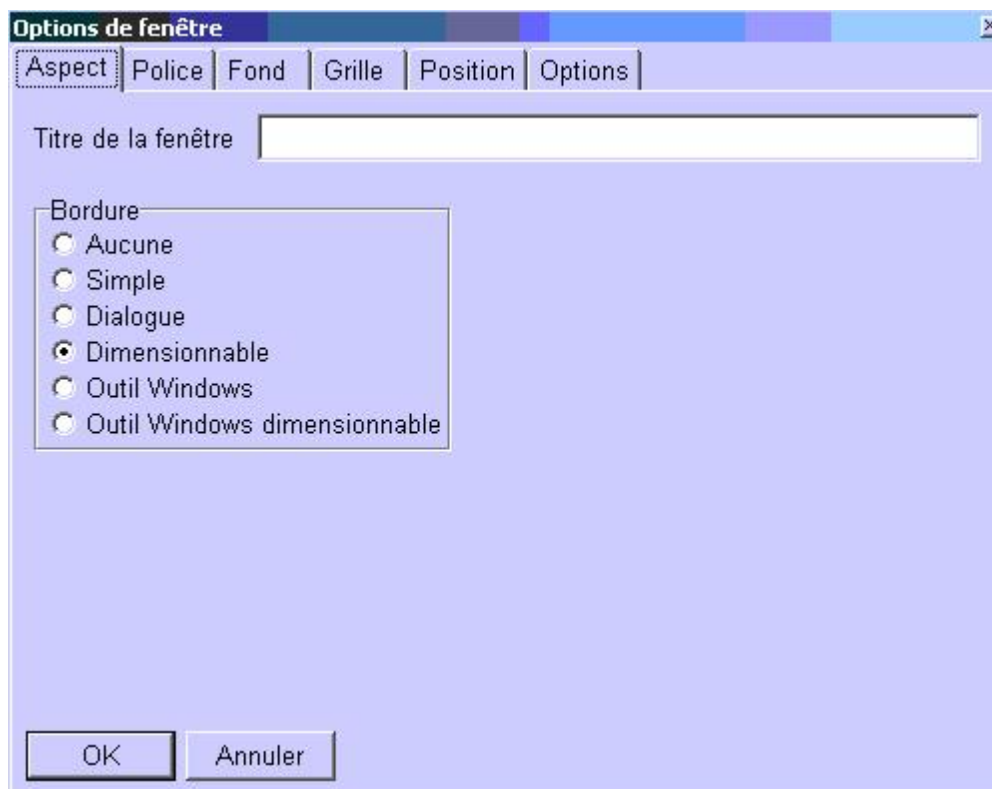
### 3.2.3 Paramétrage d'une page de supervision



Pour modifier l'apparence ou le comportement général d'une page de supervision, placer le curseur de la souris sur celle-ci puis cliquer sur le bouton de droite.

La boîte dialogue « OPTIONS FENETRE » est alors affichée. Cette page présente plusieurs onglets.

#### 3.2.3.1 Aspect



Cet onglet permet de définir l'aspect général de la page de supervision.  
Il comporte les contrôles suivants :

##### Titre Fenêtre

Ce champ permet de définir le titre affiché en haut de la fenêtre de supervision, lors de l'exécution de la page

uniquement. Pendant l'édition de la page, c'est le nom du fichier qui est affiché.

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

### Bordure

Ce contrôle permet de choisir l'aspect général de la bordure de la fenêtre.

***Attention : La bordure sélectionnée n'apparaît que lors de l'exécution de la page en dehors de l'atelier de conception.***

Les différents choix sont :

#### Aucune :

La fenêtre n'a aucune bordure et n'a pas de barre de titre

.

#### Simple :

La fenêtre est standard mais n'est pas redimensionnable avec la souris.

#### Dialogue :

La fenêtre a l'apparence d'une fenêtre de dialogue Windows. (Non redimensionnable, les boutons de maximisation et de minimisation de fenêtre ne sont pas présents dans la barre de titre, il n'y a pas d'icône dans la barre de titre).

#### Dimensionnable :

La fenêtre est standard et est redimensionnable avec la souris.

#### Outils Windows :

La fenêtre a l'apparence d'une barre d'outil Windows mais n'est pas redimensionnable.

#### Outils Windows dimensionnable :

La fenêtre a l'apparence d'une barre d'outil Windows et est redimensionnable.

### 3.2.3.2 Police



Cet onglet permet de spécifier la police par défaut qui sera utilisée pour tous les objets contenus dans la page de supervision.

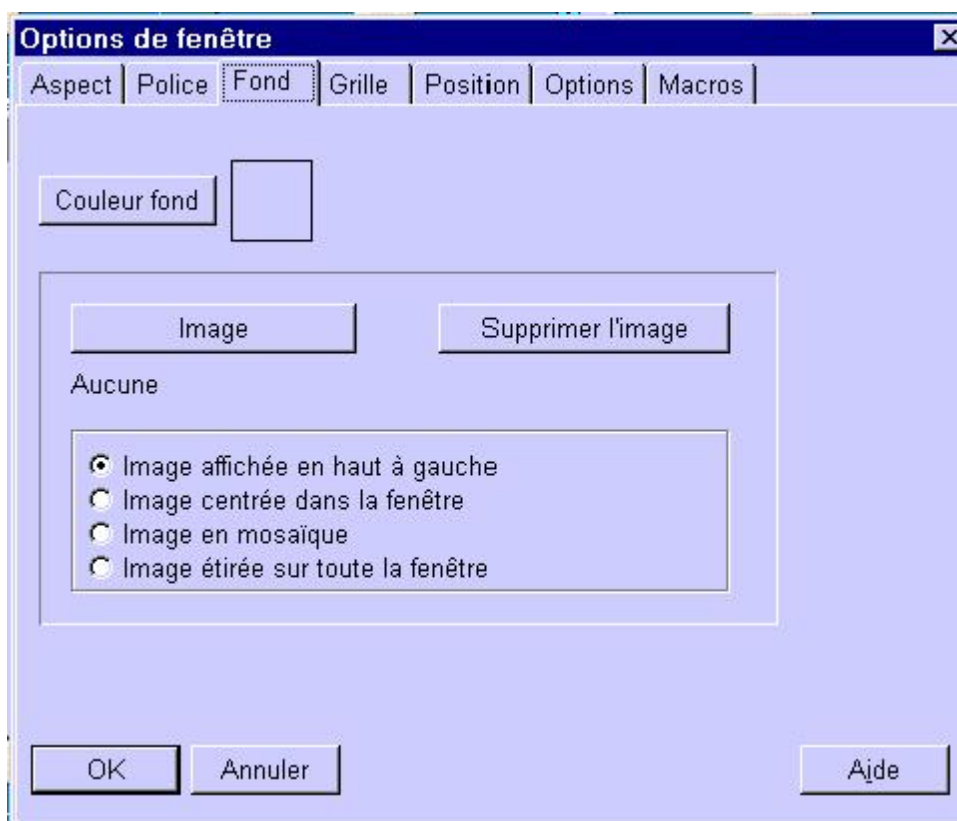
Une fenêtre montre l'aspect de la police par défaut qui est actuellement définie.

Pour changer la police par défaut, cliquer sur le bouton « Police par défaut des objets », la fenêtre de paramétrage



standard des polices s'affiche alors à l'écran.

### 3.2.3.3 Fond



La boîte de dialogue « Fond » comporte les contrôles suivants :

#### **Couleur Fond**

Permet de sélectionner la couleur de fond de la page de supervision

#### **Image**

Ce contrôle permet de spécifier un fichier image qui sera donc affiché en fond sur la page de supervision. Les formats d'image suivants sont acceptés : \*.BMP ; \*.ICO ; \*.EMF ; \*.WMF

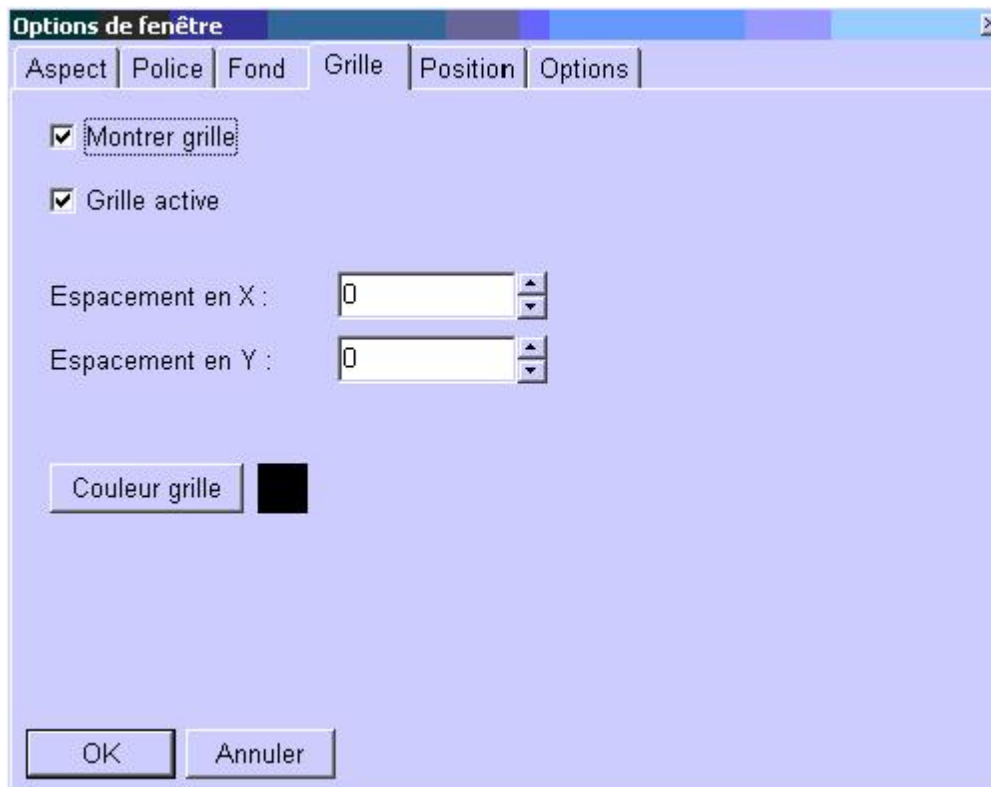
#### **Supprimer l'image**

Supprime l'image de fond de la page de supervision.

#### **Position de l'image**

Le dernier contrôle permet de spécifier l'agencement de l'image dans la page de supervision.

### 3.2.3.4 Grille



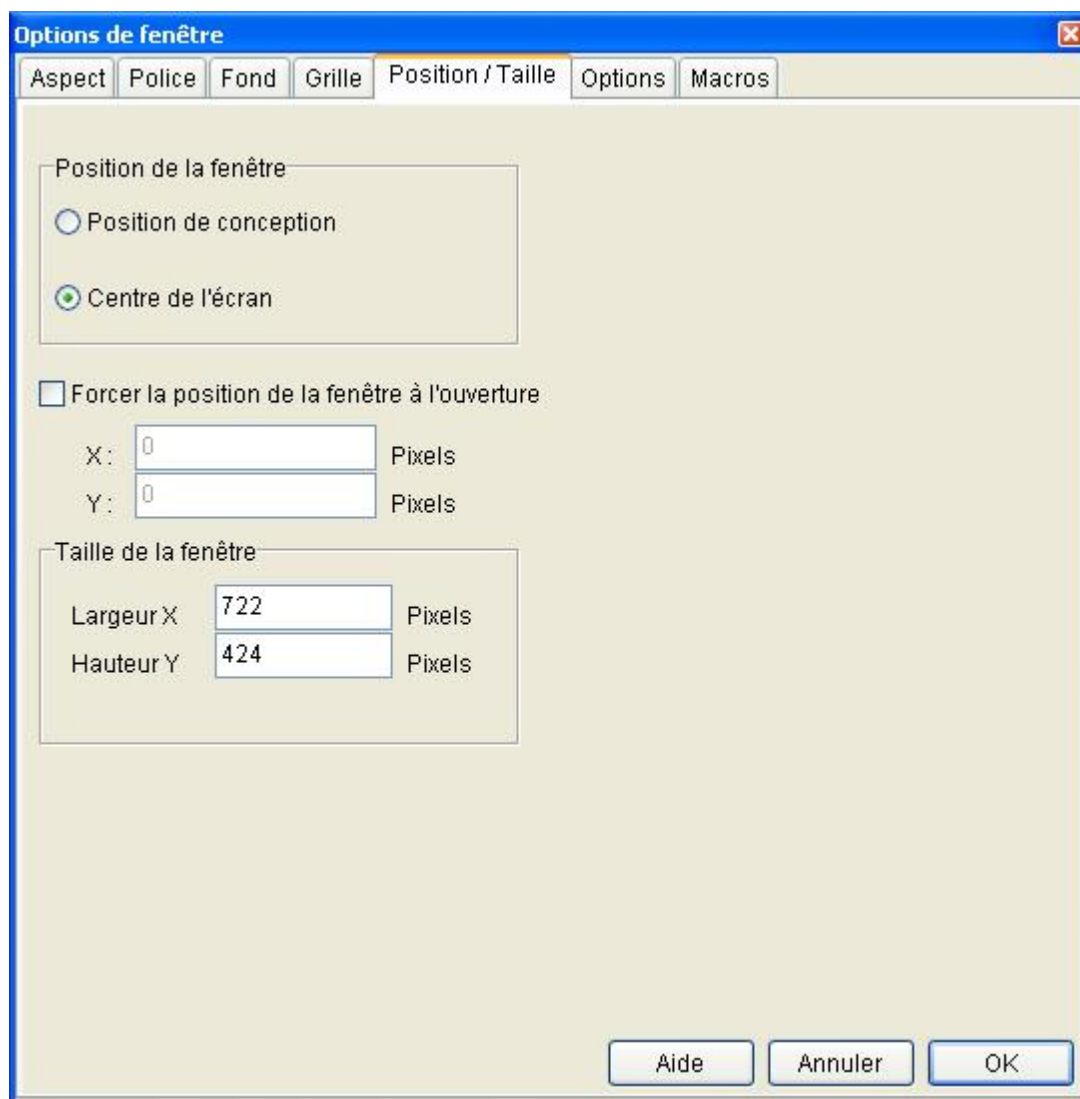
Cet onglet permet de définir les paramètres de la grille utilisée pendant l'édition des objets de supervision. Cette grille a pour but de faciliter la création d'objets situés à des positions régulières.

Les objets insérés sont placés aux positions de la grille, leur taille est également ajustée à la résolution définie lors des modifications de taille.

La résolution en X et en Y peut être définie, ainsi que la couleur des points de la grille. La grille n'est réellement affichée que si deux résolutions X et Y sont au moins égales à 2, et si l'option « Montrer grille » est activée. De même, la grille n'est active que si l'option « Grille active » est activée.

La grille n'apparaît bien entendu pas lors de l'exécution de la page.

### 3.2.3.5 Position / Taille



Cet onglet permet de choisir la position et la taille de la fenêtre lors de son exécution :

#### Position de la fenêtre :

- Position de conception (Position donnée à la page dans l'atelier de supervision)
- Centre de l'écran

#### Option "Forcer la position de la fenêtre à l'ouverture"

Si cette option est cochée, la fenêtre sera positionnée à la position X,Y définie en dessous.  
dans les champs de saisie X et Y.

#### Taille de la fenêtre :

Affiche la taille (Largeur et hauteur) actuelle de la fenêtre. Il est possible de modifier directement les valeurs qui sont exprimées en pixels.

### 3.2.3.6 Options

**Options de fenêtre**

Aspect Police Fond Grille Position / Taille **Options** Macros

Adresse IP apigraf/OPTIMA PLC par défaut :

Serveur OPC par défaut :

☐ Adresse IP évaluée par une expression

Expression de l'adresse IP :

☐ Fenêtre initialement maximisée  
☐ Fenêtre initialement minimisée  
☐ Barres de défilement si nécessaire  
☐ Exécution exclusive  
☐ Fenêtre mode plein écran  
☐ Fenêtre toujours en avant plan

☐ Fermeture automatique de la fenêtre  
 Au bout de :  Secondes

La fenêtre peut être  
☒ Maximisée  
☒ Réduite  
☒ Fermée  
☐ Ouverte simultanément plusieurs fois

Numéro du mot de passe pour ouvrir la page :   
 Niveau utilisateur minimum pour ouvrir la page : AUCUN   
 (Effectif uniquement si la gestion d'utilisateur est activée)

Aide Annuler OK

Cette zone permet de définir certaines options de fonctionnement de la fenêtre ( options prises en compte lors de l'exécution et non en mode édition) :

#### Saisie de l'adresse IP apigraf PLC par défaut

Cette fenêtre permet de saisir l'adresse IP du poste OPTIMA PLC ou APIGRAF PLC à superviser par défaut dans la page. Si le champ est vide, c'est l'adresse IP par défaut du projet qui sera utilisée. ( A moins que l'expression contienne une adresse IP )

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

#### Saisie du nom de serveur OPC par défaut

Cette fenêtre permet de saisir le nom du serveur OPC à superviser par défaut dans la page. Attention : si une adresse IP est définie dans le champ du dessus, le nom du serveur OPC ne sera pas pris en compte.

#### Fenêtre initialement maximisée

La fenêtre apparaîtra en mode plein écran dès son exécution.

#### Fenêtre initialement minimisée

La fenêtre apparaîtra minimisée dès son exécution.

#### Barres de défilement si nécessaire

Cette option permet d'indiquer si le contenu de la fenêtre peut être « défilé » par l'utilisateur. Pour cela, des barres de défilement apparaissent en bas et à droite de la fenêtre, ce qui permet de déplacer la zone visible de la fenêtre.

Ces barres n'apparaissent cependant que si le contenu de la fenêtre n'est pas entièrement visible.

#### Exécution exclusive

Si cette option est cochée, lorsque la fenêtre sera affichée, aucune autre action ne sera possible (passage à une

autre fenêtre déjà ouverte impossible, etc.), et cela jusqu'à sa fermeture.

Si une autre page de supervision est appelée depuis une page à exécution exclusive, cette autre page doit nécessairement être à exécution exclusive.

### **Fenêtre toujours en avant plan**

Si cette option est cochée, la fenêtre apparaîtra toujours en avant plan c'est à dire au dessus des fenêtres qui ne possèdent pas cette option. L'utilisateur a tout de même la possibilité d'activer une fenêtre qui se trouve en dessous.

### **La fenêtre peut être :**

#### Maximisée :

Si cette option n'est pas cochée, le bouton de maximisation de fenêtre de Windows ne sera pas présent sur la barre de titre de la fenêtre, celle-ci ne pourra donc pas être agrandie.

#### Réduite :

Si cette option n'est pas cochée, le bouton de minimisation de fenêtre de Windows ne sera pas présent sur la barre de titre de la fenêtre, celle-ci ne pourra donc pas être réduite.

#### Fermée :

Si cette option est cochée, le symbole Windows de fermeture de fenêtre ne sera pas présent sur la barre de titre de la fenêtre, le seul moyen de fermer la fenêtre sera donc de taper ALT+F4.

#### Ouverte simultanément plusieurs fois :

Si cette option est cochée, à chaque nouvel appel de la page, une nouvelle page sera ouverte.

### **Fermeture automatique de la fenêtre :**

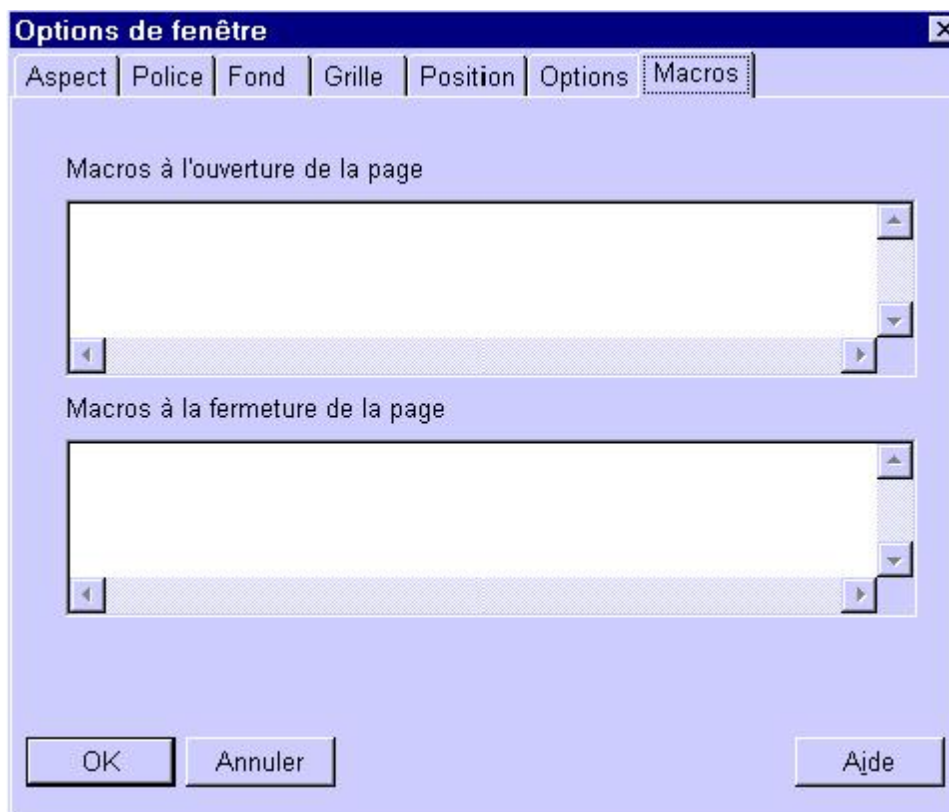
Si cette option est cochée et que le temps spécifié en secondes en dessous est différent de 0 alors la fenêtre se fermera automatiquement avec n secondes selon la valeur entrée en nombre de secondes après son ouverture.

Cette option peut être utile pour créer un message temporaire qui ne s'affiche que quelques secondes.

### **Numéro du mot de passe pour ouvrir la page :**

Permet de spécifier un numéro mot de passe pour pouvoir ouvrir la page.  
Ce mot de passe est défini en utilisant la macro locale ChnMDP.

### 3.2.3.7 Macros



#### Boîte de saisie « Macros à l'ouverture de la page »

Permet de spécifier les macros qui seront exécutées à l'ouverture de la page de supervision.

Attention : Une macro par ligne !

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

#### Boîte de saisie « Macros à la fermeture de la page »

Permet de spécifier les macros qui seront exécutées à la fermeture de la page de supervision.

Attention : Une macro par ligne !

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

### 3.2.4 Barres d'outils de l'atelier de supervision

Ces barres de menu permettent d'avoir en permanence les outils disponibles pendant l'édition d'une page de supervision.

La première barre de menu permet de créer des objets de supervision :



La deuxième barre de menu offre diverses fonctions :



**Bouton 1** : Appelle la page de paramétrage de la page de supervision en cours. Equivaut à un clic droit de la souris sur la page de supervision

**Bouton 2** : Fonction zoom de l'objet : permet d'entrer un facteur de zoom pour l'objet sélectionné en pourcentage

**Bouton 3** : Exécution de la page de supervision : passe en mode exécution sur la page de supervision en cours. Le retour au mode édition se fait en fermant la page de supervision en cours d'exécution.

**Bouton 4** : Sélection du mode « rafraîchissement des données » : Permet la mise à jour des objets suivant leurs variables associées en mode édition. (La mise à jour est forcément établie lors de l'exécution de la page)

**Bouton 5** : Suppression du ou des objets sélectionnés.  
Cette barre de menu affiche en plus sur sa partie gauche, les coordonnées du curseur souris dans le référentiel de la page de supervision.

### 3.2.5 Manipulation des objets de supervision

Les manipulations suivantes sont communes à tous les objets de supervision.

Elles permettent de créer, modifier, déplacer, etc. ces différents objets.

#### 3.2.5.1 Création d'un objet de supervision

Pour créer un nouvel objet, utiliser la barre d'outils des objets :

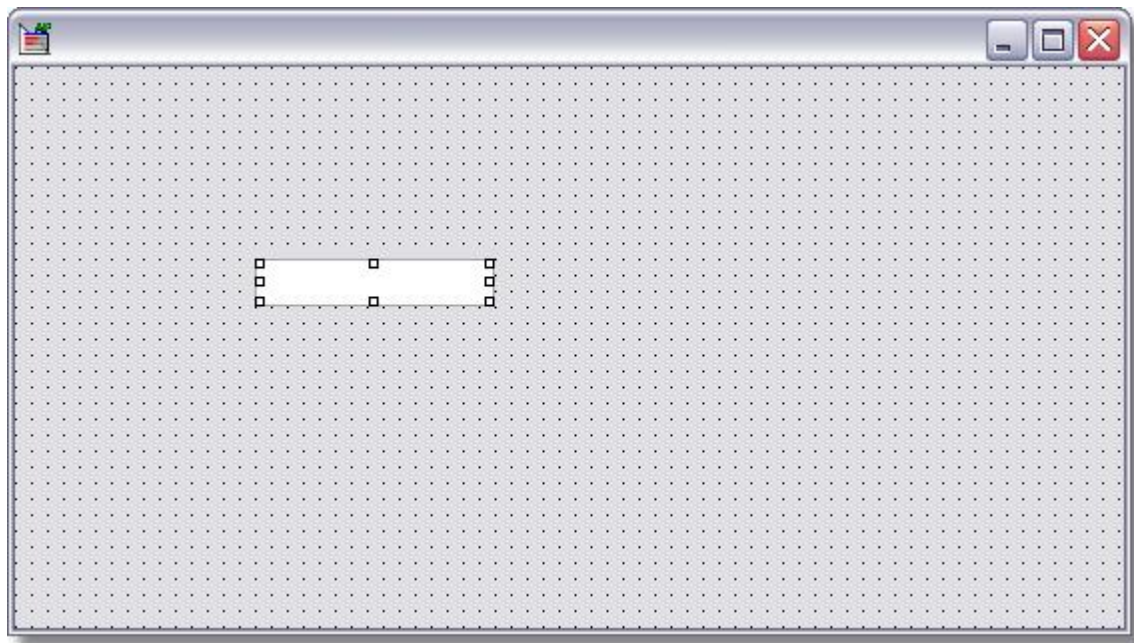


Cliquer à l'aide du bouton gauche de la souris sur l'onglet correspondant au type d'objet désiré puis cliquer sur le bouton correspondant à l'objet à créer (sur la barre des objets).

Le bouton correspondant apparaît alors enfoncé.



Déplacer le curseur de la souris sur l'endroit de la page où l'objet doit être créé et appuyer une nouvelle fois sur le bouton gauche de la souris. L'Objet est dès lors créé.



### 3.2.5.2 Sélection d'un objet ou d'un groupe d'objets de supervision

Un objet de supervision doit être sélectionné avant de pouvoir être modifié.

Lorsqu'un objet est sélectionné, des « plots » blancs apparaissent sur son contour. Plusieurs objets peuvent être sélectionnés simultanément.

#### Sélection par désignation

Un clic sur le bouton gauche de la souris sur l'objet à sélectionner a pour effet de désélectionner les objets qui l'étaient auparavant et de sélectionner l'objet cliqué.

Si le clic est effectué alors que la touche SHIFT du clavier est enfoncée, l'objet est ajouté à la sélection précédente.

#### Sélection par rectangle englobant

Il est possible de sélectionner en une seule opération tous les objets situés dans une zone rectangulaire.

Pour cela, cliquer avec la souris sur l'un des coins de cette zone (qui ne doit pas être sur un objet), puis, sans relâcher le bouton de la souris, déplacer le curseur jusqu'à l'autre coin.

Les objets situés dans ce rectangle sont alors sélectionnés. Les objets partiellement compris dans ce rectangle sont aussi sélectionnés.

### 3.2.5.3 Déplacement et changement de taille d'un objet de supervision

#### Déplacement

Avant d'être déplacé, un objet doit être sélectionné.

Pour déplacer un objet de supervision, il suffit de cliquer dessus pour le sélectionner, puis de cliquer à nouveau et, sans relâcher le bouton de la souris, de déplacer l'objet jusqu'à la position souhaitée.

Si une grille de résolution est définie, l'objet ne peut être placé qu'aux positions définies par la grille.

Si plusieurs objets sont sélectionnés, ils sont déplacés simultanément.

#### Agrandissement ou diminution

Pour être diminué ou agrandi, un objet doit être sélectionné.

Le changement de dimension est effectué en « tirant » sur un des « plots » qui entoure l'objet sélectionné.

Placer le curseur de la souris sur un plot correspondant au changement de taille souhaité (côté ou coin), cliquer sur le plot sans relâcher, puis déplacer le curseur jusqu'à ce que l'objet prenne la taille souhaitée.

#### Saisie directe de la position et de la taille

Dans certains cas, la position et la taille d'un objet doivent être saisis numériquement au clavier pour obtenir un résultat plus précis.



Pour cela, cliquer sur l'objet à l'aide du bouton droit de la souris pour faire apparaître le menu contextuel lié aux objets.

Dans ce menu choisir « Position ». Une boîte de dialogue apparaît alors permettant de saisir numériquement (en pixels) les valeurs suivantes : PositionX, PositionY, Largeur, Hauteur.

#### Position et taille de l'objet définis par des variables

Il est possible de définir les positions et les tailles des objets par des variables automates. Dans ce cas, les positions et taille de tels objets varieront avec les variables associées (Registres).

Il faut alors afficher la boîte de dialogue permettant de saisir taille et position de l'objet ( clic droit de la souris sur l'objet et sélectionner « position » dans le menu contextuel). Puis, entrer l'expression (R30 par exemple) dans le champ qui doit être défini par cette variable.

### 3.2.5.4 Changement de taille d'un objet ou d'un groupe d'objets à l'aide de la fonction zoom

Pour changer la taille d'un ou plusieurs objets, sélectionner d'abord tous les objets à redimensionner.

Ensuite, cliquer sur le bouton « Zoom » de la barre d'outils « FONCTIONS » (Voir description chapitre A4.4). Une fenêtre de dialogue apparaît :

#### Boîte de dialogue changement d'échelle



Les champs sont les suivants :

#### Saisie du Zoom

La fenêtre de saisie de la boîte de dialogue permet d'indiquer le zoom souhaité en pourcentage.

La valeur par défaut est 100% (pas de changement de taille).

#### Option « Modification de la taille du texte de l'objet »

Cette option permet quand elle est cochée, de zoomer le texte de l'objet en plus de l'objet lui-même.

Ceci a pour effet de modifier la taille de la police de caractère associée au texte de l'objet.

### 3.2.5.5 Suppression d'objets de supervision

Sélectionner le ou les objets à supprimer.

Cliquer ensuite sur le symbole corbeille de la barre d'outils « FONCTIONS » pour effacer le ou les objets sélectionnés. ( symbole N°5 )

Il est aussi possible d'utiliser la touche « Suppr » du clavier, qui a le même effet que le clic sur la corbeille.



### 3.2.5.6 Mise en avant plan ou en arrière plan d'un objet

#### Mise en avant plan d'un objet

Sélectionner l'objet à mettre en avant plan.

Cliquer sur l'objet avec le bouton droit de la souris pour faire apparaître le menu contextuel associé aux objets. Dans le menu, choisir « Mettre à l'avant plan ».



#### Mise en arrière plan d'un objet.

Sélectionner l'objet à mettre en arrière plan.

Cliquer sur l'objet avec le bouton droit de la souris pour faire apparaître le menu contextuel associé aux objets. Dans le menu, choisir « Mettre à l'arrière plan ».



**Attention :** Certains objets (Fenêtre de saisie) sont par défaut en avant plan et ne peuvent être placés en arrière plan.

### 3.2.5.7 Copie d'objets à l'aide de la souris

Il est possible d'effectuer un copier coller sur un ou plusieurs objets à l'aide de la souris.

Il faut tout d'abord sélectionner le ou les objets à copier. Maintenir ensuite la touche « CONTROL » enfoncée.

Cliquer avec le bouton droit de la souris en le maintenant enfoncé sur un des objets à copier et déplacer la copie des objets à l'endroit désiré.

Relâcher le bouton droit de la souris et relâcher la touche « CONTROL » du clavier. La copie est effectuée.

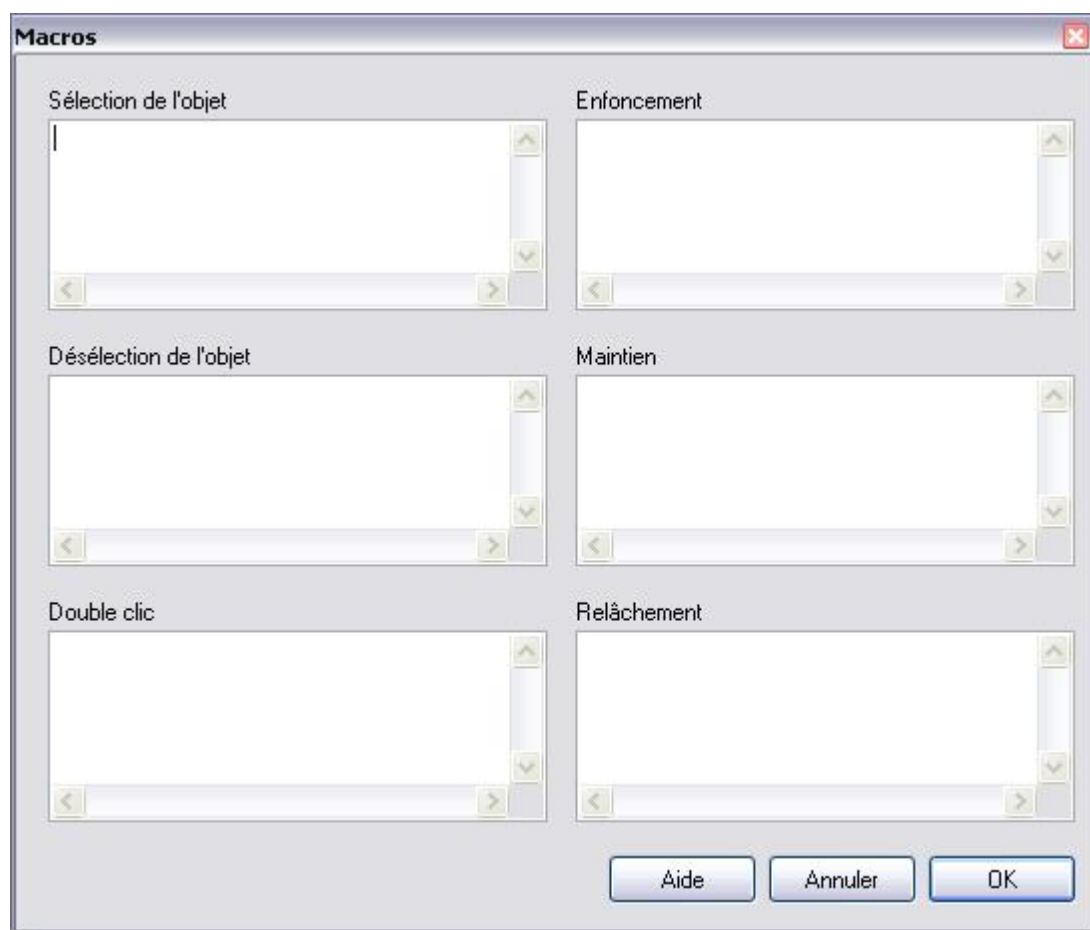
### 3.2.5.8 Saisie des macros pour les objets

Les macros sont des fonctions qui sont exécutés lors de certains événements qui se produisent sur les objets. Par exemple, un double clic sur une image.

#### Pour définir les macros associées à un objet :

Sélectionner l'objet pour lequel on veut définir des macros.

Cliquer sur l'objet avec le bouton droit de la souris pour faire apparaître le menu contextuel lié aux objets. Dans le menu, choisir « MACROS », la fenêtre générale de saisie des macros s'ouvre :



Les macros saisies peuvent être exécutées à plusieurs stades (voir les différentes fenêtres de saisie des macros), notamment lors de :

**Sélection de l'objet**

**Désélection de l'objet**

**Double clic sur l'objet**

**Enfoncement de l'objet (pour les boutons)**

**Maintien enfoncé de l'objet (pour les boutons)**

**Relâchement de l'objet (pour les boutons)**

Pour plus d'information sur les macros possibles, se reporter aux chapitres correspondants.

## **3.2.6 Objets usuels pour la création de pages de supervision**

### **3.2.6.1 Objets de l'onglet STANDARD**

#### **3.2.6.1.1 Texte**

Ce type d'objet permet d'afficher un texte statique. Il est utilisé pour créer les fonds d'écran.

#### **Boîte de dialogue paramètre Texte**



Les champs de saisie de cette boîte sont :

#### **Texte**

Saisie du texte de l'objet.

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

#### **Option Taille automatique**

Cette option quand elle est cochée permet d'ajuster automatiquement la taille de l'objet texte par rapport au texte qui se trouve dans l'objet.

#### **Option « Multi ligne automatique »**

Si cette option est cochée, le texte sera automatiquement coupé sur plusieurs lignes si le texte entier ne tient pas sur une seule ligne.

#### **Caractères**

Cette option permet de choisir la police de caractères à utiliser pour le texte affiché. Elle provoque l'affichage de la boîte de dialogue « Police de caractères ».

#### **Couleur de fond**

Cette option permet de choisir la couleur de fond de l'objet Texte.

#### **Option fond transparent**

Si cette option est cochée, le texte apparaîtra sans fond. Si on place un texte à fond transparent sur une image par exemple, le texte sera « fondu » dans l'image.

#### **Alignement horizontal**

Permet de choisir l'alignement horizontal (gauche, droit ou centré)

#### **Alignement vertical**

Permet de choisir l'alignement vertical (gauche, droit ou centré)

#### **Bouton « Hint »**

Ce bouton permet de définir les paramètres de « Hint » de l'objet. Le texte « Hint » que l'on définit est le texte qui apparaît sur fond jaune lorsque l'on passe le curseur de la souris sur l'objet.

Quand on clique sur le bouton « Hint », la fenêtre suivante s'ouvre :



Cette fenêtre présente deux contrôles :

**Option « Afficher le Hint »**

Si cette option n'est pas cochée, le texte du Hint n'apparaîtra pas quand le curseur de la souris passera sur l'objet même si celui-ci est défini. Il faut donc cocher cette option si l'on veut utiliser le « Hint » de l'objet.

**Fenêtre de saisie « Hint »**

Cette fenêtre permet d'entrer le texte du Hint. Ce texte peut être un paramètre (exemple {1}).

**Option « Texte dynamique »**

Cocher cette option fait apparaître une fenêtre de saisie de l'expression.

Le texte sera alors mis à jour dynamiquement avec la valeur de l'expression précisée dans le champ correspondant.

**« Expression »**

Entrer l'expression qui peut être de n'importe quel type (Chaîne, registre, flag, etc..).

Le texte sera alors mis à jour dynamiquement avec la valeur de l'expression.

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

**Expression de neutralisation**

Permet de spécifier la variable ou l'expression de neutralisation (Celle-ci doit obligatoirement être de type booléen). L'objet texte sera invisible chaque fois que la variable ou l'expression sera égale à true.

*Exemple : SA10*

Il est possible de spécifier l'adresse IP distante du noyau APIGRAF PLC dans l'expression. Si c'est le cas, c'est cette adresse qui sera prise en compte pour le rafraîchissement.

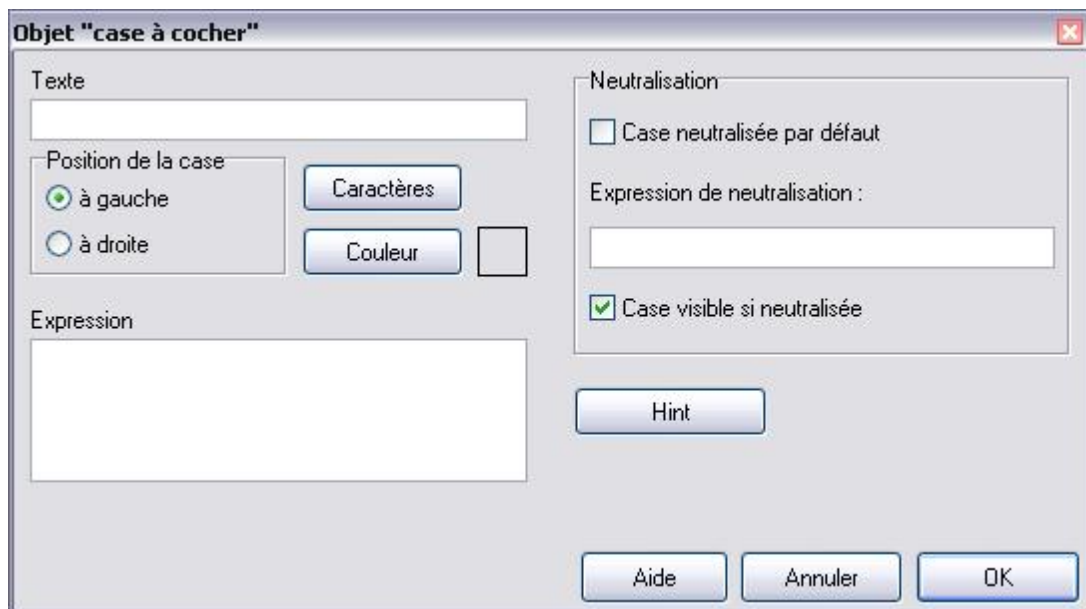
*Exemple : !193.0.1.1 !SA10*

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

### 3.2.6.1.2 Case à cocher

L'objet case à cocher permet de contrôler une variable automate de type booléen.

**Boîte de dialogue paramètre case à cocher**



Les paramètres sont les suivants :

#### Texte

Permet de saisir le texte à associer à l'objet case à cocher. Ce texte sera affiché à côté de la case à cocher.

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

#### Position de la case

Indique la position du texte par rapport à la case à cocher (à gauche ou à droite).

#### Caractères

Cette option permet de choisir la police de caractères à utiliser pour le texte affiché dans l'objet « Case à cocher ». Elle provoque l'affichage de la boîte de dialogue « Police de caractères ».

#### Couleur

Permet de sélectionner la couleur de fond de l'objet case à cocher.

#### Expression

Indiquer dans ce champ la variable automate à contrôler. Cette variable doit être de type booléen (SA,L...).

*Exemple : L10*

Il est possible de spécifier l'adresse IP distante du noyau APIGRAF PLC dans l'expression. Si c'est le cas, c'est cette adresse qui sera prise en compte pour le rafraîchissement.

*Exemple : !193.0.1.1 !L10*

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

#### Case neutralisée par défaut

Si cette option est cochée, il ne sera pas possible de modifier l'état de la case à cocher en cliquant dessus. La case permettra seulement d'afficher l'état de la variable associée. De plus la case cochée apparaîtra grisée.

#### Expression de neutralisation

Permet de spécifier la variable ou l'expression de neutralisation (Celle-ci doit obligatoirement être de type booléen). La case à cocher sera neutralisée chaque fois que la variable ou l'expression sera égale à true.

*Exemple : SA10*

Il est possible de spécifier l'adresse IP distante du noyau APIGRAF PLC dans l'expression. Si c'est le cas, c'est cette adresse qui sera prise en compte pour le rafraîchissement.

Exemple : !193.0.1.1 !SA10

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

#### Option « Case visible si neutralisé »

Si cette option est cochée, la case à cocher sera visible quand elle sera neutralisée. Cependant, elle apparaîtra grisée et l'opérateur ne pourra pas l'actionner.

Si cette option n'est pas cochée, la case à cocher sera invisible dès qu'elle sera neutralisée.

#### Bouton « Hint »

Ce bouton permet de définir les paramètres de « Hint » de l'objet. Le texte « Hint » que l'on définit est le texte qui apparaît sur fond jaune lorsque l'on passe le curseur de la souris sur l'objet.

Quand on clique sur le bouton « Hint », la fenêtre suivante s'ouvre :



Cette fenêtre présente deux contrôles :

#### Option « Afficher le Hint »

Si cette option n'est pas cochée, le texte du Hint n'apparaîtra pas quand le curseur de la souris passera sur l'objet même si celui-ci est défini. Il faut donc cocher cette option si l'on veut utiliser le « Hint » de l'objet.

#### Fenêtre de saisie « Hint »

Cette fenêtre permet d'entrer le texte du Hint. Ce texte peut être un paramètre (exemple {1}).

### 3.2.6.1.3 Radio group

L'objet groupe Radio permet d'effectuer un choix dans une liste d'options.

Dans un groupe radio, une seule option peut être active (différent de l'objet case à cocher). On associe à cet objet une variable automate entière numérique ( de type registre).

Exemple : R12

Il est possible de spécifier l'adresse IP distante du noyau APIGRAF PLC dans l'expression. Si c'est le cas, c'est cette adresse qui sera prise en compte pour le rafraîchissement.

Exemple : !193.0.1.1 !R12

Le registre associé est alors affecté par la position de l'option qui est active en partant du haut avec 0.

Exemple :



Dans cet exemple, si la variable associée est R3, la valeur 0 est affectée à R3. Si le choix 2 était actif, ce serait la valeur 1 qui lui serait affectée.

### **Boîte de dialogue paramètre Groupe radio**

Les champs sont les suivants :

#### **Titre**

Le titre saisi dans ce champ apparaîtra au-dessus de la liste des boutons radio.

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

#### **Caractères**

Ce bouton permet de spécifier la police de caractères du groupe radio quand celui-ci n'est pas neutralisé

#### **Expression**

Indiquer dans ce champ la variable automate à utiliser (de type numérique entier )

*Exemple : R12*

Il est possible de spécifier l'adresse IP distante du noyau APIGRAF PLC dans l'expression. Si c'est le cas, c'est cette adresse qui sera prise en compte pour le rafraîchissement.

*Exemple : !193.0.1.1 !R12*

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

On notera que si la liste ne comporte que deux boutons radios, on pourra utiliser une variable automate de type booléen puisque les valeurs possibles seront 0 ou 1.

#### **Liste des boutons radio**

Saisir directement les désignations des boutons dans leur ordre d'apparition. Valider une saisie avec « ENTREE » pour passer à la ligne suivante.

Il est possible de rentrer des références à des paramètres dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

#### **Expression de neutralisation**



Permet de spécifier la variable ou l'expression de neutralisation (Celle-ci doit obligatoirement être de type booléen). Le groupe radio sera neutralisé chaque fois que la variable ou l'expression sera égale à true.

*Exemple* : SA10

Il est possible de spécifier l'adresse IP distante du noyau APIGRAF PLC dans l'expression. Si c'est le cas, c'est cette adresse qui sera prise en compte pour le rafraîchissement.

*Exemple* : !193.0.1.1 !SA10

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

#### **Option « Visible si neutralisé »**

Si cette option est active, le groupe radio sera visible quand celui-ci sera neutralisé. Il sera cependant impossible d'actionner le groupe radio.

Si cette option n'est pas cochée, la groupe radio sera invisible quand il sera neutralisé.

#### **Bouton « Caractères » dans la fenêtre « Neutralisation »**

Permet de définir la police de caractères utilisée pour afficher le groupe radio quand celui-ci est neutralisé et que l'option « Visible si neutralisé » est cochée.

#### **Bouton « Hint »**

Ce bouton permet de définir les paramètres de « Hint » de l'objet. Le texte « Hint » que l'on définit est le texte qui apparaît sur fond jaune lorsque l'on passe le curseur de la souris sur l'objet.

Quand on clique sur le bouton « Hint », la fenêtre suivante s'ouvre :



Cette fenêtre présente deux contrôles :

#### **Option « Afficher le Hint »**

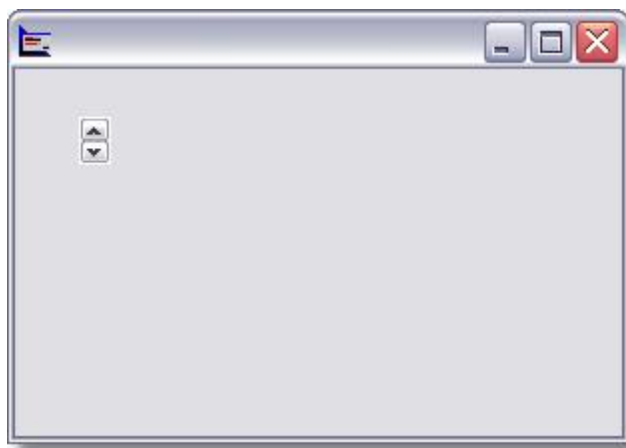
Si cette option n'est pas cochée, le texte du Hint n'apparaîtra pas quand le curseur de la souris passera sur l'objet même si celui-ci est défini. Il faut donc cocher cette option si l'on veut utiliser le « Hint » de l'objet.

#### **Fenêtre de saisie « Hint »**

Cette fenêtre permet d'entrer le texte du Hint. Ce texte peut être un paramètre (exemple {1}).

### **3.2.6.1.4 Flèches Haut/Bas**

Cet objet permet de faire varier une valeur numérique entière à l'aide de flèches haut bas :



### **Boîte de dialogue "Flèches Haut/Bas" :**

Les champs de saisie de cette boîte sont les suivants :

#### **Expression**

Indiquer dans ce champ le nom de la variable à contrôler. Cette variable doit être de type numérique entier (Registre).

*Exemple* : R12

Il est possible de spécifier l'adresse IP distante du noyau APIGRAF PLC dans l'expression. Si c'est le cas, c'est cette adresse qui sera prise en compte pour le rafraîchissement.

*Exemple* : !193.0.1.1 !R12

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

#### **Valeur Min**

Cette option définit la valeur minimale. C'est la valeur en dessous de laquelle on ne pourra pas descendre en actionnant la flèche du bas.

Il est possible de spécifier dans ce champ une valeur constante ou l'expression d'une variable automate :

*Exemple* : R20

Pour les variables automates, on peut de plus spécifier l'adresse IP distante du noyau APIGRAF PLC dans l'expression. Si c'est le cas, c'est cette adresse qui sera prise en compte pour le rafraîchissement de la valeur Min.

*Exemple : !193.0.1.1 !R12*

Dans ce cas, la borne min du curseur sera variable.

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

### **Valeur Max**

Cette option définit la valeur maximale. C'est la valeur au dessus de laquelle on ne pourra pas descendre en actionnant la flèche du haut.

Ce champ offre les mêmes options d'utilisation que le champ « Valeur Min » ci dessus.

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

### **Valeur Incrément**

Spécifie la valeur d'incrément/décrément à chaque click sur les flèches. La valeur par défaut est 1.

Cette variable doit être de type numérique entier (Registre).

*Exemple : R12*

Il est possible de spécifier l'adresse IP distante du noyau APIGRAF PLC dans l'expression. Si c'est le cas, c'est cette adresse qui sera prise en compte pour le rafraîchissement.

*Exemple : !193.0.1.1 !R12*

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

### **Orientation**

Cette option permet de définir si les flèches doivent être orientées horizontalement ou verticalement.

### **Expression de neutralisation**

Permet de spécifier la variable ou l'expression de neutralisation (Celle-ci doit obligatoirement être de type booléen). Les flèches seront neutralisées chaque fois que la variable ou l'expression sera égale à true.

*Exemple : SA10*

Il est possible de spécifier l'adresse IP distante du noyau APIGRAF PLC dans l'expression. Si c'est le cas, c'est cette adresse qui sera prise en compte pour le rafraîchissement.

*Exemple : !193.0.1.1 !SA10*

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

### **Option « Visible si neutralisé »**

Si cette option est cochée, les flèches seront visibles quand l'objet sera neutralisé. Cependant, l'opérateur ne pourra pas agir dessus.

Si cette option n'est pas cochée, l'objet sera invisible quand il sera neutralisé.

### **Bouton « Hint »**

Ce bouton permet de définir les paramètres de « Hint » de l'objet. Le texte « Hint » que l'on définit est le texte qui apparaît sur fond jaune lorsque l'on passe le curseur de la souris sur l'objet.

Quand on clique sur le bouton « Hint », la fenêtre suivante s'ouvre :



Cette fenêtre présente deux contrôles :

#### Option « Afficher le Hint »

Si cette option n'est pas cochée, le texte du Hint n'apparaîtra pas quand le curseur de la souris passera sur l'objet même si celui-ci est défini. Il faut donc cocher cette option si l'on veut utiliser le « Hint » de l'objet.

#### Fenêtre de saisie « Hint »

Cette fenêtre permet d'entrer le texte du Hint. Ce texte peut être un paramètre (exemple {1}).

### 3.2.6.2 Objets de l'onglet BOUTONS

#### 3.2.6.2.1 Bouton simple

Cet objet appelé aussi « bouton Windows » est utilisé pour effectuer des macros locales comme :

- Ouvrir une autre page de supervision
- Fermer la page, etc.

Il permet d'effectuer des macros sur des noyaux APIGRAF PLC.

Exemple : SETSA1:=1; ( Voir le chapitre dédié au macros pour plus de détails ).

**Attention** : Ce bouton ne peut pas être utilisé pour contrôler des variables automates, il faut dans ce cas utiliser le bouton API.

#### Boîte de dialogue bouton simple



Les options sont les suivantes :

### Texte

Cette option détermine le texte qui sera visible sur le bouton simple.

Un raccourci de type Windows® (ALT + une lettre) peut être affecté aux boutons simples. Il suffit d'ajouter dans le texte du bouton le symbole « & » avant la lettre qui servira de raccourci.

*Exemple* : si le texte saisi est : M&odules : le raccourci du bouton sera ALT O

De plus le texte du bouton aura l'apparence suivante : Modules.

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

### Caractères

Cette option permet de choisir la police de caractère à utiliser pour le texte du bouton. Elle provoque l'affichage de la boîte de dialogue « police de caractères »

### Macro enfacement

Ce contrôle permet de saisir rapidement les macros à exécuter lors de l'enfoncement du bouton.

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

### Expression de neutralisation

Permet de spécifier la variable ou l'expression de neutralisation (Celle-ci doit obligatoirement être de type booléen). Le bouton sera alors neutralisé chaque fois que la variable ou l'expression sera égale à true.

*Exemple* : SA10

Il est possible de spécifier l'adresse IP distante du noyau APIGRAF PLC dans l'expression. Si c'est le cas, c'est cette adresse qui sera prise en compte pour le rafraîchissement.

Exemple : !193.0.1.1 !SA10

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

#### Option « Visible si neutralisé »

Si cette option est active, le bouton sera visible quand il sera neutralisé mais celui-ci ne pourra être actionné. De plus le texte du bouton apparaîtra en grisé.

Si cette option n'est pas active, le bouton sera invisible dès que l'expression de neutralisation sera à VRAI (Si bien sûr, l'expression de neutralisation est définie).

#### Option « Texte du bouton neutralisé »

Cette option doit être cochée si l'on veut avoir un texte différent sur le bouton lorsque celui-ci est neutralisé.

Dès que l'on coche cette option, une fenêtre de saisie apparaît en dessous de cette option, cette fenêtre permet de saisir le texte du bouton neutralisé.

#### Macro relâchement

Ce contrôle permet de saisir rapidement les macros à exécuter lors du relâchement du bouton.

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

#### Bouton « Hint »

Ce bouton permet de définir les paramètres de « Hint » de l'objet. Le texte « Hint » que l'on définit est le texte qui apparaît sur fond jaune lorsque l'on passe le curseur de la souris sur l'objet.

Quand on clique sur le bouton « Hint », la fenêtre suivante s'ouvre :



Cette fenêtre présente deux contrôles :

#### Option « Afficher le Hint »

Si cette option n'est pas cochée, le texte du Hint n'apparaîtra pas quand le curseur de la souris passera sur l'objet même si celui-ci est défini. Il faut donc cocher cette option si l'on veut utiliser le « Hint » de l'objet.

#### Fenêtre de saisie « Hint »

Cette fenêtre permet d'entrer le texte du Hint. Ce texte peut être un paramètre (exemple {1}).

#### Expression (Optionnel)

Cette option doit contenir le nom d'une variable automate de type booléen, c'est à dire à deux états 0 ou 1.

Exemple : SA10

Il est possible de spécifier l'adresse IP distante du noyau OPTIMA PLC ou APIGRAF dans l'expression. Si c'est le cas, c'est cette adresse qui sera prise en compte pour le rafraîchissement.

Exemple : !193.0.1.1 !SA10

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

Si une expression est spécifiée dans ce champ. Alors la variable automate sera mise à 1 dès que le bouton sera enfoncé à l'aide de la souris et sera remis à 0 dès que le bouton sera relâché.

### 3.2.6.2.2 Bouton API

Cet objet permet d'obtenir très simplement un fonctionnement de type bouton poussoir.

Une variable automate de type booléen est alors associée au bouton.

#### Boîte de dialogue paramètre bouton API

**Objet bouton Apigraf**

Général | Forme | Neutralisation

Expression

Type de bouton

- ☒ monostable
- ☐ bistable
- ☐ neutralisé

Raccourci clavier

Hint

Bouton relâché

Texte

Caractères

Image (Aucune)

☐ Transparence Automatique de l'image

☐ Ajustement de l'image à la taille du bouton

Couleur

Supprimer Image

Bouton enfoncé

☐ Texte

Image (Aucune)

☐ Transparence Automatique de l'image

☐ Ajustement de l'image à la taille du bouton

Couleur

Supprimer Image

Aide Annuler OK

Cette boîte dialogue comporte trois onglets : Général, Forme, Neutralisation.

**Boîte de dialogue paramètre bouton API : Général**

Les options sont les suivantes :

**Expression**

Cette option doit contenir le nom d'une variable automate de type booléen, c'est à dire à deux états.

En mode « Exécution » ou « rafraîchissement données en édition », les boutons apparaîtront neutralisés si l'expression de la variable ne peut être évaluée par l'automate.

*Exemple : SA10*

Il est possible de spécifier l'adresse IP distante du noyau APIGRAF PLC dans l'expression. Si c'est le cas, c'est cette adresse qui sera prise en compte pour le rafraîchissement.

*Exemple : !193.0.1.1 !SA10*

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

**Type de bouton**

Cette option permet d'indiquer si le bouton est monostable, bistable ou neutralisé. S'il est monostable, il revient lorsque l'on cesse de le presser.

S'il est bistable, un clic positionnera le bouton enfoncé, il faut alors cliquer une seconde fois sur le bouton pour le relâcher.

Si le bouton est neutralisé, l'opérateur ne pourra pas activer ou désactiver le bouton. Cependant le bouton apparaîtra relâché ou enfoncé suivant la valeur de l'expression qui lui est associée. Dans ce cas, l'apparence du bouton (couleur etc..) est définie dans l'onglet neutralisation.

**Zone « Bouton relâché »**

Cette zone permet de définir l'aspect général du bouton lorsque celui-ci est relâché. Cette zone présente les contrôles suivants :

- **Couleur**

Permet de définir la couleur de fond du bouton quand celui-ci est relâché.

- **Texte**

Permet de définir le texte sur le bouton quand celui-ci est relâché.

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

- **Bouton « Caractères »**

Ouvre la fenêtre de sélection de la police de caractère qui sera utilisée pour le texte du bouton quand celui-ci est relâché.

- **Bouton « Image »**

Permet de définir une image de fond (Bitmap) qui apparaît sur le bouton quand celui-ci est relâché.

- **Option « Transparence Automatique de l'image »**

Cocher cette option affiche l'image du bouton en transparence. La couleur de transparence est définie automatiquement.

- **Option « Ajustement de l'image à la taille du bouton »**

Cette option permet d'ajuster la taille de l'image à la taille du bouton.

- **Bouton « Supprimer Image »**

Ce contrôle permet d'effacer l'image actuelle.



## Zone « Bouton enfoncé »

Cette zone permet de définir l'aspect général du bouton lorsque celui-ci est pressé. Cette zone présente les contrôles suivants :

- **Couleur**

Permet de définir la couleur de fond du bouton quand celui-ci est enfoncé.

- **Option « Texte »**

Cette option doit être cochée si l'on veut que le texte du bouton pressé soit différent du texte du bouton relâché. Cocher cette option permet donc de saisir le texte du bouton pressé ainsi que la police de caractères utilisée.

- **Texte**

Ce contrôle n'est visible que si l'option « Texte » est cochée.

Permet de définir le texte sur le bouton quand celui-ci est enfoncé.

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

- **Bouton « Caractères »**

Ce contrôle n'est visible que si l'option « Texte » est cochée.

Ouvre la fenêtre de sélection de la police de caractère qui sera utilisée pour le texte du bouton quand celui-ci est enfoncé.

- **Bouton « Image »**

Permet de définir une image de fond (Bitmap) qui apparaît sur le bouton quand celui-ci est enfoncé.

- **Option « Transparence Automatique de l'image »**

Cocher cette option affiche l'image du bouton en transparence. La couleur de transparence est définie automatiquement.

- **Option « Ajustement de l'image à la taille du bouton »**

Cette option permet d'ajuster la taille de l'image à la taille du bouton.

## Raccourci clavier

Cette option permet de spécifier une touche dont l'utilisation permet d'activer directement le bouton APIGRAF. Les valeurs acceptées dans ce champ sont :

Lettres de A à Z, les majuscules et les minuscules n'étant pas différenciées.

Les chiffres de 0 à 9

*Attention à l'utilisation conjointe de fenêtres de saisie et de touche de raccourci de type alphanumérique sur des boutons ! !. En effet si lors d'une saisie, on tape le chiffre 9 et que dans la même page se trouve un bouton avec « 9 » pour raccourci, le bouton sera enclenché.*

Les touches de fonctions de F2 à F12 (La touche F1 étant réservée pour l'aide Windows®)  
INSERT, SUPPR, ESCAPE, BACK

Les combinaisons de touches avec ALT : ALT A, ALT B, ALT C, ..... , ALT Z, ALT 0,... ALT 9

## Bouton « Hint »

Ce bouton permet de définir les paramètres de « Hint » de l'objet.

Le texte « Hint » que l'on définit est le texte qui apparaît sur fond jaune lorsque l'on passe le curseur de la souris sur l'objet. Quand on clique sur le bouton « Hint », la fenêtre suivante s'ouvre :



Cette fenêtre présente deux contrôles :

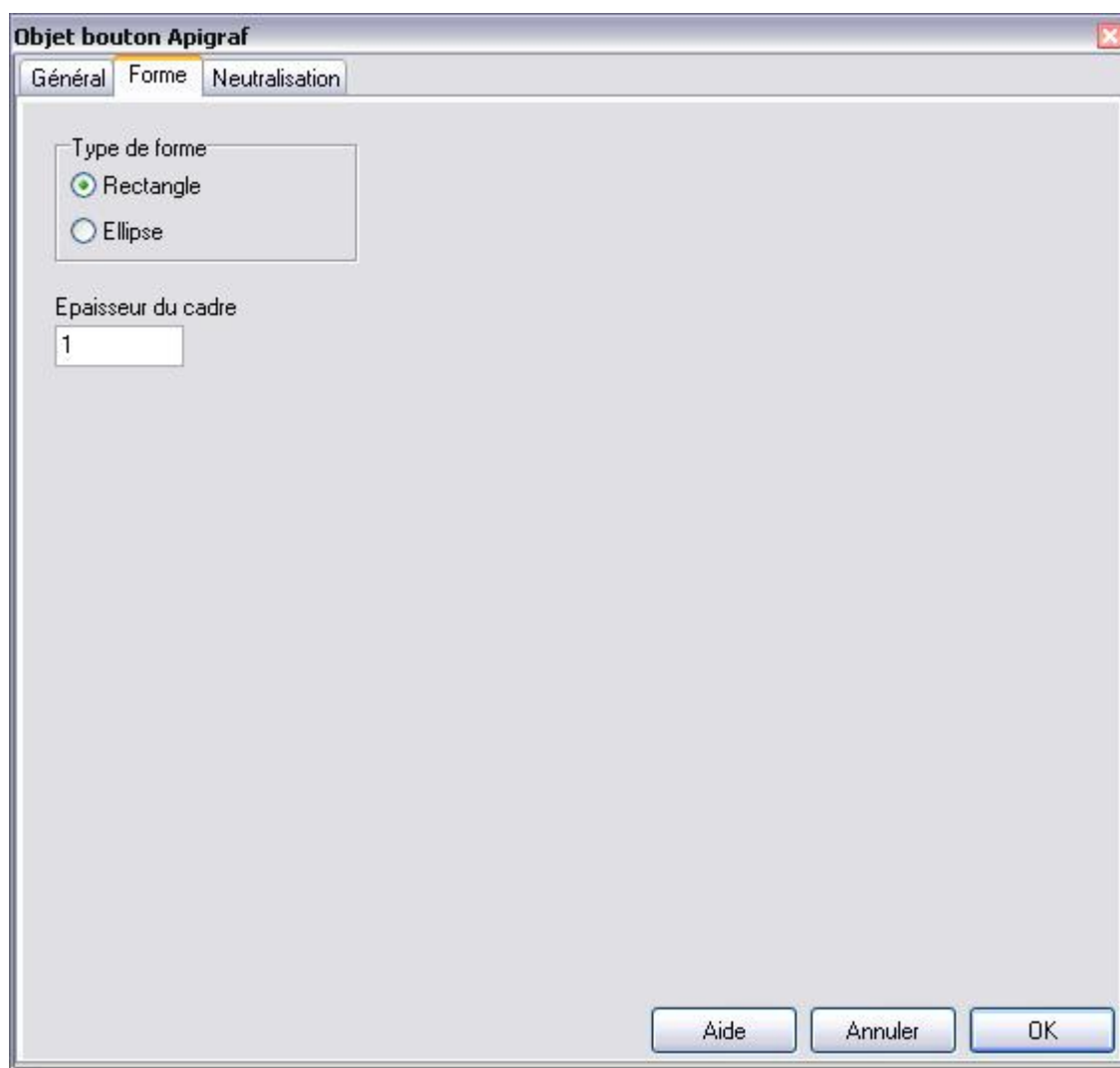
**Option « Afficher le Hint »**

Si cette option n'est pas cochée, le texte du Hint n'apparaîtra pas quand le curseur de la souris passera sur l'objet même si celui-ci est défini. Il faut donc cocher cette option si l'on veut utiliser le « Hint » de l'objet.

**Fenêtre de saisie « Hint »**

Cette fenêtre permet d'entrer le texte du Hint. Ce texte peut être un paramètre (exemple {1}).

**Boîte de dialogue paramètre bouton API : Forme**



Les options sont les suivantes :

- **Type de Forme**

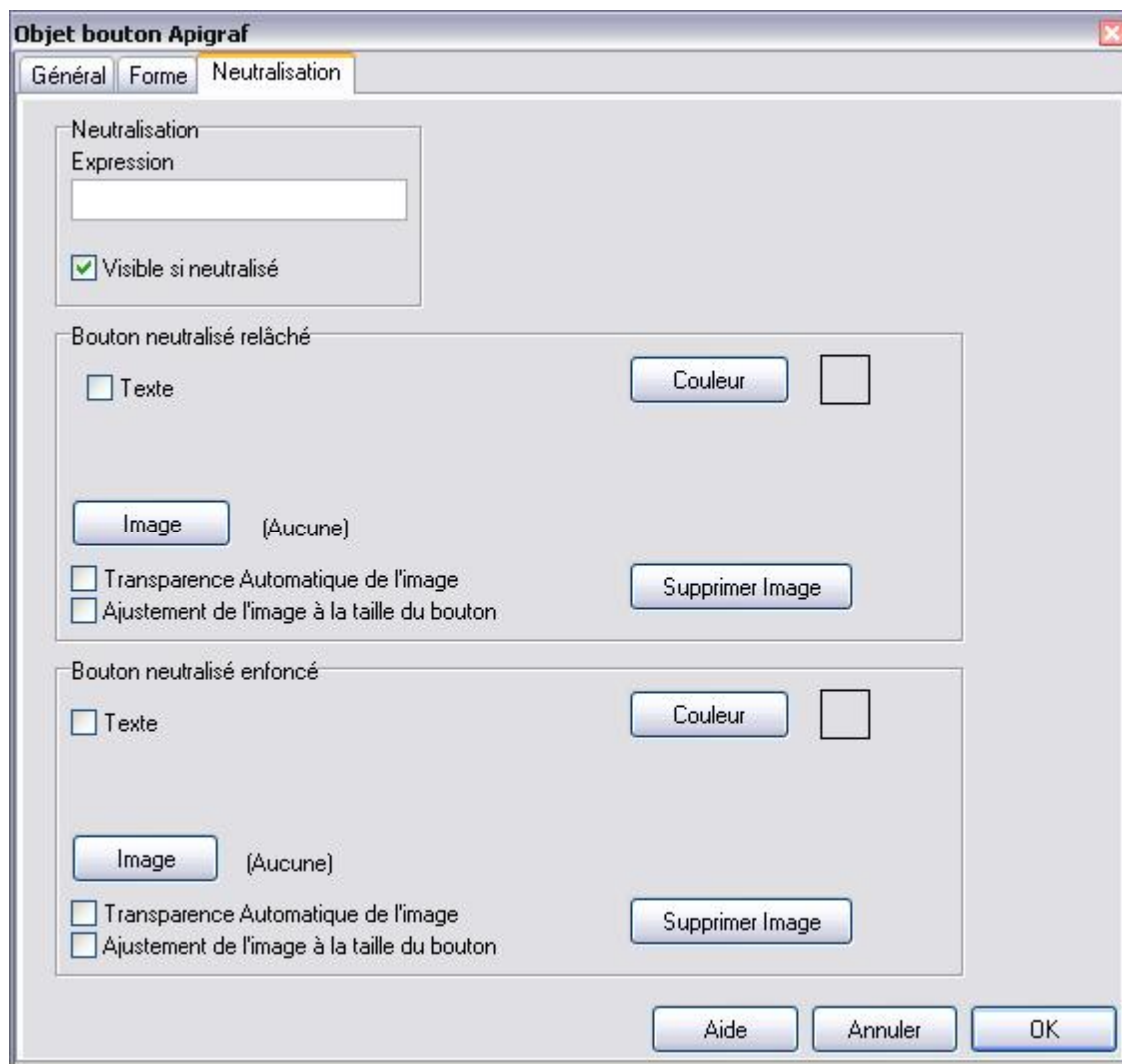
Permet de choisir la forme du bouton Rectangle ou Ellipse.

- **Epaisseur du cadre**

Cette option permet de choisir l'épaisseur du cadre du bouton. La valeur par défaut est 1 pixel.

### **Boîte de dialogue paramètre bouton API : Neutralisation**

Cette boîte permet de paramétrer les options de neutralisation du bouton API. Dans le cas où le bouton est neutralisé, celui-ci n'est plus accessible, et peut même être rendu invisible ( voir plus loin).



Les options sont les suivantes :

- **Expression de neutralisation**

Permet de spécifier la variable ou l'expression de neutralisation (Celle-ci doit obligatoirement être de type booléen). Le bouton sera alors neutralisé chaque fois que la variable ou l'expression sera égale à true.

*Exemple : L9*

Il est possible de spécifier l'adresse IP distante du noyau APIGRAF PLC dans l'expression. Si c'est le cas, c'est cette adresse qui sera prise en compte pour le rafraîchissement.

*Exemple : !193.0.1.1 !L9*

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

- **Option visible si neutralisé**

Si cette option n'est pas cochée, le bouton sera invisible lorsqu'il sera neutralisé.

- **Zone « Bouton neutralisé relâché »**

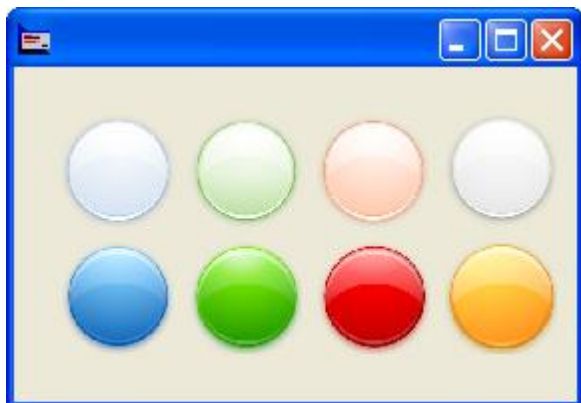
Cette zone permet de définir la couleur, l'image et le texte du bouton quand celui-ci sera neutralisé et non activé. (La valeur de l'expression du bouton est alors égale à 0)

- **Zone « Bouton neutralisé pressé »**

Cette zone permet de définir la couleur, l'image et le texte du bouton quand celui-ci sera neutralisé et activé.

### 3.2.6.2.3 Bouton poussoir type 1

Cet objet représente un bouton poussoir qui permet d'activer ou de désactiver une valeur de type booléen. L'apparence du bouton provient de fichiers images :



#### Boîte de dialogue de l'objet "Bouton Poussoir type 1" - Onglet Général :

Les champs de saisie de cette boîte sont les suivants :

#### **Expression**

Indiquer dans ce champ le nom de la variable à contrôler. Cette variable doit être de type booléen

*Exemple : SA10*

Il est possible de spécifier l'adresse IP distante du noyau OPTIMA PLC ou APIGRAF PLC dans l'expression. Si c'est le cas, c'est cette adresse qui sera prise en compte pour le rafraîchissement.

*Exemple : !193.0.1.1 !SA10*

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

Si la variable provient d'un serveur OPC, le bouton "OPC" permet d'explorer le serveur OPC et de trouver la variable à superviser dans un explorateur.

### Style

Permet de définir l'apparence du voyant : Bleu, vert, rouge ou jaune.

### Type de bouton

Cette option permet d'indiquer si le bouton est monostable ou bistable. S'il est monostable, il revient lorsque l'on cesse de le presser.

S'il est bistable, un clic positionnera le bouton enfoncé, il faut alors cliquer une seconde fois sur le bouton pour le relâcher.

### Option "Bouton à accroche"

Si cette option est activée, appuyer sur le bouton activera la variable booléenne de l'expression associée. On ne pourra alors pas désactiver la variable en appuyant à nouveau sur le bouton.

### Bouton « Hint »

Ce bouton permet de définir les paramètres de « Hint » de l'objet. Le texte « Hint » que l'on définit est le texte qui apparaît sur fond jaune lorsque l'on passe le curseur de la souris sur l'objet.

Quand on clique sur le bouton « Hint », la fenêtre suivante s'ouvre :



Cette fenêtre présente deux contrôles :

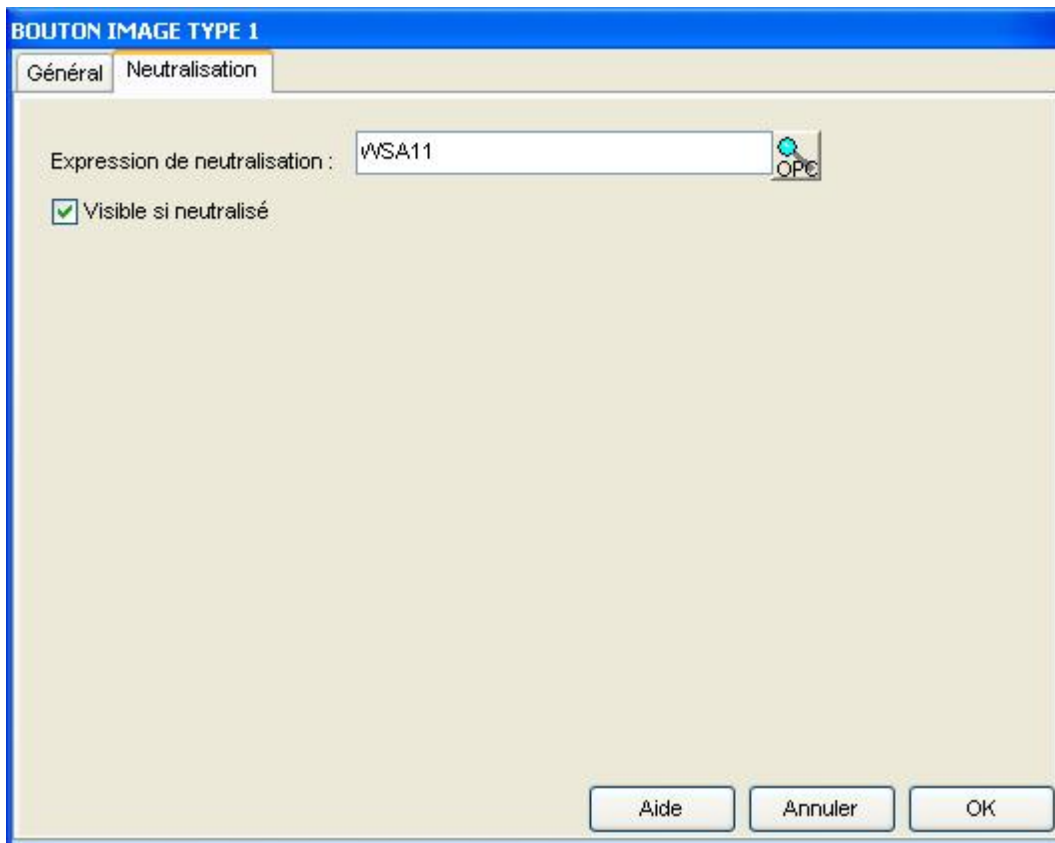
### Option « Afficher le Hint »

Si cette option n'est pas cochée, le texte du Hint n'apparaîtra pas quand le curseur de la souris passera sur l'objet même si celui-ci est défini. Il faut donc cocher cette option si l'on veut utiliser le « Hint » de l'objet.

### Fenêtre de saisie « Hint »

Cette fenêtre permet d'entrer le texte du Hint. Ce texte peut être un paramètre (exemple {1}).

### Boîte de dialogue de l'objet "Bouton Poussoir type 1" - Onglet Neutralisation :



### Expression de neutralisation

Permet de spécifier la variable ou l'expression de neutralisation (Celle-ci doit obligatoirement être de type booléen). Le bouton poussoir sera neutralisé chaque fois que la variable ou l'expression sera égale à true.

*Exemple : SA10*

Il est possible de spécifier l'adresse IP distante du noyau OPTIMA PLC ou APIGRAF PLC dans l'expression. Si c'est le cas, c'est cette adresse qui sera prise en compte pour le rafraîchissement.

*Exemple : !193.0.1.1 !SA10*

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

### Option « Visible si neutralisé »

Si cette option est cochée, le bouton poussoir sera visible quand l'objet sera neutralisé. Cependant, l'opérateur ne pourra pas agir dessus.

Si cette option n'est pas cochée, l'objet sera invisible quand il sera neutralisé.

### 3.2.6.2.4 Bouton à LED

Cet objet représente un bouton standard avec une LED incorporée qui indique l'état de la variable booléenne associée au bouton.

Le bouton peut être monostable, bistable ou neutralisé par défaut.

Une variable de neutralisation peut également être définie.



**Boîte de dialogue de l'objet "Bouton à LED"- Onglet "général" :**

### Expression

Cette option doit contenir le nom d'une variable automate de type booléen, c'est à dire à deux états.

En mode « Exécution » ou « rafraîchissement données en édition », les boutons apparaîtront neutralisés si l'expression de la variable ne peut être évaluée par l'automate.

*Exemple* : SA10

Il est possible de spécifier l'adresse IP distante du noyau APIGRAF PLC dans l'expression. Si c'est le cas, c'est cette adresse qui sera prise en compte pour le rafraîchissement.

*Exemple* : !193.0.1.1 !SA10

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

### Type de bouton

Cette option permet d'indiquer si le bouton est monostable, bistable ou neutralisé. S'il est monostable, il revient lorsque l'on cesse de le presser.

S'il est bistable, un clic positionnera le bouton enfoncé, il faut alors cliquer une seconde fois sur le bouton pour le relâcher.

Si le bouton est neutralisé, l'opérateur ne pourra pas activer ou désactiver le bouton.

### Type de bouton

Définit l'aspect de la LED (Bleue, rouge, triangulaire,...)

### Bouton « Caractères »

Ouvre la fenêtre de sélection de la police de caractère qui sera utilisée pour le texte du bouton quand celui-ci est relâché.

### Raccourci clavier

Cette option permet de spécifier une touche dont l'utilisation permet d'activer directement le bouton APIGRAF. Les valeurs acceptées dans ce champ sont :

Lettres de A à Z, les majuscules et les minuscules n'étant pas différenciées.  
Les chiffres de 0 à 9

*Attention à l'utilisation conjointe de fenêtres de saisie et de touche de raccourci de type alphanumérique sur des boutons !! En effet si lors d'une saisie, on tape le chiffre 9 et que dans la même page se trouve un bouton avec « 9 » pour raccourci, le bouton sera enclenché.*

Les touches de fonctions de F2 à F12 (La touche F1 étant réservée pour l'aide Windows®)  
INSERT, SUPPR, ESCAPE, BACK

Les combinaisons de touches avec ALT : ALT A, ALT B, ALT C, ..... , ALT Z, ALT 0,... ALT 9

### Bouton « Hint »

Ce bouton permet de définir les paramètres de « Hint » de l'objet.

Le texte « Hint » que l'on définit est le texte qui apparaît sur fond jaune lorsque l'on passe le curseur de la souris sur l'objet. Quand on clique sur le bouton « Hint », la fenêtre suivante s'ouvre :



Cette fenêtre présente deux contrôles :

#### Option « Afficher le Hint »

Si cette option n'est pas cochée, le texte du Hint n'apparaîtra pas quand le curseur de la souris passera sur l'objet même si celui-ci est défini. Il faut donc cocher cette option si l'on veut utiliser le « Hint » de l'objet.

#### Fenêtre de saisie « Hint »

Cette fenêtre permet d'entrer le texte du Hint. Ce texte peut être un paramètre (exemple {1}).

### Boîte de dialogue de l'objet "Bouton à LED"- Onglet "Neutralisation" :



**Bouton à LED**

Général Neutralisation

Expression de neutralisation :

☐ Visible si neutralisé

☐ Texte du bouton neutralisé

Caractères

Aide Annuler OK

### Expression de neutralisation

Permet de spécifier la variable ou l'expression de neutralisation (Celle-ci doit obligatoirement être de type booléen). Le bouton sera alors neutralisé chaque fois que la variable ou l'expression sera égale à true.

*Exemple : SA10*

Il est possible de spécifier l'adresse IP distante du noyau OPTIMA PLC ou APIGRAF PLC dans l'expression. Si c'est le cas, c'est cette adresse qui sera prise en compte pour le rafraîchissement.

*Exemple : !193.0.1.1 !SA10*

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

### Option « Visible si neutralisé »

Si cette option est active, le bouton sera visible quand il sera neutralisé mais celui-ci ne pourra être actionné. De plus le texte du bouton apparaîtra en grisé.

Si cette option n'est pas active, le bouton sera invisible dès que l'expression de neutralisation sera à VRAI (Si bien sûr, l'expression de neutralisation est définie).

### Option « Texte du bouton neutralisé »

Cette option doit être cochée si l'on veut avoir un texte différent sur le bouton lorsque celui-ci est neutralisé.

Dès que l'on coche cette option, une fenêtre de saisie apparaît en dessous de cette option, cette fenêtre permet de saisir le texte du bouton neutralisé.

### 3.2.6.2.5 Bouton de Fonction

Cet objet est utilisé pour effectuer des macros locales comme :

- Ouvrir une autre page de supervision ( MACRO SUPERG(NomPage))
- Fermer la page, etc.

Il permet d'effectuer des macros sur des noyaux OPTIMA PLC ou APIGRAF.

Exemple avec Optima PLC : SETSA1:=1; ( Voir le chapitre dédié au macros pour plus de détails ).

Ce bouton peut intégrer une image (icône,bitmap,PNG,...).



### Boîte de dialogue bouton de fonction - Onglet Général :

#### **Texte**

Cette option détermine le texte qui sera visible sur le bouton simple.

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

#### **Caractères**

Cette option permet de choisir la police de caractère à utiliser pour le texte du bouton. Elle provoque l'affichage de la boîte de dialogue « police de caractères »

#### **Macro enfoncement**

Ce contrôle permet de saisir rapidement les macros à exécuter lors de l'enfoncement du bouton.

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

### Macro relachement

Ce contrôle permet de saisir rapidement les macros à exécuter lors du relachement du bouton.  
Ce sont ces macros qui seront exécutées si le bouton est appelé à l'aide de son raccourci clavier.

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

### Raccourci clavier

Cette option permet de spécifier une touche dont l'utilisation permet d'activer directement le bouton APIGRAF. Les valeurs acceptées dans ce champ sont :

Lettres de A à Z, les majuscules et les minuscules n'étant pas différenciées.  
Les chiffres de 0 à 9

*Attention à l'utilisation conjointe de fenêtres de saisie et de touche de raccourci de type alphanumérique sur des boutons ! !. En effet si lors d'une saisie, on tape le chiffre 9 et que dans la même page se trouve un bouton avec « 9 » pour raccourci, le bouton sera enclenché.*

Les touches de fonctions de F2 à F12 (La touche F1 étant réservée pour l'aide Windows®)  
INSERT, SUPPR, ESCAPE, BACK

Les combinaisons de touches avec ALT : ALT A, ALT B, ALT C, ..... , ALT Z, ALT 0,... ALT 9

### Bouton « Hint »

Ce bouton permet de définir les paramètres de « Hint » de l'objet. Le texte « Hint » que l'on définit est le texte qui apparaît sur fond jaune lorsque l'on passe le curseur de la souris sur l'objet.

Quand on clique sur le bouton « Hint », la fenêtre suivante s'ouvre :



Cette fenêtre présente deux contrôles :

#### Option « Afficher le Hint »

Si cette option n'est pas cochée, le texte du Hint n'apparaîtra pas quand le curseur de la souris passera sur l'objet même si celui-ci est défini. Il faut donc cocher cette option si l'on veut utiliser le « Hint » de l'objet.

#### Fenêtre de saisie « Hint »

Cette fenêtre permet d'entrer le texte du Hint. Ce texte peut être un paramètre (exemple {1})

#### Expression (Optionnel)

Cette option doit contenir le nom d'une variable automate de type booléen, c'est à dire à deux états 0 ou 1.

Exemple : SA10

Il est possible de spécifier l'adresse IP distante du noyau OPTIMA PLC ou APIGRAF dans l'expression. Si c'est le cas, c'est cette adresse qui sera prise en compte pour le rafraîchissement.

Exemple : !193.0.1.1 !SA10

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

Si une expression est spécifiée dans ce champ. Alors la variable automate sera mise a 1 dès que le bouton sera enfoncé à l'aide de la souris et sera remis à 0 dès que le bouton sera relâché.

### **Boîte de dialogue bouton de fonction - Onglet Formes :**

#### **Bouton Arrondi**

Si cette option est cochée, les coins du boutons seront arrondis.

**BOUTON DE FONCTION**

Général **Forme** Neutralisation Icônes

☒ Bouton Arrondi

**Bouton au repos**

- Couleur départ fond gradient haut
- Couleur fin fond gradient haut
- Couleur départ fond gradient bas
- Couleur fin fond gradient bas

**Bouton sélectionné**

- Couleur départ fond gradient haut
- Couleur fin fond gradient haut
- Couleur départ fond gradient bas
- Couleur fin fond gradient bas

**Bouton enfoncé**

- Couleur départ fond gradient haut
- Couleur fin fond gradient haut
- Couleur départ fond gradient bas
- Couleur fin fond gradient bas

**Bouton enfoncé**

- Couleur départ fond gradient haut
- Couleur fin fond gradient haut
- Couleur départ fond gradient bas
- Couleur fin fond gradient bas

Aide Annuler OK

Il est également possible de paramétrer toutes les couleurs utilisées pour l'affichage du bouton dans ses différents états.

### **Boîte de dialogue bouton de fonction - Onglet Neutralisation :**

## Expression de neutralisation

Permet de spécifier la variable ou l'expression de neutralisation (Celle-ci doit obligatoirement être de type booléen). Le bouton sera alors neutralisé chaque fois que la variable ou l'expression sera égale à true.

Exemple : SA10

Il est possible de spécifier l'adresse IP distante du noyau APIGRAF PLC dans l'expression. Si c'est le cas, c'est cette adresse qui sera prise en compte pour le rafraîchissement.

Exemple : !193.0.1.1 !SA10

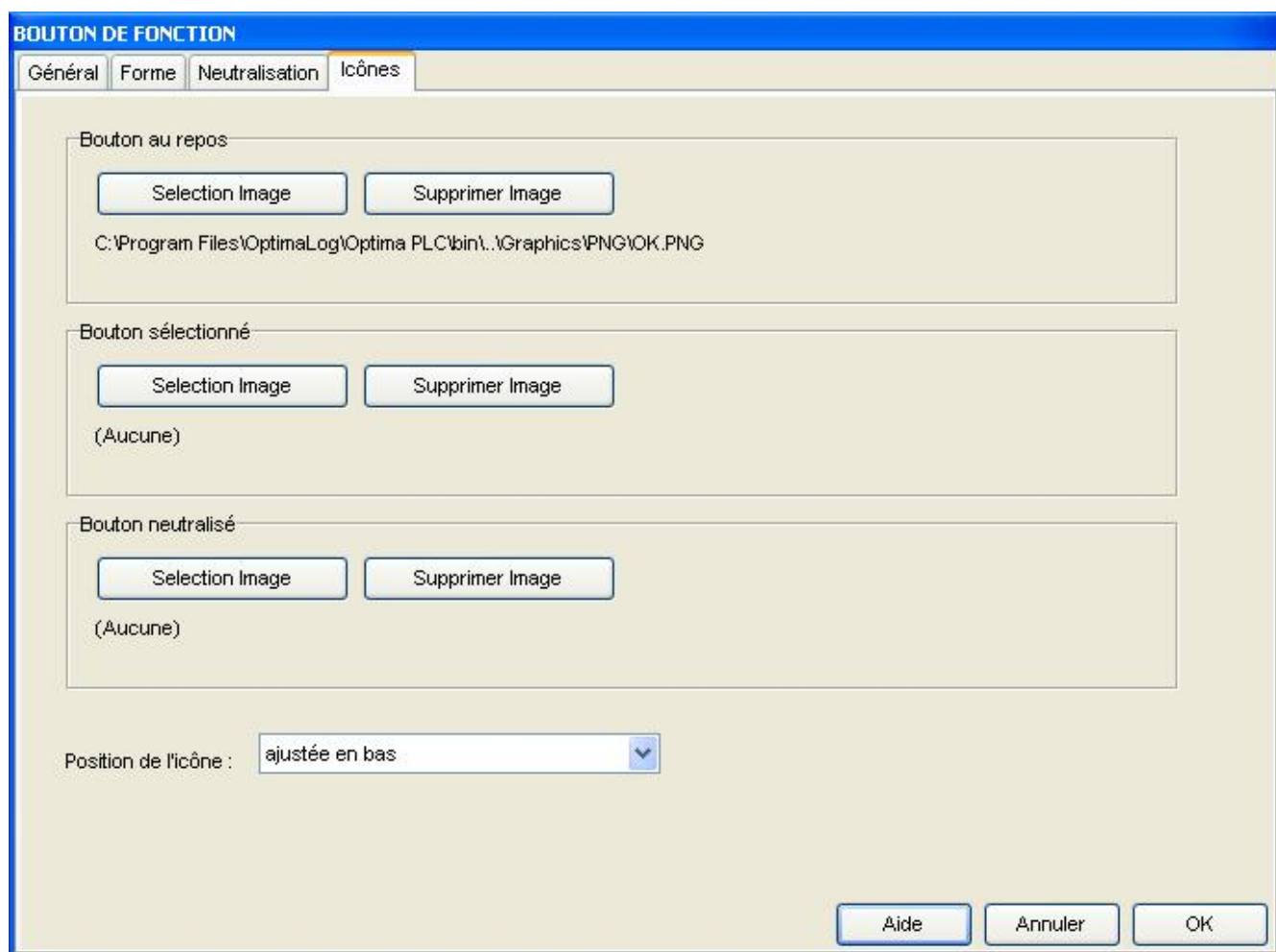
Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

## Option « Visible si neutralisé »

Si cette option est active, le bouton sera visible quand il sera neutralisé mais celui-ci ne pourra être actionné. De plus le texte du bouton apparaîtra en grisé.

Si cette option n'est pas active, le bouton sera invisible dès que l'expression de neutralisation sera à VRAI (Si bien sûr, l'expression de neutralisation est définie).

## Boîte de dialogue bouton de fonction - Onglet Icônes :



Sélectionner les images à intégrer dans le bouton. Tous les formats sont acceptés, le format PNG étant préconisé.

Lors de la sélection d'une première image, le répertoire ouvert par défaut est le répertoire de l'application PNG qui contient une librairie d'images au format PNG.

Il est également possible de choisir la position de l'image dans le bouton avec le menu déroulant "Position de l'icône".

### 3.2.6.2.6 Bouton de Fonction translucide

Cet objet est utilisé pour effectuer des macros locales comme :

- Ouvrir une autre page de supervision ( MACRO SUPERG(NomPage))
- Fermer la page, etc.

Il permet aussi d'effectuer des macros sur des noyaux OPTIMA PLC ou APIGRAF.

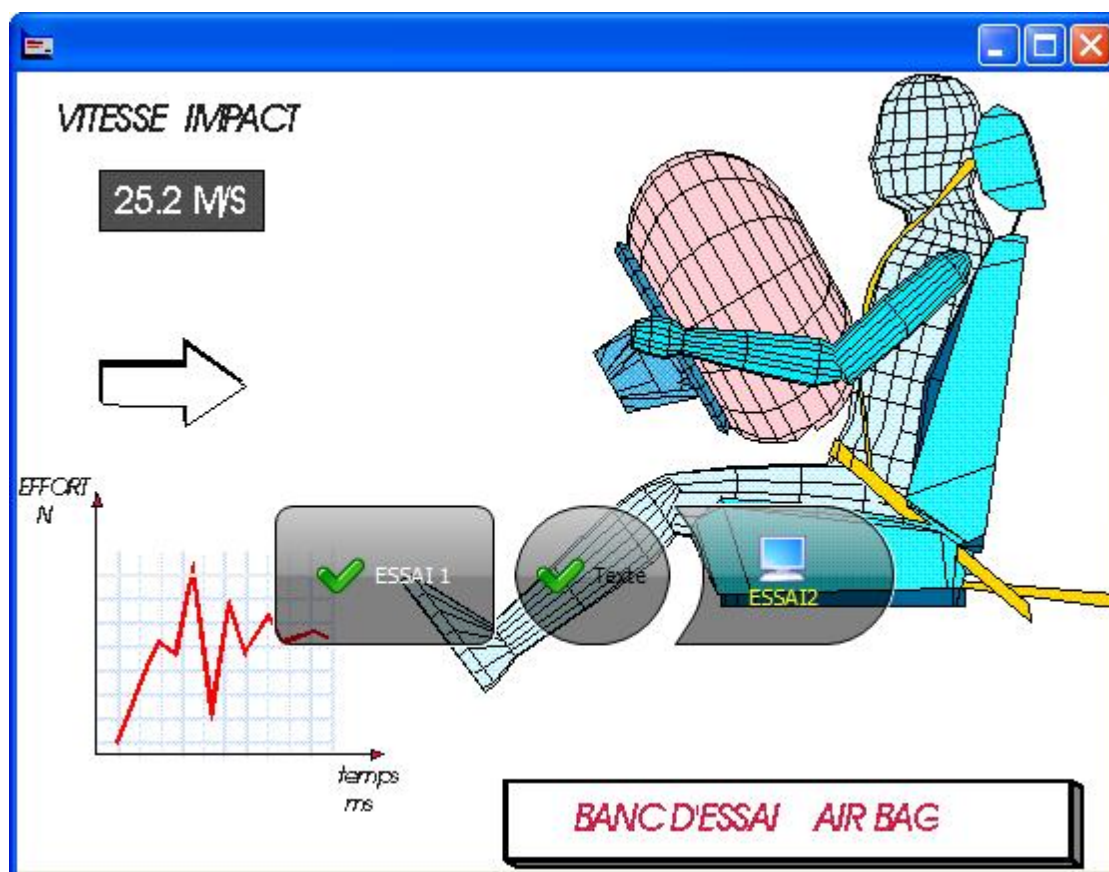
Exemple ouverture page "TEST" en local : SUPERG(TEST)

Exemple fermeture page courante : CLOSE

Exemple avec Optima PLC : SETSA1:=1; ( Voir le chapitre dédié au macros pour plus de détails ).

Ce bouton peut intégrer une image (icône,bitmap,PNG,...).

Son apparence est translucide :



Dans cet exemple, les boutons laissent apparaître l'image de fond de la fenêtre.

**Boîte de dialogue bouton de fonction - Onglet Général :**

**Bouton translucide**

Général Neutralisation Forme Icônes

Texte :  
{SSAI1}

Macro enfacement

Macro relâchement

Caractères

Raccourci clavier

Hint

Aide Annuler OK

### Texte

Cette option détermine le texte qui sera visible sur le bouton simple.

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

### Caractères

Cette option permet de choisir la police de caractère à utiliser pour le texte du bouton. Elle provoque l'affichage de la boîte de dialogue « police de caractères »

### Macro enfacement

Ce contrôle permet de saisir rapidement les macros à exécuter lors de l'enfacement du bouton.

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

### Macro relâchement

Ce contrôle permet de saisir rapidement les macros à exécuter lors du relâchement du bouton.  
Ce sont ces macros qui seront exécutées si le bouton est appelé à l'aide de son raccourci clavier.

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

### Raccourci clavier

Cette option permet de spécifier une touche dont l'utilisation permet d'activer directement le bouton APIGRAF. Les valeurs acceptées dans ce champ sont :

Lettres de A à Z, les majuscules et les minuscules n'étant pas différenciées.  
Les chiffres de 0 à 9

*Attention à l'utilisation conjointe de fenêtres de saisie et de touche de raccourci de type alphanumérique sur des boutons !! En effet si lors d'une saisie, on tape le chiffre 9 et que dans la même page se trouve un bouton avec « 9 » pour raccourci, le bouton sera enclenché.*

Les touches de fonctions de F2 à F12 (La touche F1 étant réservée pour l'aide Windows®)  
INSERT, SUPPR, ESCAPE, BACK

Les combinaisons de touches avec ALT : ALT A, ALT B, ALT C, ..... , ALT Z, ALT 0,... ALT 9

### Bouton « Hint »

Ce bouton permet de définir les paramètres de « Hint » de l'objet. Le texte « Hint » que l'on définit est le texte qui apparaît sur fond jaune lorsque l'on passe le curseur de la souris sur l'objet.

Quand on clique sur le bouton « Hint », la fenêtre suivante s'ouvre :



Cette fenêtre présente deux contrôles :

### Option « Afficher le Hint »

Si cette option n'est pas cochée, le texte du Hint n'apparaîtra pas quand le curseur de la souris passera sur l'objet même si celui-ci est défini. Il faut donc cocher cette option si l'on veut utiliser le « Hint » de l'objet.

### Fenêtre de saisie « Hint »

Cette fenêtre permet d'entrer le texte du Hint. Ce texte peut être un paramètre (exemple {1})

### Expression (Optionnel)

Cette option doit contenir le nom d'une variable automate de type booléen, c'est à dire à deux états 0 ou 1.

*Exemple : SA10*

Il est possible de spécifier l'adresse IP distante du noyau OPTIMA PLC ou APIGRAF dans l'expression. Si c'est le cas, c'est cette adresse qui sera prise en compte pour le rafraîchissement.

*Exemple : !193.0.1.1 !SA10*

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

Si une expression est spécifiée dans ce champ. Alors la variable automate sera mise à 1 dès que le bouton sera enfoncé à l'aide de la souris et sera remis à 0 dès que le bouton sera relâché.

### Boîte de dialogue bouton de fonction - Onglet Neutralisation :

#### Expression de neutralisation



Permet de spécifier la variable ou l'expression de neutralisation (Celle-ci doit obligatoirement être de type booléen). Le bouton sera alors neutralisé chaque fois que la variable ou l'expression sera égale à true.

*Exemple : SA10*

Il est possible de spécifier l'adresse IP distante du noyau APIGRAF PLC dans l'expression. Si c'est le cas, c'est cette adresse qui sera prise en compte pour le rafraîchissement.

*Exemple : !193.0.1.1 !SA10*

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

### Option « Visible si neutralisé »

Si cette option est active, le bouton sera visible quand il sera neutralisé mais celui-ci ne pourra être actionné. De plus le texte du bouton apparaîtra en grisé.

Si cette option n'est pas active, le bouton sera invisible dès que l'expression de neutralisation sera à VRAI (Si bien sûr, l'expression de neutralisation est définie).

### Boîte de dialogue bouton de fonction - Onglet Forme :

**Bouton translucide**

Général Neutralisation **Forme** Icônes

Forme du bouton : Rectangle

Sens bouton incurvé : à gauche

Rayon de courbure des coins du bouton rectangulaire : 8 Pixels

Aide Annuler OK

#### Forme du bouton

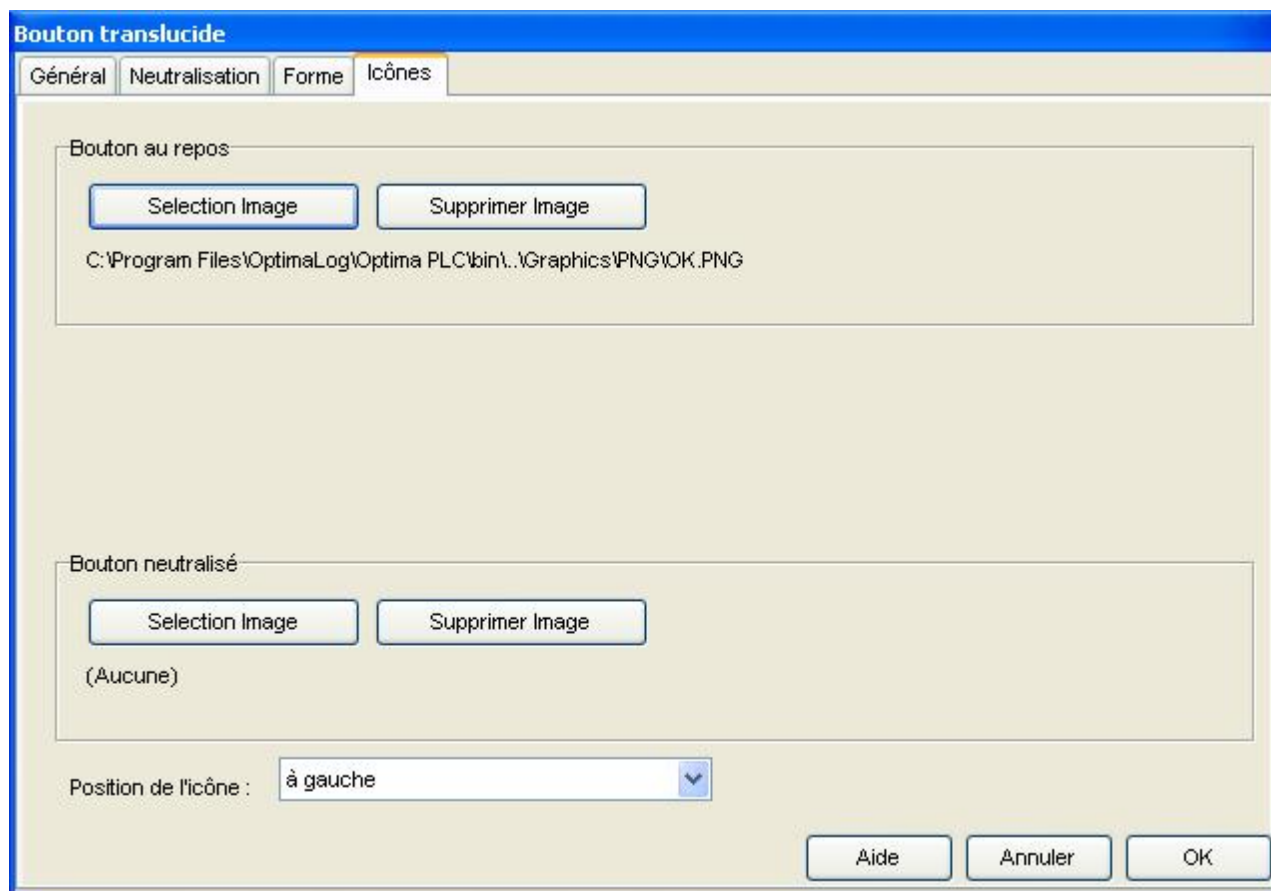
Permet de choisir en un bouton rectangle ou un bouton incurvé dans un sens

#### Sens Bouton incurvé

Permet de sélectionner le sens du bouton dans le cas où la forme incurvée a été sélectionnée?

#### Rayon de courbure des coins

Ce paramètre est actif si la forme du bouton est rectangle. Il spécifie la taille des arrondis des coins du bouton en pixels.

**Boîte de dialogue bouton de fonction - Onglet Icônes :**

Sélectionner les images à intégrer dans le bouton. Tous les formats sont acceptés, le format PNG étant préconisé. Lors de la sélection d'une première image, le répertoire ouvert par défaut est le répertoire de l'application PNG qui contient une librairie d'images au format PNG.

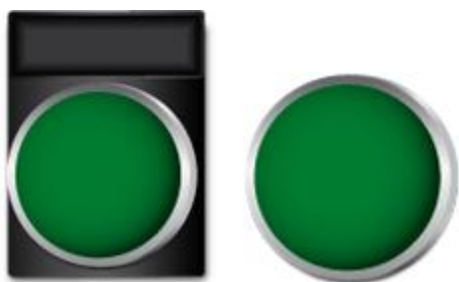
Il est également possible de choisir la position de l'image dans le bouton avec le menu déroulant "Position de l'icône".

**3.2.6.2.7 Bouton poussoir industriel**

Le bouton poussoir industriel représente un bouton type TELEMECANIQUE.

On peut définir une expression pour la valeur du bouton ( enfoncé ou relâché ) et également de manière facultative une expression pour la lampe associée au bouton. Dans ce cas, l'expression de lampe allumera le bouton quand sa valeur vaudra 1.

Le bouton industriel peut être avec ou sans étiquette ( attention : il faut choisir l'objet correspondant dans la barre d'outils !! )



Le bouton peut donc avoir 4 aspects différents :

- relâché éteint
- relâché allumé
- enfoncé éteint
- enfoncé allumé



#### **Boîte de dialogue de l'objet "Bouton Poussoir Industriel " - Onglet "Général" :**

Les champs de saisie de cette boîte sont les suivants :

#### **Expression**

Indiquer dans ce champ le nom de la variable à contrôler par le bouton. Cette variable doit être de type booléen

*Exemple : SA10*

Il est possible de spécifier l'adresse IP distante du noyau APIGRAF PLC dans l'expression. Si c'est le cas, c'est cette adresse qui sera prise en compte pour le rafraîchissement.

Exemple : !193.0.1.1 !SA10

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

Les champs de saisie de cette boîte sont les suivants :

### Expression lampe

Indiquer dans ce champ le nom de la variable qui va définir l'état de la LED dans le bouton.

### Couleur

5 couleurs sont disponibles : Vert, Blanc, Bleu, Jaune et Rouge.

### Type de bouton

Cette option permet d'indiquer si le bouton est monostable ou bistable. S'il est monostable, il revient lorsque l'on cesse de le presser.

S'il est bistable, un clic positionnera le bouton enfoncé, il faut alors cliquer une seconde fois sur le bouton pour le relâcher.

### Option "Bouton à accroche"

Si cette option est activée, appuyer sur le bouton activera la variable booléenne de l'expression associée. On ne pourra alors pas désactiver la variable en appuyant à nouveau sur le bouton.

### Bouton « Hint »

Ce bouton permet de définir les paramètres de « Hint » de l'objet. Le texte « Hint » que l'on définit est le texte qui apparaît sur fond jaune lorsque l'on passe le curseur de la souris sur l'objet.

Quand on clique sur le bouton « Hint », la fenêtre suivante s'ouvre :



Cette fenêtre présente deux contrôles :

### Option « Afficher le Hint »

Si cette option n'est pas cochée, le texte du Hint n'apparaîtra pas quand le curseur de la souris passera sur l'objet même si celui-ci est défini. Il faut donc cocher cette option si l'on veut utiliser le « Hint » de l'objet.

### Fenêtre de saisie « Hint »

Cette fenêtre permet d'entrer le texte du Hint. Ce texte peut être un paramètre (exemple {1}).

### Boîte de dialogue de l'objet "Bouton Poussoir Industriel " - Onglet "Neutralisation" :

#### Expression de neutralisation

Permet de spécifier la variable ou l'expression de neutralisation (Celle-ci doit obligatoirement être de type booléen). Le bouton poussoir sera neutralisé chaque fois que la variable ou l'expression sera égale à true.

Exemple : SA10

Il est possible de spécifier l'adresse IP distante du noyau OPTIMA PLC ou APIGRAF PLC dans l'expression. Si c'est le cas, c'est cette adresse qui sera prise en compte pour le rafraîchissement.

Exemple : !193.0.1.1 !SA10

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

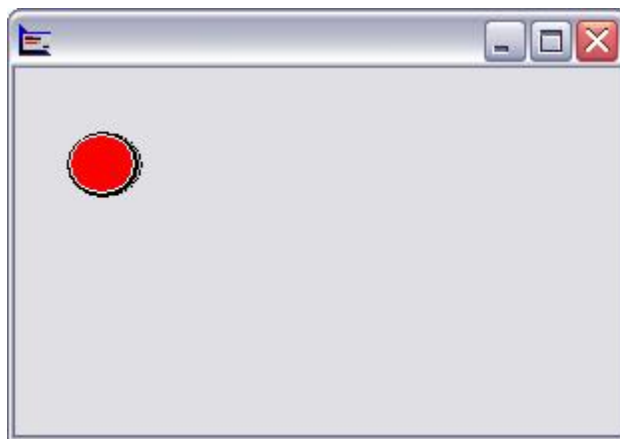
#### Option « Visible si neutralisé »

Si cette option est cochée, le bouton poussoir sera visible quand l'objet sera neutralisé. Cependant, l'opérateur ne pourra pas agir dessus.

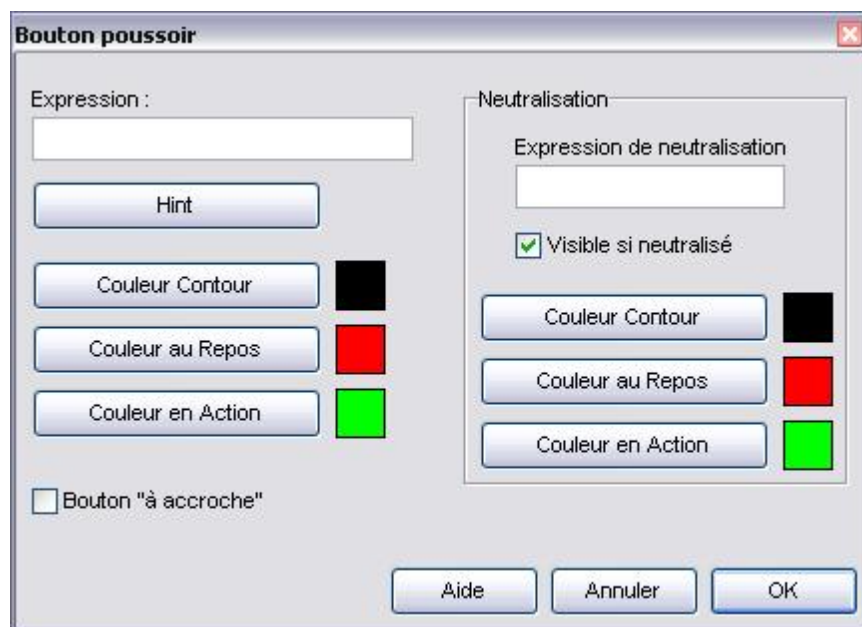
Si cette option n'est pas cochée, l'objet sera invisible quand il sera neutralisé.

### 3.2.6.2.8 Bouton poussoir

Cet objet représente un bouton poussoir qui permet d'activer ou de désactiver une valeur de type booléen.



#### Boîte de dialogue de l'objet "Bouton Poussoir" :



Les champs de saisie de cette boîte sont les suivants :

#### Expression

Indiquer dans ce champ le nom de la variable à contrôler. Cette variable doit être de type booléen

Exemple : SA10

Il est possible de spécifier l'adresse IP distante du noyau APIGRAF PLC dans l'expression. Si c'est le cas, c'est cette adresse qui sera prise en compte pour le rafraîchissement.

Exemple : !193.0.1.1 !SA10

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

#### Bouton « Hint »

Ce bouton permet de définir les paramètres de « Hint » de l'objet. Le texte « Hint » que l'on définit est le texte qui apparaît sur fond jaune lorsque l'on passe le curseur de la souris sur l'objet.

Quand on clique sur le bouton « Hint », le fenêtre suivante s'ouvre :



Cette fenêtre présente deux contrôles :

#### Option « Afficher le Hint »

Si cette option n'est pas cochée, le texte du Hint n'apparaîtra pas quand le curseur de la souris passera sur l'objet même si celui-ci est défini. Il faut donc cocher cette option si l'on veut utiliser le « Hint » de l'objet.

#### Fenêtre de saisie « Hint »

Cette fenêtre permet d'entrer le texte du Hint. Ce texte peut être un paramètre (exemple {1}).

#### Option "Bouton à accroche"

Si cette option est activée, appuyer sur le bouton activera la variable booléenne de l'expression associée. On ne pourra alors pas désactiver la variable en appuyant à nouveau sur le bouton.

#### Expression de neutralisation

Permet de spécifier la variable ou l'expression de neutralisation (Celle-ci doit obligatoirement être de type booléen). Le bouton poussoir sera neutralisé chaque fois que la variable ou l'expression sera égale à true.

Exemple : SA10

Il est possible de spécifier l'adresse IP distante du noyau APIGRAF PLC dans l'expression. Si c'est le cas, c'est cette adresse qui sera prise en compte pour le rafraîchissement.

Exemple : !193.0.1.1 !SA10

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

#### Option « Visible si neutralisé »

Si cette option est cochée, le bouton poussoir sera visible quand l'objet sera neutralisé. Cependant, l'opérateur ne pourra pas agir dessus.

Si cette option n'est pas cochée, l'objet sera invisible quand il sera neutralisé.

### 3.2.6.3 Objets de l'onglet FENETRES DE SAISIE

#### 3.2.6.3.1 Fenêtre de saisie

Cet objet permet de visualiser une variable numérique ou alphanumérique, et éventuellement de la modifier par saisie au clavier. Il est aussi appelé « fenêtre de saisie ».

#### Boîte de dialogue paramètre fenêtre d'édition - Onglet "Général"

Les champs de saisie de cette boîte sont les suivants :

### Expression

Indiquer dans ce champ le nom de la variable automate à visualiser ou à saisir.

*Exemple : R12*

Il est possible de spécifier l'adresse IP distante du noyau APIGRAF PLC dans l'expression. Si c'est le cas, c'est cette adresse qui sera prise en compte pour le rafraîchissement.

*Exemple : !193.0.1.1 !R12*

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

### Valeur min

Ce champ contient la limite inférieure admissible souhaitée lors de la saisie d'une variable numérique. Si cette limite n'est pas respectée lors de la saisie, l'utilisateur en sera averti par un message et ne pourra pas réellement achever la saisie.

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

### Valeur max

Ce champ contient la limite supérieure admissible souhaitée lors de la saisie d'une variable numérique. Si cette limite n'est pas respectée lors de la saisie, l'utilisateur en sera averti par un message et ne pourra pas réellement achever la saisie.

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

### Nb Décimales

Si une variable numérique décimale est affichée dans la fenêtre d'édition, cette option permet de choisir le nombre de décimales de la variable qui seront affichées.

### Option Supprimer les décimales superflues

Si une variable numérique décimale est affichée dans la fenêtre d'édition, cette option permet de ne pas afficher les dernières décimales qui seraient égales à 0.

### Couleur fond

Permet de spécifier la couleur de fond de la fenêtre d'édition.

### Caractères

Cette option permet de choisir la police de caractères à utiliser pour le texte affiché dans l'objet « Edition ». Elle provoque l'affichage de la boîte de dialogue « Police de caractères ».

### Option Taille verticale automatique

Adapte automatiquement la hauteur de la fenêtre de saisie en fonction de la police de caractère qui a été définie.

### Option Saisie Autorisée

Si cette option n'est pas cochée, l'objet permettra uniquement de visualiser la valeur de la variable, sans modification. Si elle est cochée, l'objet pourra alors être sélectionné et la valeur pourra être modifiée par une frappe au clavier.

### Affichage format Time/date OPTIMA PLC

Permet d'afficher une variable de type time, date, date\_and\_time ou time\_of\_day d'OPTIMA PLC directement au format Date/Heure.

Exemple :

12/02/2008 22:10:45

### Type Time OPTIMA PLC

Spécifie le type time de la variable OPTIMA PLC à afficher. Les choix suivants sont possibles :

Time, Date, Date\_and\_time, Time\_of day

### Format d'affichage

Ce champ permet de spécifier le format d'affichage Date/Heure dans la fenêtre de saisie.

Il est possible de sélectionner un format d'horodatage dans la liste déroulante associée à la fenêtre de saisie ou de saisir son propre format.

#### Exemples de format :

dd/mm/yyyy"---"hh:nn:ss	affiche 02/03/2002---18:23:15
hh:nn AM/PM	affiche 6:23PM
dddd" d" "mm" "yyyy	affiche Samedi 2 Mars 2002

### Bouton « Hint »

Ce bouton permet de définir les paramètres de « Hint » de l'objet. Le texte « Hint » que l'on définit est le texte qui apparaît sur fond jaune lorsque l'on passe le curseur de la souris sur l'objet.

Quand on clique sur le bouton « Hint », le fenêtre suivante s'ouvre :



Cette fenêtre présente deux contrôles :

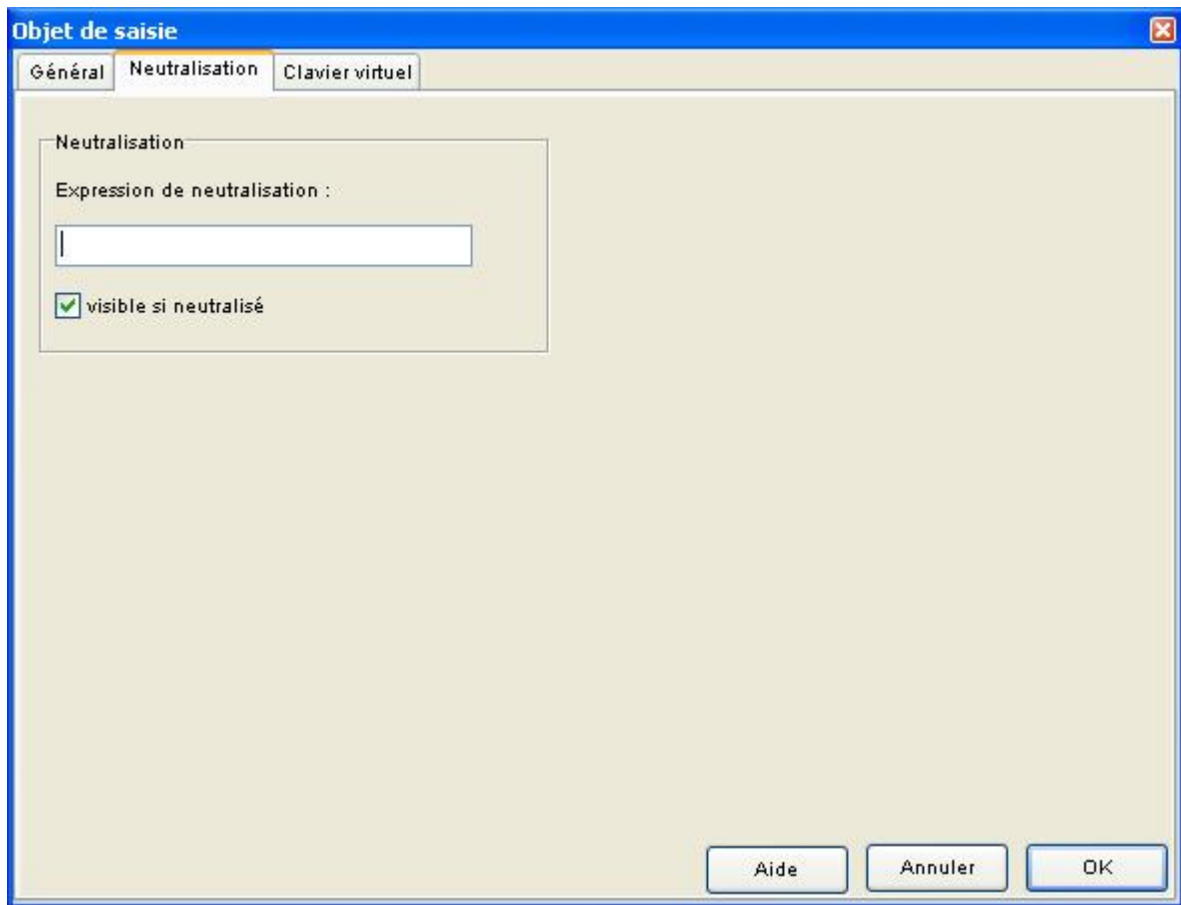
### Option « Afficher le Hint »

Si cette option n'est pas cochée, le texte du Hint n'apparaîtra pas quand le curseur de la souris passera sur l'objet même si celui-ci est défini. Il faut donc cocher cette option si l'on veut utiliser le « Hint » de l'objet.



**Fenêtre de saisie « Hint »**

Cette fenêtre permet d'entrer le texte du Hint. Ce texte peut être un paramètre (exemple {1}).

**Boîte de dialogue paramètre fenêtre d'édition - Onglet "Neutralisation"****Expression de neutralisation**

Permet de spécifier la variable ou l'expression de neutralisation (Celle-ci doit obligatoirement être de type booléen). La fenêtre d'édition sera neutralisée chaque fois que la variable ou l'expression sera égale à true.

*Exemple : SA10*

Il est possible de spécifier l'adresse IP distante du noyau APIGRAF PLC dans l'expression. Si c'est le cas, c'est cette adresse qui sera prise en compte pour le rafraîchissement.

*Exemple : !193.0.1.1 !SA10*

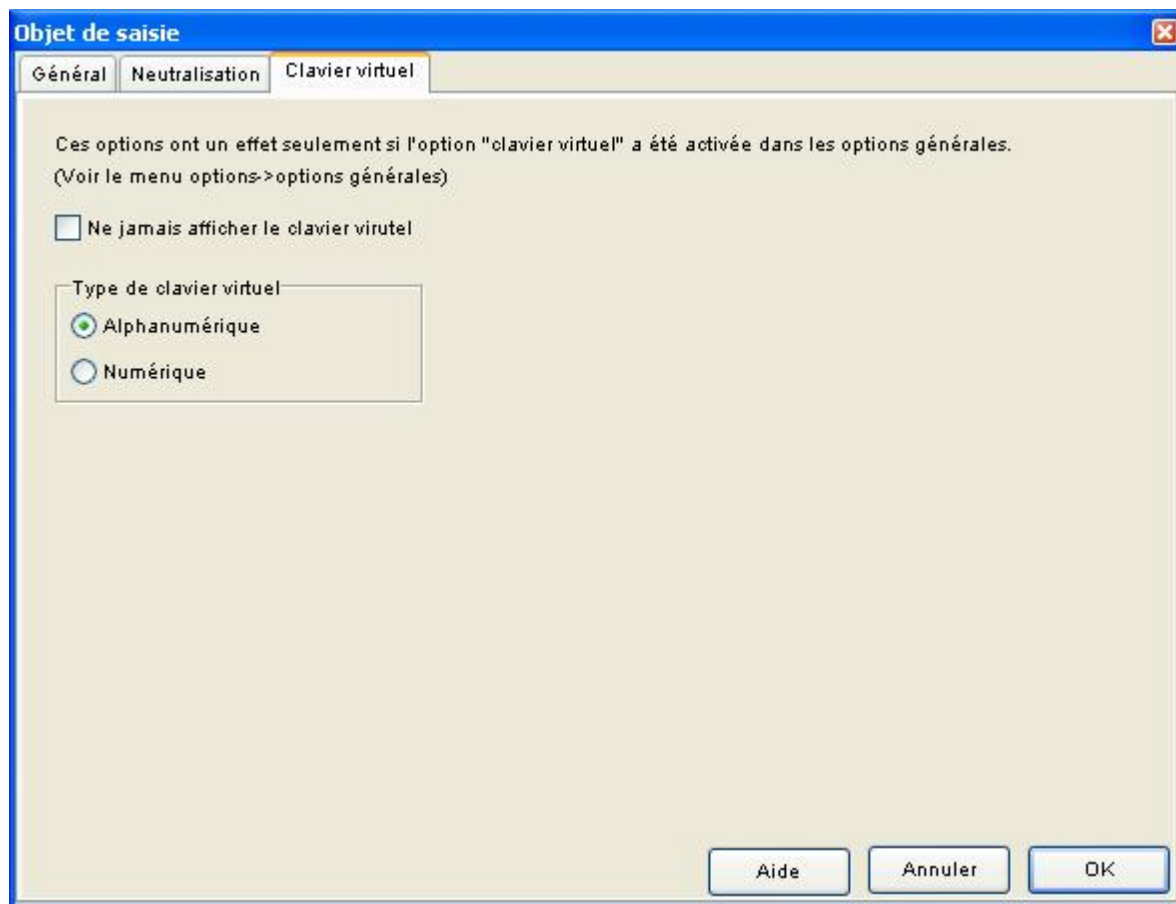
Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

**Option « Visible si neutralisé »**

Si cette option est active, la fenêtre sera visible quand celle-ci sera neutralisée. Il sera cependant impossible à l'opérateur de modifier le contenu de la fenêtre.

Si cette option n'est pas cochée, la fenêtre d'édition sera invisible quand elle sera neutralisée.

**Boîte de dialogue paramètre fenêtre d'édition - Onglet "Clavier virtuel"**



Le clavier virtuel est utilisé sur les écrans tactiles qui ne sont pas munis de clavier. C'est un clavier qui s'ouvre à l'écran à côté de la fenêtre de saisie et qui permet à l'utilisateur de saisir des caractères alphanumériques en cliquant sur les boutons de ce clavier.

Pour activer le clavier virtuel automatique dans votre projet, il faut aller dans le menu "Option" puis "Options générales" et activer l'option "clavier virtuel automatique".

Les paramètres de cette boîte de dialogue auront un effet seulement si l'option Clavier virtuel est active.

#### **Option "Ne jamais afficher la clavier virtuel"**

Cette option permet de désactiver le clavier virtuel pour une fenêtre de saisie en particulier.

#### **Choix "Type de clavier virtuel"**

Par défaut le clavier est de type alphanumérique, il est cependant possible de spécifier un clavier uniquement numérique pour une fenêtre de saisie en particulier.

Exemple : Clavier virtuel alphanumérique



### 3.2.6.3.2 Fenêtre de saisie spéciale

L'objet fenêtre d'édition spéciale est semblable à l'objet « Fenêtre d'édition » mais présente des fonctionnalités supplémentaires avec entre autres, la possibilité de choisir l'alignement du texte : Gauche, centré ou à droite.

#### Boîte de dialogue « Fenêtre d'édition spéciale » : Onglet général

Les champs de saisie de cette boîte sont les suivants :

#### **Expression**

Indiquer dans ce champ le nom de la variable automate à visualiser ou à saisir.

*Exemple* : R12

Il est possible de spécifier l'adresse IP distante du noyau APIGRAF PLC dans l'expression. Si c'est le cas, c'est cette adresse qui sera prise en compte pour le rafraîchissement.

*Exemple* : !193.0.1.1 !R12

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

### **Justification**

Permet de définir l'alignement du texte à l'intérieur de la fenêtre de saisie, les choix possibles sont :  
A gauche, centré, A droite.

### **Option Saisie Autorisée**

Si cette option n'est pas cochée, l'objet permettra uniquement de visualiser la valeur de la variable, sans modification. Si elle est cochée, l'objet pourra alors être sélectionné et la valeur pourra être modifiée par une frappe au clavier.

### **Option Taille verticale automatique**

Adapte automatiquement la hauteur de la fenêtre de saisie en fonction de la police de caractère qui a été définie.

### **Couleur fond**

Permet de spécifier la couleur de fond de la fenêtre d'édition.

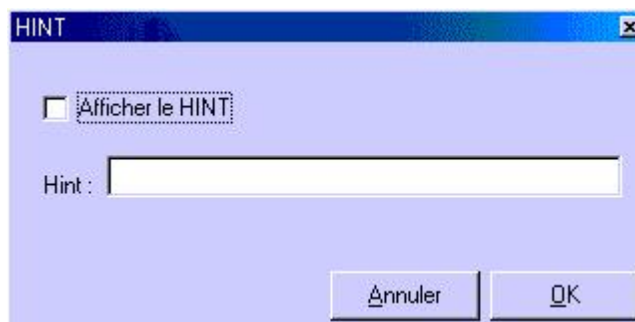
### **Caractères**

Cette option permet de choisir la police de caractères à utiliser pour le texte affiché dans l'objet « Edition ». Elle provoque l'affichage de la boîte de dialogue « Police de caractères ».

### **Bouton « Hint »**

Ce bouton permet de définir les paramètres de « Hint » de l'objet. Le texte « Hint » que l'on définit est le texte qui apparaît sur fond jaune lorsque l'on passe le curseur de la souris sur l'objet.

Quand on clique sur le bouton « Hint », la fenêtre suivante s'ouvre :



Cette fenêtre présente deux contrôles :

### **Option « Afficher le Hint »**

Si cette option n'est pas cochée, le texte du Hint n'apparaîtra pas quand le curseur de la souris passera sur l'objet même si celui-ci est défini. Il faut donc cocher cette option si l'on veut utiliser le « Hint » de l'objet.

### **Fenêtre de saisie « Hint »**

Cette fenêtre permet d'entrer le texte du Hint. Ce texte peut être un paramètre (exemple {1}).

### **Affichage format Time/date OPTIMA PLC**

Permet d'afficher une variable de type time, date, date\_and\_time ou time\_of\_day d'OPTIMA PLC directement au format Date/Heure.

Exemple :

12/02/2008 22:10:45

### **Type Time OPTIMA PLC**

Spécifie le type time de la variable OPTIMA PLC à afficher. Les choix suivants sont possibles :

Time, Date, Date\_and\_time, Time\_of day

### **Format d'affichage**

Ce champ permet de spécifier le format d'affichage Date/Heure dans la fenêtre de saisie.

Il est possible de sélectionner un format d'horodatage dans la liste déroulante associée à la fenêtre de saisie ou de saisir son propre format.

Exemples de format :

dd/mm/yyyy"---"hh:nn:ss	affiche 02/03/2002---18:23:15
hh:nn AM/PM	affiche 6:23PM
dddd" "d" "mm" "yyyy	affiche Samedi 2 Mars 2002

**Valeur min**

Ce champ contient la limite inférieure admissible souhaitée lors de la saisie d'une variable numérique. Si cette limite n'est pas respectée lors de la saisie, l'utilisateur en sera averti par un message et ne pourra pas réellement achever la saisie.

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

**Valeur max**

Ce champ contient la limite supérieure admissible souhaitée lors de la saisie d'une variable numérique. Si cette limite n'est pas respectée lors de la saisie, l'utilisateur en sera averti par un message et ne pourra pas réellement achever la saisie.

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

**Nb Décimales**

Si une variable numérique décimale est affichée dans la fenêtre d'édition, cette option permet de choisir le nombre de décimales de la variable qui seront affichées.

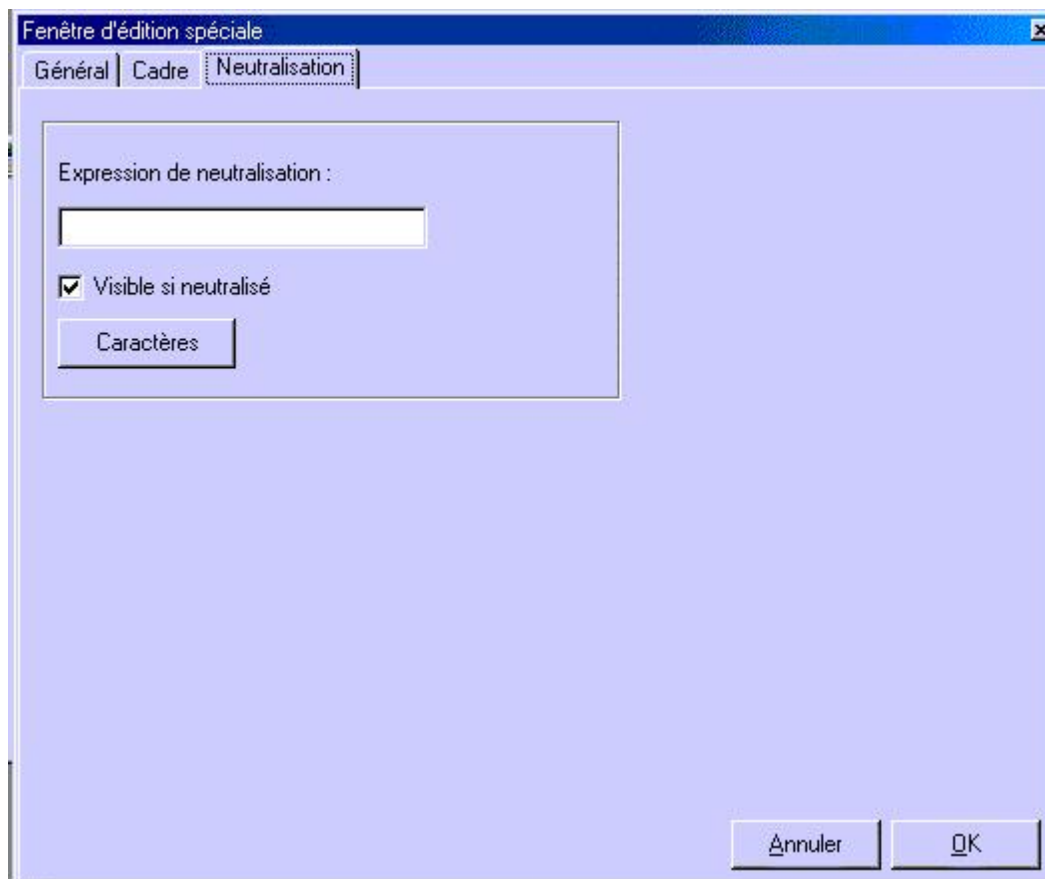
**Option Supprimer les décimales à 0**

Si une variable numérique décimale est affichée dans la fenêtre d'édition, cette option permet de ne pas afficher les dernières décimales qui seraient égales à 0.

**Boîte de dialogue « Fenêtre d'édition spéciale » : Onglet cadre****Option « Afficher le cadre »**

Si cette option n'est pas cochée, la fenêtre d'édition n'aura pas de cadre. Cette option peut être utile si on veut obtenir une fenêtre d'édition non visible.

**Boîte de dialogue « Fenêtre d'édition spéciale » : Onglet neutralisation**



### Expression de neutralisation

Permet de spécifier la variable ou l'expression de neutralisation (Celle-ci doit obligatoirement être de type booléen). La fenêtre d'édition spéciale sera neutralisée chaque fois que la variable ou l'expression sera égale à true.

*Exemple : SA10*

Il est possible de spécifier l'adresse IP distante du noyau APIGRAF PLC dans l'expression. Si c'est le cas, c'est cette adresse qui sera prise en compte pour le rafraîchissement.

*Exemple : !193.0.1.1 !SA10*

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

### Option « Visible si neutralisé »

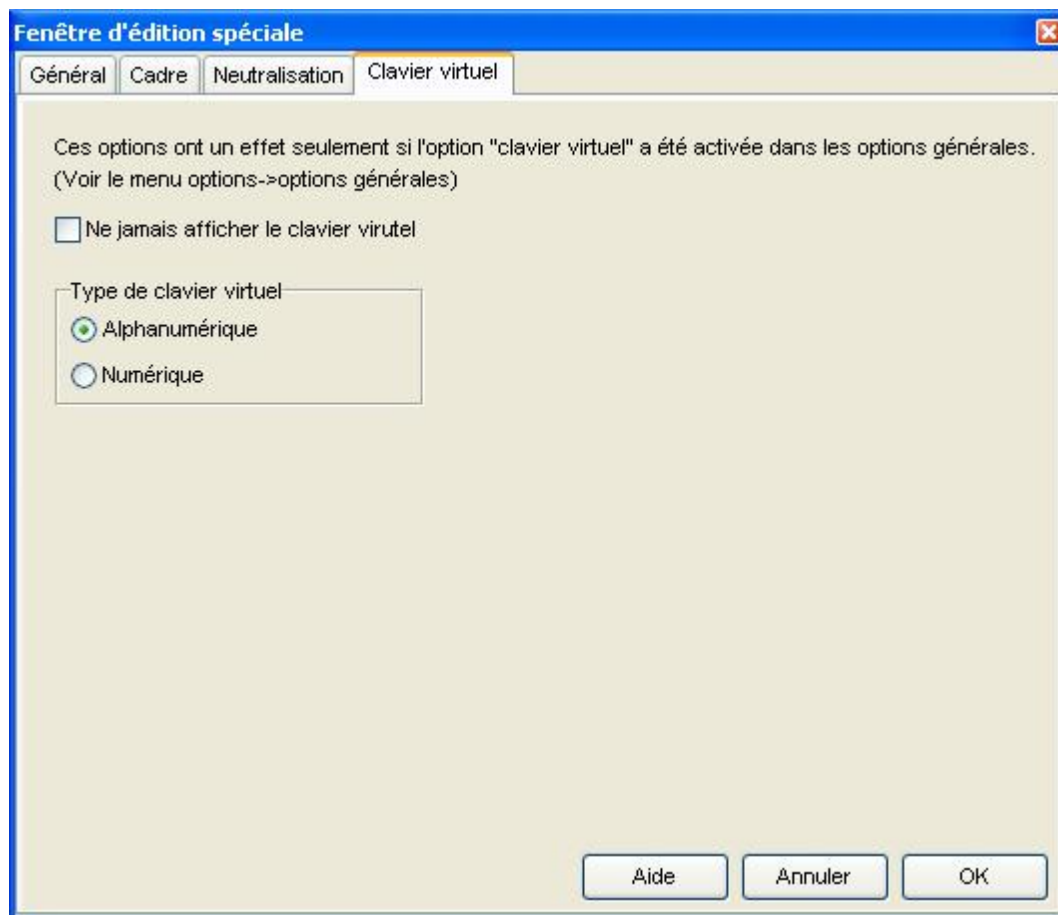
Si cette option est active, la fenêtre sera visible quand celle-ci sera neutralisée. Il sera cependant impossible à l'opérateur de modifier le contenu de la fenêtre.

Si cette option n'est pas cochée, la fenêtre d'édition sera invisible quand elle sera neutralisée.

### Bouton « Caractères »

Permet de spécifier une police de caractères pour la fenêtre quand celle-ci est neutralisée.

### Boîte de dialogue paramètre fenêtre d'édition - Onglet "Clavier virtuel"



Le clavier virtuel est utilisé sur les écrans tactiles qui ne sont pas munis de clavier. C'est un clavier qui s'ouvre à l'écran à côté de la fenêtre de saisie et qui permet à l'utilisateur de saisir des caractères alphanumériques en cliquant sur les boutons de ce clavier.

Pour activer le clavier virtuel automatique dans votre projet, il faut aller dans le menu "Option" puis "Options générales" et activer l'option "clavier virtuel automatique".

Les paramètres de cette boîte de dialogue auront un effet seulement si l'option Clavier virtuel est active.

#### **Option "Ne jamais afficher la clavier virtuel"**

Cette option permet pour de désactiver le clavier virtuel pour une fenêtre de saisie en particulier.

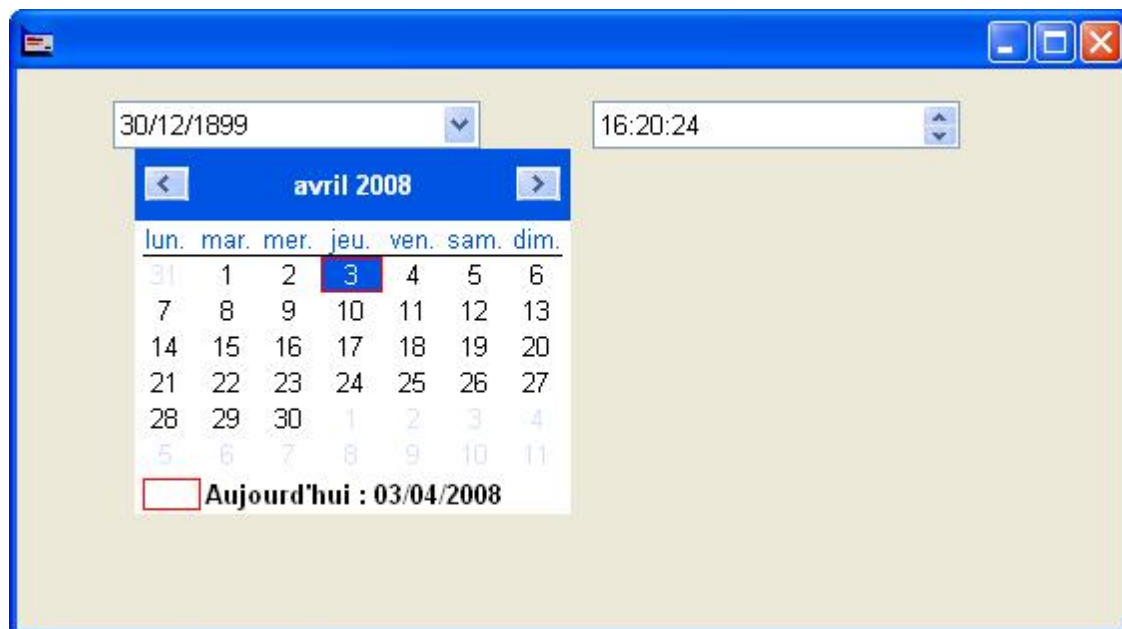
#### **Choix "Type de clavier virtuel"**

Par défaut le clavier est de type alphanumérique, il est cependant possible de spécifier un clavier uniquement numérique pour une fenêtre de saisie en particulier.

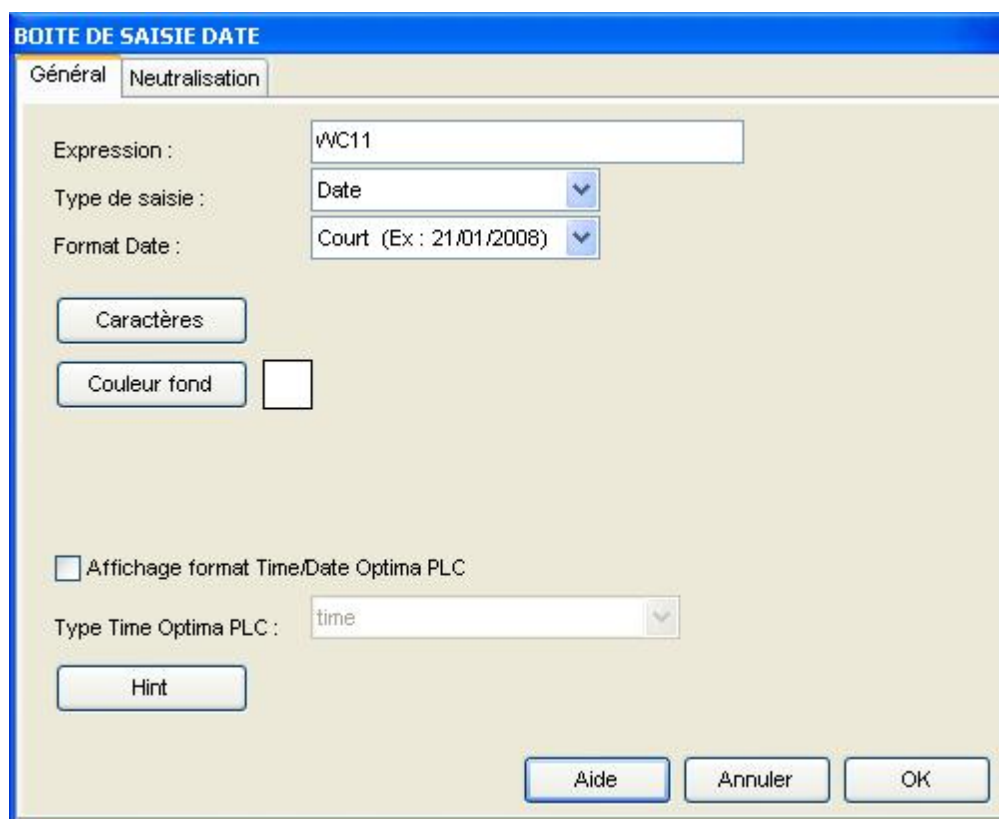
### **3.2.6.3.3 Fenêtre de saisie Date/Heure**

Cet objet permet de saisir une date ou une heure et de la modifier. Les expressions concernées sont de type string (exemple C12) ou de type time,date,date\_and\_time,time\_of\_day provenant d'optima PLC.

Dans le cas de la date, lors de la modification, un calendrier s'ouvre automatiquement pour permettre à l'utilisateur de choisir une nouvelle date à affecter à l'expression.



### Boîte de dialogue "Fenêtre de saisie Date/Heure" - Onglet Général



Les différents champs sont :

- **Expression**

Indiquer dans ce champ le nom d'une la variable de **type chaîne** ou de **type time,date,date\_and\_time,time\_of\_day** provenant d'**optima PLC**.

Cette variable sera affectée par la date/ heure sélectionnée dans l'objet calendrier. Si la date/heure contenue dans la variable change et est valide alors la date/heure courant(e) sera également modifié dans la fenêtre de saisie.

Exemple : C10  
WC10  
date1



Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

### Type de saisie

Spécifie le type de saisie désiré : Date ou heure.

Dans le cas de la date, lors de la modification, un calendrier s'ouvre automatiquement pour permettre à l'utilisateur de choisir une nouvelle date à affecter à l'expression.

### Format date

Ce champ est utilisé dans le cas le type de saisie sélectionné est le type Date. Il permet de spécifier le format d'affichage de la date :

Exemple format court : 21/02/2008

Exemple format long : Samedi 2 Février 2008

### Caractère

Permet de définir la police à utiliser pour afficher la date (Police, taille)

### Couleur fond

Permet de définir la couleur de fond de la fenêtre de saisie

### Affichage format Time/date OPTIMA PLC

Spécifie si l'expression est une variable de type time, date, date\_and\_time ou time\_of\_day d'OPTIMA PLC directement au format Date/Heure.

### Type Time OPTIMA PLC

Spécifie le type time de la variable OPTIMA PLC à afficher. Les choix suivants sont possibles :

Time, Date, Date\_and\_time, Time\_of day

### Hint

Ce bouton permet de définir les paramètres de « Hint » de l'objet. Le texte « Hint » que l'on définit est le texte qui apparaît sur fond jaune lorsque l'on passe le curseur de la souris sur l'objet.

Quand on clique sur le bouton « Hint », le fenêtre suivante s'ouvre :



Cette fenêtre présente deux contrôles :

#### Option « Afficher le Hint »

Si cette option n'est pas cochée, le texte du Hint n'apparaîtra pas quand le curseur de la souris passera sur l'objet même si celui-ci est défini. Il faut donc cocher cette option si l'on veut utiliser le « Hint » de l'objet.

**Fenêtre de saisie « Hint »**

Cette fenêtre permet d'entrer le texte du Hint. Ce texte peut être un paramètre (exemple {1}).

**Boîte de dialogue "Fenêtre de saisie Date/Heure" - Onglet Neutralisation****Expression de neutralisation**

Permet de spécifier la variable ou l'expression de neutralisation (Celle-ci doit obligatoirement être de type booléen). La boîte de saisie sera neutralisée chaque fois que la variable ou l'expression sera égale à true.

*Exemple : SA10*

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

**Visible si neutralisé**

Si cette option est cochée, la fenêtre de saisie sera visible quand elle sera neutralisée. Cependant, l'opérateur ne pourra pas agir dessus.

Si cette option n'est pas cochée, la fenêtre de saisie sera invisible quand elle sera neutralisée.

**3.2.6.3.4 Fenêtre d'édition avec menu déroulant**

Cet objet est similaire à l'objet fenêtre d'édition mais il présente en plus une liste de choix possibles pour la saisie.

L'opérateur peut saisir une valeur directement dans la fenêtre ou cliquer sur la flèche associée à la fenêtre pour choisir une saisie dans la liste déroulante.

**Boîte de dialogue Fenêtre d'édition avec menu déroulant - Onglet "Général"**

The screenshot shows a Windows-style dialog box titled "Edition avec liste déroulante". It has three tabs: "Général" (selected), "Neutralisation", and "Clavier virtuel". Inside the "Général" tab, there is an "Expression :" label followed by a text input field. Below this is a "Caractères" button. Then, there is a "Liste du menu déroulant :" label followed by a large empty list box. Below the list box are two checkboxes: "Affectation de l'index de l'élément à l'expression" (unchecked) and "Saisie autorisée" (checked). At the bottom left is a "Hint" button. At the bottom right are three buttons: "Aide", "Annuler", and "OK".

Les champs sont les suivants :

**« Expression »**

Indiquer dans ce champ le nom de la variable à contrôler. Cette variable doit être de type numérique entier

(Registre).

*Exemple* : R12

Il est possible de spécifier l'adresse IP distante du noyau APIGRAF PLC dans l'expression. Si c'est le cas, c'est cette adresse qui sera prise en compte pour le rafraîchissement.

*Exemple* : !193.0.1.1 !R12

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

#### « Liste du menu déroulant »

Spécifier ici la liste des choix possibles accessibles dans la liste associée à la fenêtre d'édition. Aller à la ligne pour séparer chaque élément de la liste.

Exemple :

Choix1

Choix2

Choix3

Il est possible de rentrer des références à des paramètres dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

#### Option « Affectation de l'index de l'élément à l'expression »

Par défaut cette option n'est pas cochée. C'est la valeur de l'élément de la liste qui est affecté à l'expression.

Exemple :

Expression : WR100

Liste :

100

200

500

1000

Si on choisit l'élément 200 de la liste alors la valeur 200 est affectée au registre WR100

Si l'option "Affectation de l'index à l'élément de la liste" est cochée alors si on choisit 200 dans la liste alors on va affecter la valeur 1 au registre WR100.

( 100 est l'élément d'index 0, 200 l'élément d'index 1, 500 l'élément d'index 2, ....). La valeur 1 correspond à l'index de l'élément sélectionné dans la liste déroulante.

#### Option « Saisie Autorisée »

Si cette option est cochée, l'opérateur pourra modifier le contenu de la fenêtre en saisissant une nouvelle valeur.

Si cette option n'est pas cochée, l'opérateur pourra choisir uniquement les valeurs contenues dans la liste déroulante.

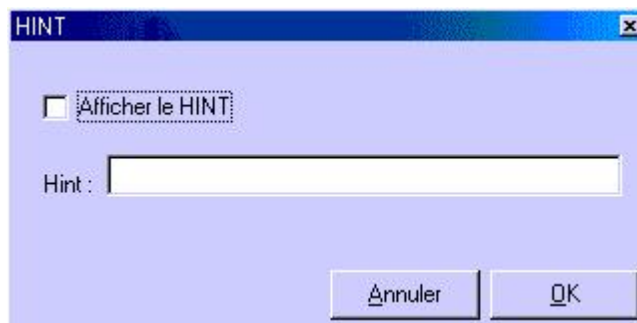
#### Bouton « Caractères » dans la fenêtre « Neutralisation »

Permet de définir la police de caractères utilisée pour afficher le fenêtre d'édition avec menu déroulant quand celle-ci est neutralisée et que l'option « Visible si neutralisé » est cochée.

#### Bouton « Hint »

Ce bouton permet de définir les paramètres de « Hint » de l'objet. Le texte « Hint » que l'on définit est le texte qui apparaît sur fond jaune lorsque l'on passe le curseur de la souris sur l'objet.

Quand on clique sur le bouton « Hint », le fenêtre suivante s'ouvre :



Cette fenêtre présente deux contrôles :

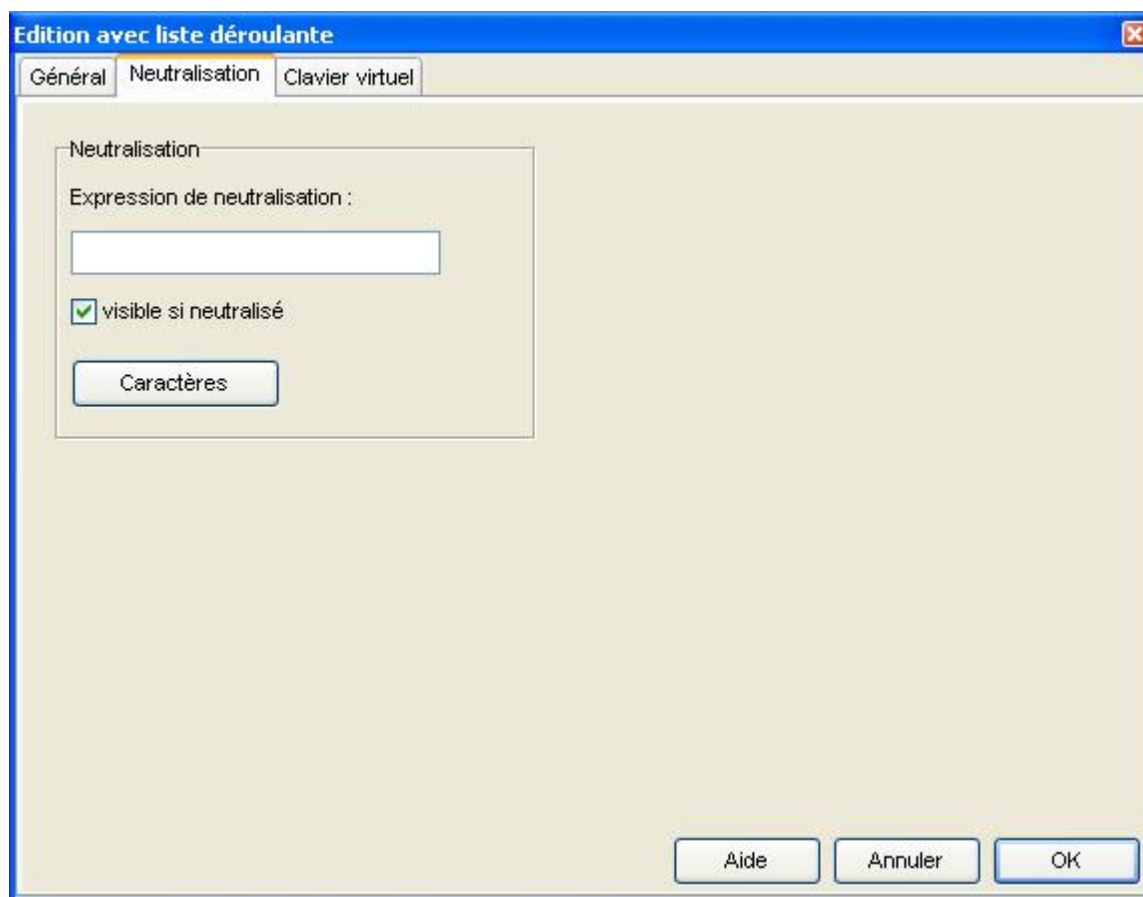
#### Option « Afficher le Hint »

Si cette option n'est pas cochée, le texte du Hint n'apparaîtra pas quand le curseur de la souris passera sur l'objet même si celui-ci est défini. Il faut donc cocher cette option si l'on veut utiliser le « Hint » de l'objet.

#### Fenêtre de saisie « Hint »

Cette fenêtre permet d'entrer le texte du Hint. Ce texte peut être un paramètre (exemple {1}).

#### Boîte de dialogue Fenêtre d'édition avec menu déroulant - Onglet "Neutralisation"



#### Expression de neutralisation

Permet de spécifier la variable ou l'expression de neutralisation (Celle-ci doit obligatoirement être de type booléen). Le fenêtr d'édition sera neutralisée chaque fois que la variable ou l'expression sera égale à true.

*Exemple : SA10*

Il est possible de spécifier l'adresse IP distante du noyau APIGRAF PLC dans l'expression. Si c'est le cas, c'est cette adresse qui sera prise en compte pour le rafraîchissement.

Exemple : !193.0.1.1 !SA10

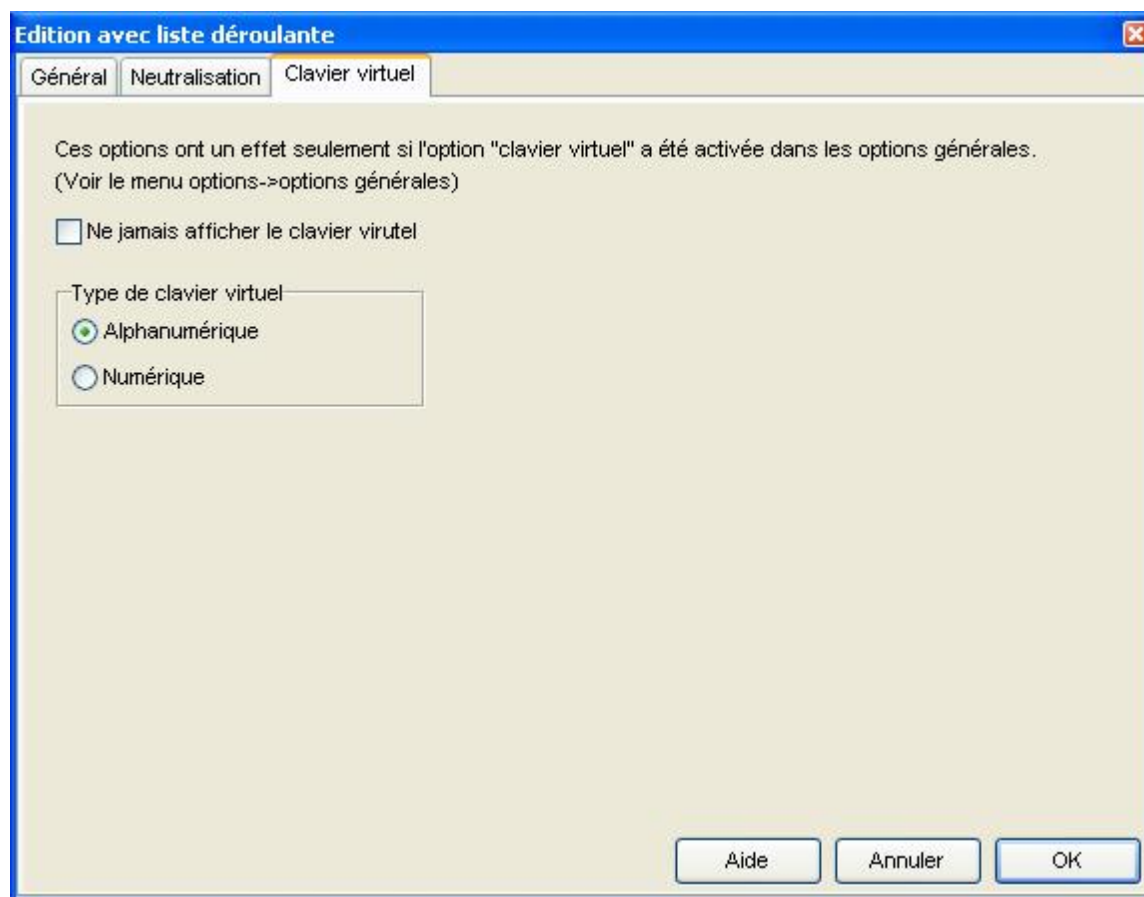
Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

#### Option « Visible si neutralisé »

Si cette option est active, la fenêtre sera visible quand celle-ci sera neutralisée. Il sera cependant impossible d'en modifier le contenu.

Si cette option n'est pas cochée, la fenêtre sera invisible quand elle sera neutralisée.

#### **Boîte de dialogue Fenêtre d'édition avec menu déroulant - Onglet "Clavier virtuel"**



Le clavier virtuel est utilisé sur les écrans tactiles qui ne sont pas munis de clavier. C'est un clavier qui s'ouvre à l'écran à côté de la fenêtre de saisie et qui permet à l'utilisateur de saisir des caractères alphanumériques en cliquant sur les boutons de ce clavier.

Pour activer le clavier virtuel automatique dans votre projet, il faut aller dans le menu "Option" puis "Options générales" et activer l'option "clavier virtuel automatique".

Les paramètres de cette boîte de dialogue auront un effet seulement si l'option Clavier virtuel est active.

#### **Option "Ne jamais afficher la clavier virtuel"**

Cette option permet pour de désactiver le clavier virtuel pour une fenêtre de saisie en particulier.

#### **Choix "Type de clavier virtuel"**

Par défaut le clavier est de type alphanumérique, il est cependant possible de spécifier un clavier uniquement numérique pour une fenêtre de saisie en particulier.

### 3.2.6.4 Objets de l'onglet CONTROLES

#### 3.2.6.4.1 Classeur

Cet Objet permet de créer des pages à onglets.

##### Boîte de dialogue paramètre objet classeur

Les options sont les suivantes :

##### **Nom de la page**

Indiquer dans ce champ le titre d'un nouvel onglet à ajouter dans le classeur.

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

##### **Ajouter**

Ce contrôle permet d'ajouter un onglet dans le classeur avec le nom qui est présent dans le champ « Nom de la page ».

##### **Pages présentes**

Cette fenêtre indique la liste des onglets présents dans le classeur. L'onglet en cours de sélection apparaît en surbrillance dans cette liste.

##### **Monter**

Déplace l'onglet d'un cran au-dessus dans la liste des onglets.

##### **Descendre**

Déplace l'onglet d'un cran au-dessous dans la liste des onglets.

##### **Renommer**

Permet de modifier le titre d'un onglet.

### Supprimer

Supprime l'onglet sélectionné dans la liste (en surbrillance).

### Caractères

Permet de choisir la police de caractère pour l'affichage des titres des onglets dans l'objet.

### Adresse IP par défaut :

Permet de spécifier une adresse IP par défaut pour chaque page du classeur. Veiller à bien sélectionner la page dans la liste (celle-ci doit apparaître en surbrillance) avant de modifier l'adresse IP par défaut. Tous les objets appartenant à la page du classeur utiliseront cette adresse par défaut pour leur mise à jour (à moins qu'une adresse IP soit redéfinie au niveau de l'objet appartenant à la page du classeur).

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

### Saisie du nom de serveur OPC par défaut

Cette fenêtre permet de saisir le nom du serveur OPC à superviser par défaut dans la page du classeur.

Attention : si une adresse IP est définie dans le champ du dessus, le nom du serveur OPC ne sera pas pris en compte.

### Bouton « Hint »

Ce bouton permet de définir les paramètres de « Hint » de l'objet. Le texte « Hint » que l'on définit est le texte qui apparaît sur fond jaune lorsque l'on passe le curseur de la souris sur l'objet.

Quand on clique sur le bouton « Hint », la fenêtre suivante s'ouvre :



Cette fenêtre présente deux contrôles :

### Option « Afficher le Hint »

Si cette option n'est pas cochée, le texte du Hint n'apparaîtra pas quand le curseur de la souris passera sur l'objet même si celui-ci est défini. Il faut donc cocher cette option si l'on veut utiliser le « Hint » de l'objet.

### Fenêtre de saisie « Hint »

Cette fenêtre permet d'entrer le texte du Hint. Ce texte peut être un paramètre (exemple {1}).

### Expression de neutralisation

Permet de spécifier la variable ou l'expression de neutralisation (Celle-ci doit obligatoirement être de type booléen). Le classeur

sera neutralisé chaque fois que la variable ou l'expression sera égale à true.

*Exemple : SA10*

Il est possible de spécifier l'adresse IP distante du noyau APIGRAF PLC dans l'expression. Si c'est le cas, c'est cette adresse qui sera prise en compte pour le rafraîchissement.

*Exemple : !193.0.1.1 !SA10*

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

### Option « Visible si neutralisé »

Si cette option est active, le classeur sera visible quand celle-ci sera neutralisée. Il sera cependant impossible de

changer les pages du classeur et d'utiliser les objets contenus dans le classeur.

Si cette option n'est pas cochée, le classeur sera invisible quand il sera neutralisé.

#### **Bouton « Caractères » dans la fenêtre « Neutralisation »**

Permet de définir la police de caractères utilisée pour afficher le classeur quand celui-ci est neutralisé et que l'option « Visible si neutralisé » est cochée.

#### **Position des onglets**

Permet de spécifier si les onglets doivent être en haut ou en bas du classeur.

#### **Option « Affichage des onglets sur plusieurs lignes »**

Dans le cas où les onglets sont nombreux, ils ne peuvent pas tous être affichés. Si cette option est cochée, tous les onglets seront affichés mais sur plusieurs lignes.

#### **Style des onglets**

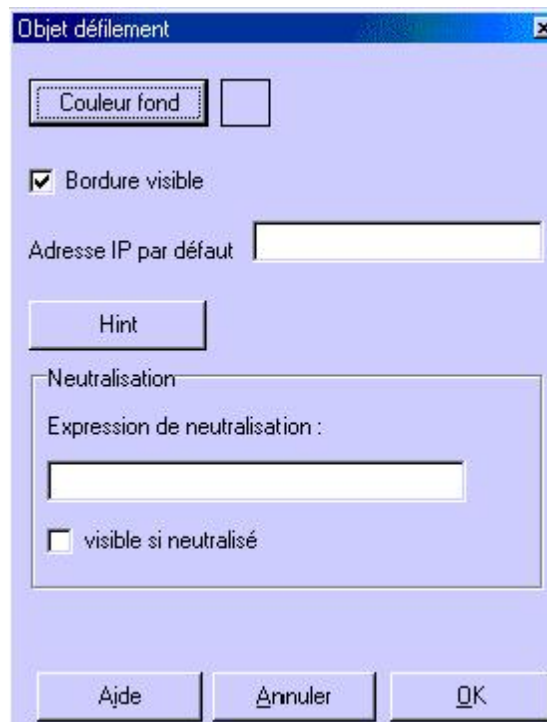
Permet de définir l'apparence des onglets :  
Onglets normaux, boutons ou boutons plats.

### **3.2.6.4.2 Défilement**

L'objet défilement permet de créer une « sous-fenêtre » dans la page. Si la taille des objets contenus dans l'objet défilement est plus importante que l'objet défilement lui-même, alors des barres de défilement apparaissent en bas et à droite de la fenêtre de défilement.

Si on veut inclure des objets dans un objet défilement, il faut créer ces objets à l'intérieur de la zone de défilement.

#### **Boîte de dialogue paramètre Objet Défilement**



Les options sont les suivantes :

#### **Couleur fond**

Permet de choisir la couleur de fond de la fenêtre de défilement.

#### **Option Bordure visible**

Si cette option n'est pas cochée, la bordure de l'objet défilement ne sera pas visible.

#### **Adresse IP par défaut**

Permet de spécifier une adresse IP par défaut pour la zone de défilement. Tous les objets appartenant à la zone



défilement utiliseront cette adresse par défaut pour leur mise à jour (à moins qu'une adresse IP soit redéfinie au niveau de l'objet appartenant à la zone de défilement ).

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

#### Expression de neutralisation

Permet de spécifier la variable ou l'expression de neutralisation (Celle-ci doit obligatoirement être de type booléen). La zone de défilement sera neutralisée chaque fois que la variable ou l'expression sera égale à true.

*Exemple : SA10*

Il est possible de spécifier l'adresse IP distante du noyau APIGRAF PLC dans l'expression. Si c'est le cas, c'est cette adresse qui sera prise en compte pour le rafraîchissement.

*Exemple : !193.0.1.1 !SA10*

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

#### Option « Visible si neutralisé »

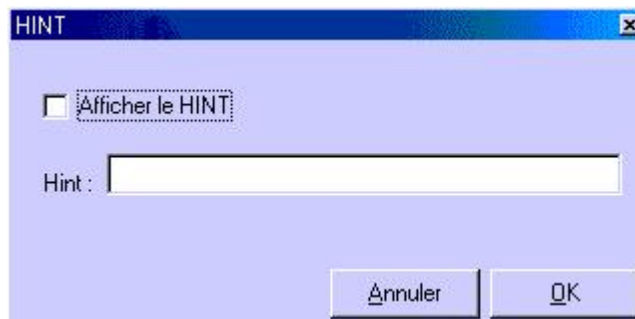
Si cette option est cochée, la zone de défilement sera visible quand elle sera neutralisée. Cependant, celle-ci ainsi que tous les objets contenus dans celle-ci ne seront pas accessibles à l'utilisateur.

Si cette option n'est pas cochée, la zone de défilement ainsi que tous les objets qu'elle contient seront invisibles quand la zone sera neutralisée.

#### Bouton « Hint »

Ce bouton permet de définir les paramètres de « Hint » de l'objet. Le texte « Hint » que l'on définit est le texte qui apparaît sur fond jaune lorsque l'on passe le curseur de la souris sur l'objet.

Quand on clique sur le bouton « Hint », le fenêtre suivante s'ouvre :



Cette fenêtre présente deux contrôles :

#### Option « Afficher le Hint »

Si cette option n'est pas cochée, le texte du Hint n'apparaîtra pas quand le curseur de la souris passera sur l'objet même si celui-ci est défini. Il faut donc cocher cette option si l'on veut utiliser le « Hint » de l'objet.

#### Fenêtre de saisie « Hint »

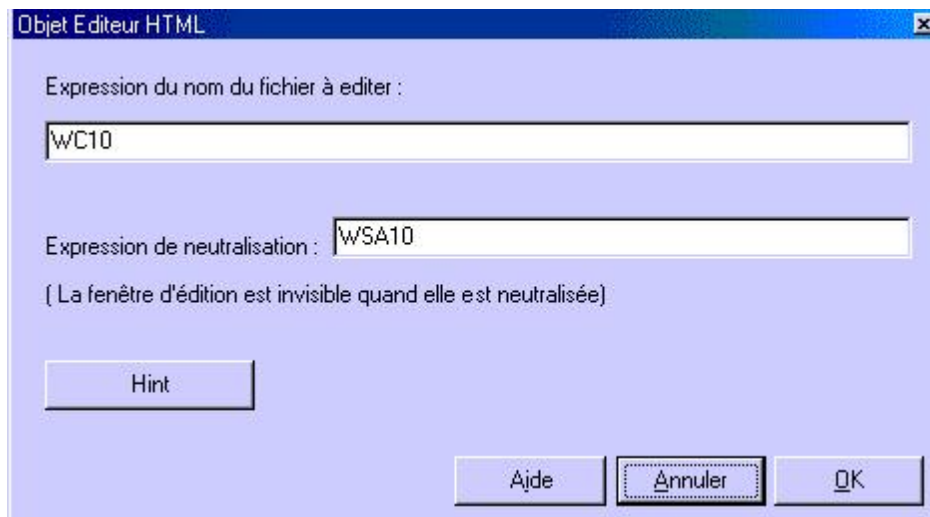
Cette fenêtre permet d'entrer le texte du Hint. Ce texte peut être un paramètre (exemple {1}).

### 3.2.6.4.3 Visionneur HTML

Cet objet permet de visualiser dans une fenêtre des fichiers HTML.

Le nom du fichier à visualiser peut être fixe ou être fonction d'une variable de type chaîne.

#### Boîte de dialogue "Visionneur HTML"



Les champs sont les suivants :

**Expression du nom du fichier à éditer :**

Ce paramètre donne le nom du fichier HTML à éditer. Il peut être constant :

'C:\README.HTML'

Dans ce cas n'oubliez pas de mettre le nom de fichiers entre côtes.

Il peut également dépendre d'une variable de type string :

C30

Il est possible de spécifier l'adresse IP distante du noyau APIGRAF PLC dans l'expression. Si c'est le cas, c'est cette adresse qui sera prise en compte pour le rafraîchissement.

Exemple : !193.0.1.1 !SA20

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

Dans ce cas, à chaque changement de la valeur de C30, le fichier correspondant à la valeur de C30 sera affiché.

**Bouton « Hint »**

Ce bouton permet de définir les paramètres de « Hint » de l'objet. Le texte « Hint » que l'on définit est le texte qui apparaît sur fond jaune lorsque l'on passe le curseur de la souris sur l'objet.

Quand on clique sur le bouton « Hint », la fenêtre suivante s'ouvre :



Cette fenêtre présente deux contrôles :

**Option « Afficher le Hint »**

Si cette option n'est pas cochée, le texte du Hint n'apparaîtra pas quand le curseur de la souris passera sur l'objet même si celui-ci est défini. Il faut donc cocher cette option si l'on veut utiliser le « Hint » de l'objet.

**Fenêtre de saisie « Hint »**

Cette fenêtre permet d'entrer le texte du Hint. Ce texte peut être un paramètre (exemple {1}).

### 3.2.6.4.4 Panneau

L'objet panneau est de type statique, il permet d'afficher un titre sous forme de panneau.

#### Boîte de dialogue paramètre Panneau

The dialog box 'Objet Panneau' is used for configuring a static panel object. It includes fields for text, font, background color, border visibility, text alignment, frame styles, frame thickness, neutralization expression, and default IP/OPC server addresses.

Les options propres à ce type d'objet sont :

#### Texte

Taper dans cet éditeur le texte à afficher dans l'objet.

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

#### Position Texte

Cette option permet de choisir la justification du texte dans l'objet, les choix possibles sont :

- Justifié à gauche : le texte sera aligné à gauche dans l'objet
- Centré : le texte sera centré dans l'objet panneau.
- Justifié à droite : le texte sera aligné à droite dans l'objet.

#### Caractères

Cette option permet de choisir la police de caractères à utiliser pour le texte affiché dans l'objet « Panneau ». Elle provoque l'affichage de la boîte de dialogue « Police de caractères ».

#### Couleur Fond

Permet de préciser la couleur de fond du panneau.

**Option bordure visible**

Si cette option n'est pas cochée, la bordure du panneau ne sera pas visible à l'écran. Cependant, si des cadres extérieurs et intérieurs sont définis dans l'objet, ceux-ci seront visibles.

**Cadre intérieur**

Permet de choisir un style de cadre intérieur, les différents choix sont :

- Aucun
- En creux
- En relief

**Cadre Extérieur**

Permet de choisir un style de cadre extérieur, les différents choix sont :

- Aucun
- En creux
- En relief

**Epaisseur des cadres (en pixel)**

Cette option permet de définir l'épaisseur de la bordure du panneau. La valeur par défaut est 1 pixel.

**Neutralisation Expression**

Cette option permet de définir une variable de neutralisation (variable booléenne). Lorsque la variable ou l'expression est évaluée à 1 alors le panneau sera invisible (ainsi que tous les objets sur le panneau).

Le panneau (ainsi que tous les objets créés sur le panneau) sera visible lorsque l'expression sera évaluée à 0.

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

**Expression de neutralisation**

Permet de spécifier la variable ou l'expression de neutralisation (Celle-ci doit obligatoirement être de type booléen). Le panneau sera neutralisé chaque fois que la variable ou l'expression sera égale à true.

*Exemple : SA10*

Il est possible de spécifier l'adresse IP distante du noyau APIGRAF PLC dans l'expression. Si c'est le cas, c'est cette adresse qui sera prise en compte pour le rafraîchissement.

*Exemple : !193.0.1.1 !SA10*

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

**Option « Visible si neutralisé »**

Si cette option est cochée, le panneau sera visible quand il sera neutralisé. Cependant, celui-ci ainsi que tous les objets qu'il contient ne seront pas accessibles à l'utilisateur.

Si cette option n'est pas cochée, le panneau ainsi que tous les objets qu'il contient seront invisibles quand le panneau sera neutralisé.

**Adresse IP par défaut :**

Permet de spécifier une adresse IP par défaut pour le panneau. Tous les objets appartenant au panneau utiliseront cette adresse par défaut pour leur mise à jour (à moins qu'une adresse IP soit redéfinie au niveau de l'objet appartenant au panneau ).

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

**Saisie du nom de serveur OPC par défaut**

Cette fenêtre permet de saisir le nom du serveur OPC à superviser par défaut dans le panneau. Attention : si une adresse IP est définie dans le champ du dessus, le nom du serveur OPC ne sera pas pris en compte.

**Bouton « Hint »**

Ce bouton permet de définir les paramètres de « Hint » de l'objet. Le texte « Hint » que l'on définit est le texte qui apparaît sur fond jaune lorsque l'on passe le curseur de la souris sur l'objet.

Quand on clique sur le bouton « Hint », la fenêtre suivante s'ouvre :



Cette fenêtre présente deux contrôles :

#### Option « Afficher le Hint »

Si cette option n'est pas cochée, le texte du Hint n'apparaîtra pas quand le curseur de la souris passera sur l'objet même si celui-ci est défini. Il faut donc cocher cette option si l'on veut utiliser le « Hint » de l'objet.

#### Fenêtre de saisie « Hint »

Cette fenêtre permet d'entrer le texte du Hint. Ce texte peut être un paramètre (exemple {1}).

### 3.2.6.4.5 Liste déroulante

La liste déroulante affiche une liste d'éléments dans une fenêtre munie d'un ascenseur pour faire défiler la liste. L'objet a l'apparence suivante :



Il y a plusieurs manières d'utiliser cette liste :

- Liste de textes simples : l'utilisateur entre une liste de textes prédéfinis ainsi qu'une variable automate est associée à la liste. L'index de l'élément sélectionné en mode execution est alors affecté à la valeur de la variable. ( valeur allant de 0 pour le premier élément à nbTotal-1 pour le dernier élément
- Liste d'expressions : l'utilisateur entre une liste de variables PLC, les valeurs évaluées de ces variables sont alors mise à jour dans la liste. De plus, l'utilisateur entre une variable automate qui est associée à la liste. L'index de l'élément sélectionné en mode execution est alors affecté à la valeur de la variable. ( valeur allant de 0 pour le premier élément à nbTotal-1 pour le dernier élément

**Exemple :** Liste d'expression R10,R12,R14 avec (R10=1,R12=3 et R14=15) , la liste affichera alors 3 éléments : 1,3,15.

Si l'utilisateur a défini l'expression principale R50, alors :

Quand l'utilisateur clique sur la valeur 1, 0 sera affecté à l'expression R50.

Quand l'utilisateur clique sur la valeur 3, 0 sera affecté à l'expression R50.

Quand l'utilisateur clique sur la valeur 15, 0 sera affecté à l'expression R50.

- Liste d'expressions indexées : Le mode est le même que le mode précédent à la différence suivante: L'utilisateur rentre une expression indexée (Exemple : C[\*]) , le symbole "\*" indique l'emplacement de l'indice. Il rentre également l'indice de début (exemple : 0) et l'indice de fin (exemple : 19). Dans cet exemple, les valeurs des éléments : C0,C1,C2,...C19 sont listées dans l'objet. De la même façon, l'indice de l'élément sélectionné dans la liste en mode exécution sera affecté à l'expression générale de l'objet liste.

### **Boîte de dialogue "Liste déroulante" avec liste de textes simples**

The dialog box 'Objet Liste' is shown with the following configuration:

- Expression :** VR5
- Caractères** button
- Type de liste:**
  - ☐ Liste d'expressions
  - ☐ Liste d'une expression indexée
  - ☒ Liste de textes simples
- Neutralisation:**
  - Expression de neutralisation :** (empty field)
  - ☒ visible si neutralisé
- Liste de textes simples:**
  - Zone N°1
  - Zone N°2
  - Quai 224
  - Quai 225
  - Quai 226
  - Quai 227
- Hint** button
- Annuler** and **OK** buttons

Les paramètres sont :

#### **Expression**

Entrer la variable numérique qui sera contrôlée par la liste déroulante. Cette variable sera affectée par la valeur de l'indice de l'élément sélectionné dans la liste. (de 0 à Count - 1)

*Exemple :* R10

Il est possible de spécifier l'adresse IP distante du noyau OPTIMA PLC ou APIGRAF PLC dans l'expression. Si c'est le cas, c'est cette adresse qui sera prise en compte pour le rafraîchissement.

*Exemple :* !193.0.1.1!R10

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

Si la variable provient d'un serveur OPC, le bouton "OPC" permet d'explorer le serveur OPC et de trouver la variable à superviser dans un explorateur.

### Caractères

Cette option permet de choisir la police de caractères à utiliser pour les textes affichés dans liste. Elle provoque l'affichage de la boîte de dialogue « Police de caractères ».

### Type de liste

Permet de sélectionner le mode de fonctionnement de l'objet ( voir ci-dessus en haut de cette page la présentation générale). Suivant le type de liste sélectionné, un mémo de saisie s'affiche à l'écran pour entrer tous les éléments de la liste (textes simples ou expressions). Entrer dans le mémo un seul élément par ligne.

Dans le cas du type de liste "Liste d'une expression indicée", trois fenêtres de saisie apparaissent sur le formulaire : la saisie de l'expression indicée (Exemple : C[\*]), la saisie de l'index de début (exemple : 0 ou 1) et l'indice de l'index de fin (exemple : 20).

### Expression de neutralisation

Permet de spécifier la variable ou l'expression de neutralisation (Celle-ci doit obligatoirement être de type booléen). La liste déroulante sera neutralisée à chaque fois que la variable ou l'expression sera égale à true.

*Exemple : SA10*

Il est possible de spécifier l'adresse IP distante du noyau APIGRAF PLC dans l'expression. Si c'est le cas, c'est cette adresse qui sera prise en compte pour le rafraîchissement.

*Exemple : !193.0.1.1 !SA10*

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

### Option « Visible si neutralisé »

Si cette option est cochée, la liste déroulante sera visible quand elle sera neutralisée. Cependant, l'utilisateur ne pourra pas sélectionner d'éléments dans liste.

Si cette option n'est pas cochée, la liste déroulante sera invisible quand elle sera neutralisée.

### Bouton « Hint »

Ce bouton permet de définir les paramètres de « Hint » de l'objet. Le texte « Hint » que l'on définit est le texte qui apparaît sur fond jaune lorsque l'on passe le curseur de la souris sur l'objet.

Quand on clique sur le bouton « Hint », le fenêtre suivante s'ouvre :



Cette fenêtre présente deux contrôles :

### Option « Afficher le Hint »

Si cette option n'est pas cochée, le texte du Hint n'apparaîtra pas quand le curseur de la souris passera sur l'objet même si celui-ci est défini. Il faut donc cocher cette option si l'on veut utiliser le « Hint » de l'objet.

#### Fenêtre de saisie « Hint »

Cette fenêtre permet d'entrer le texte du Hint. Ce texte peut être un paramètre (exemple {1}).

#### 3.2.6.4.6 Zone de clic

La zone de clic est une zone délimitée lors de la conception de la page de supervision et invisible pendant la phase d'exécution.

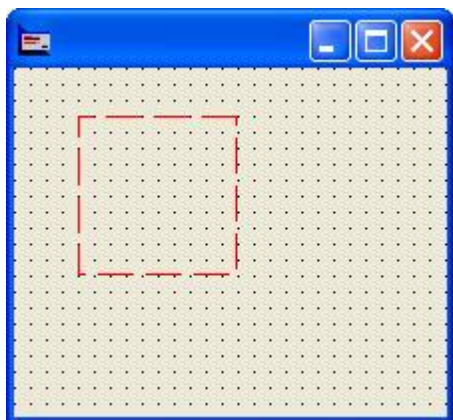
Cette zone de clic peut répondre aux clics de souris de plusieurs manières :

Suite à l'enfoncement du bouton gauche de la souris sur la zone, des macros peuvent être exécutées  
Suite au relachement du bouton gauche de la souris sur la zone, d'autres macros peuvent être exécutées.

Une variable automate ou expression peut également être associée à la zone de clic de sorte que la variable soit mise à 1 pendant l'enfoncement du bouton gauche de la souris sur la zone puis remise à 0 lors du relachement du bouton de la souris (Mode monostable)

En mode bistable, le premier clic met à 1 la variable et le clic suivant remet à 0 la variable.


En phase de conception dans l'atelier de supervision, l'objet "zone de clic" a l'apparence suivante :



#### Boîte de dialogue "Zone de clic"



**ZONE DE CLICK**

Expression  

Type d'activation  
☐ Bistable  
☒ Monostable

Macro enfacement

Macro relâchement

Les paramètres sont :

### Expression (facultatif)

Entrer la variable booléenne qui sera contrôlée par la zone de clic.

*Exemple* : SA10

Il est possible de spécifier l'adresse IP distante du noyau OPTIMA PLC ou APIGRAF PLC dans l'expression. Si c'est le cas, c'est cette adresse qui sera prise en compte pour le rafraîchissement.

*Exemple* : !193.0.1.1 !SA10

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

Si la variable provient d'un serveur OPC, le bouton "OPC" permet d'explorer le serveur OPC et de trouver la variable à superviser dans un explorateur.

L'utilisateur peut laisser ce champ vide dans le cas où la zone de clic lancera uniquement des macros.

*Exemple* : L'utilisateur veut ouvrir une autre page de supervision "TEST" en cliquant dans la zone de clic. Il peut alors simplement entrer dans le champ "Macro relâchement": SUPERG(TEST)

### Type d'activation : Bistable ou monostable

Dans le cas où une expression est définie, il est possible de choisir le type d'activation de la variable booléenne.

En mode monostable, la variable est mise à 1 tant que le bouton gauche de la souris est enfoncé sur la zone de clic et remise à 0 dès que le bouton est relâché.

En mode bistable, la variable est mise à 1 lors du premier clic et remise à 0 lors du clic suivant.

### Macro enfacement

Entrer la ou les macros à exécuter lors de l'enfoncement du bouton gauche de la souris sur la zone de clic.

Si il y a plusieurs macros, entrer une seule macro par ligne.

**Macro relâchement**

Entrer la ou les macros à exécuter lors du relâchement du bouton gauche de la souris.

Si il y a plusieurs macros, entrer une seule macro par ligne.

**Bouton « Hint »**

Ce bouton permet de définir les paramètres de « Hint » de l'objet. Le texte « Hint » que l'on définit est le texte qui apparaît sur fond jaune lorsque l'on passe le curseur de la souris sur l'objet.

Quand on clique sur le bouton « Hint », la fenêtre suivante s'ouvre :



Cette fenêtre présente deux contrôles :

**Option « Afficher le Hint »**

Si cette option n'est pas cochée, le texte du Hint n'apparaîtra pas quand le curseur de la souris passera sur l'objet même si celui-ci est défini. Il faut donc cocher cette option si l'on veut utiliser le « Hint » de l'objet.

**Fenêtre de saisie « Hint »**

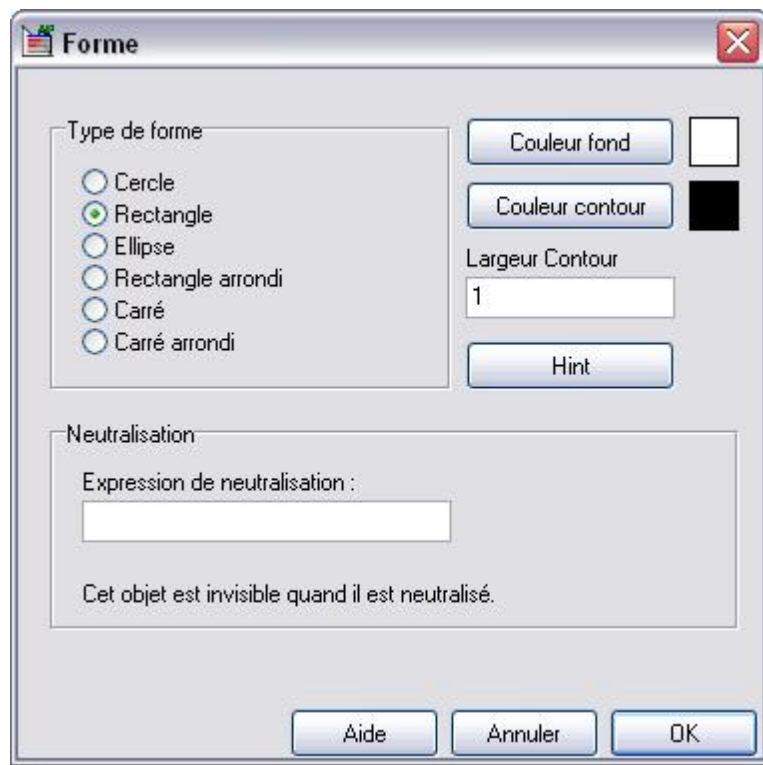
Cette fenêtre permet d'entrer le texte du Hint. Ce texte peut être un paramètre (exemple {1}).

### 3.2.6.5 Objets de l'onglet DESSIN

#### 3.2.6.5.1 Forme

Cet objet permet d'afficher une forme graphique.

**Boîte de dialogue Forme**



Les options sont les suivantes :

#### Type de forme

Permet de choisir le type de forme, les choix possibles sont :

- Cercle
- Ellipse
- Rectangle
- Rectangle arrondi
- Carré
- Carré arrondi

#### Couleur Fond

Cette option permet de régler la couleur de fond de la forme.

#### Couleur contour

Cette option permet de régler la couleur du contour de la forme.

#### Largeur contour (en pixels)

Précise la largeur du contour de la forme, la valeur par défaut est 1 pixel.

#### Bouton « Hint »

Ce bouton permet de définir les paramètres de « Hint » de l'objet. Le texte « Hint » que l'on définit est le texte qui apparaît sur fond jaune lorsque l'on passe le curseur de la souris sur l'objet.

Quand on clique sur le bouton « Hint », le fenêtre suivante s'ouvre :



Cette fenêtre présente deux contrôles :

#### Option « Afficher le Hint »

Si cette option n'est pas cochée, le texte du Hint n'apparaîtra pas quand le curseur de la souris passera sur l'objet même si celui-ci est défini. Il faut donc cocher cette option si l'on veut utiliser le « Hint » de l'objet.

#### Fenêtre de saisie « Hint »

Cette fenêtre permet d'entrer le texte du Hint. Ce texte peut être un paramètre (exemple {1}).

#### Expression de neutralisation

Permet de spécifier la variable ou l'expression de neutralisation (Celle-ci doit obligatoirement être de type booléen). La forme sera invisible chaque fois que la variable ou l'expression sera égale à true.

*Exemple : SA10*

Il est possible de spécifier l'adresse IP distante du noyau APIGRAF PLC dans l'expression. Si c'est le cas, c'est cette adresse qui sera prise en compte pour le rafraîchissement.

*Exemple : !193.0.1.1 !SA10*

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

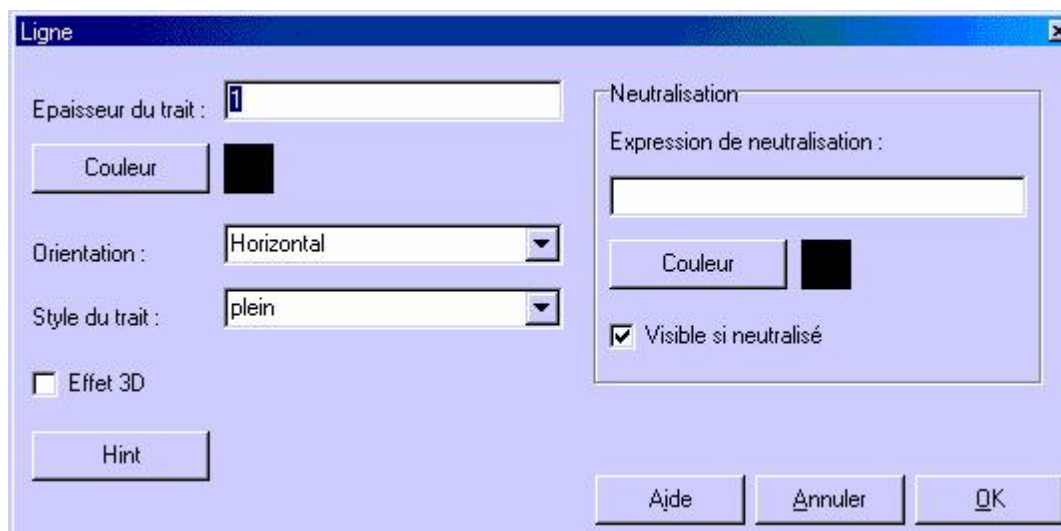
### 3.2.6.5.2 Ligne

Cet objet graphique permet de tracer des lignes sur une page de supervision.

Les lignes d'épaisseur égale à 1 peuvent être affichées en pointillés, en tirets, en trait plein, en tirets alternés avec des pointillés.

Les lignes d'épaisseur supérieure à 1 peuvent être affichées avec un effet 3D, qui donne un aspect de tube à celles-ci.

#### Boîte de dialogue « Ligne »



Les champs sont :

#### Epaisseur du trait

Spécifie l'épaisseur de la ligne.

Attention : Seules les lignes d'épaisseur égale à 1 peuvent être affichées en pointillés ou en tirets. Néanmoins, les lignes d'épaisseur supérieure à 1 peuvent être affichées avec un effet 3D.

#### Bouton « Couleur »

Permet de choisir la couleur de la ligne.

#### Contrôle «Orientation »

Spécifie l'orientation de la ligne. La ligne peut être en effet Horizontale, verticale, oblique descendante ou oblique montante.

Dans le cas où la ligne est oblique, le fait d'étirer à l'aide de la souris le cadre de l'objet ligne modifiera la pente de la ligne.

#### Contrôle « Style de trait »

Permet pour les lignes d'épaisseur 1 de les afficher en pointillés ou en tirets. Ce contrôle apparaît dans la boîte de dialogue uniquement si l'épaisseur de la ligne est 1.

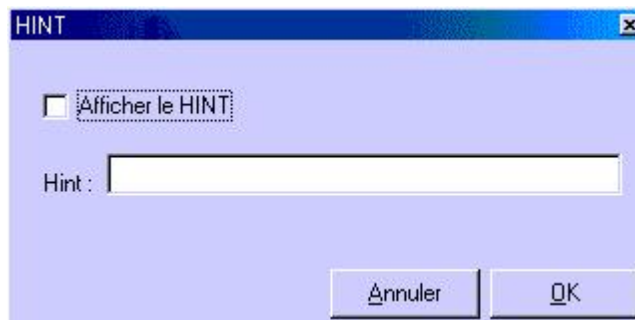
#### Option « Effet 3D »

Si cette option est cochée et si l'épaisseur de la ligne est supérieure à 1, alors la ligne apparaîtra avec un effet 3D qui lui donne un aspect « Tube »

#### Bouton « Hint »

Ce bouton permet de définir les paramètres de « Hint » de l'objet. Le texte « Hint » que l'on définit est le texte qui apparaît sur fond jaune lorsque l'on passe le curseur de la souris sur l'objet.

Quand on clique sur le bouton « Hint », la fenêtre suivante s'ouvre :



Cette fenêtre présente deux contrôles :

#### Option « Afficher le Hint »

Si cette option n'est pas cochée, le texte du Hint n'apparaîtra pas quand le curseur de la souris passera sur l'objet même si celui-ci est défini. Il faut donc cocher cette option si l'on veut utiliser le « Hint » de l'objet.

#### Fenêtre de saisie « Hint »

Cette fenêtre permet d'entrer le texte du Hint. Ce texte peut être un paramètre (exemple {1}).

#### Expression de neutralisation

Permet de spécifier la variable ou l'expression de neutralisation (Celle-ci doit obligatoirement être de type booléen). La ligne sera neutralisée chaque fois que la variable ou l'expression sera égale à true.

*Exemple : SA10*

Il est possible de spécifier l'adresse IP distante du noyau APIGRAF PLC dans l'expression. Si c'est le cas, c'est cette adresse qui sera prise en compte pour le rafraîchissement.

*Exemple : !193.0.1.1 !SA10*

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

#### Couleur

Ce bouton permet de définir la couleur de la ligne quand celle-ci est neutralisée et que l'option « Visible si neutralisé » est cochée.

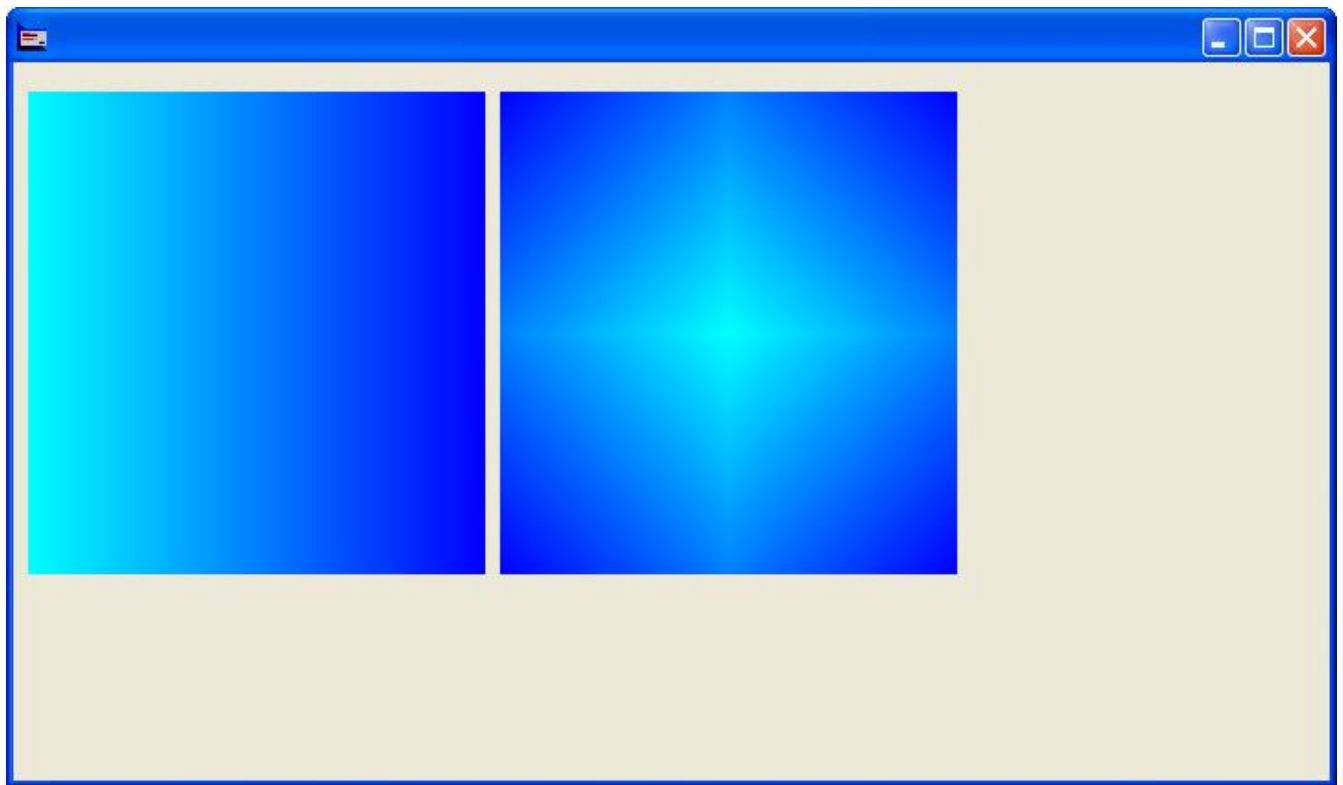
#### Option « Visible si neutralisé »

Si cette option est active, la ligne sera visible quand celle-ci sera neutralisée. Elle aura cependant la couleur définie dans la fenêtre de neutralisation.

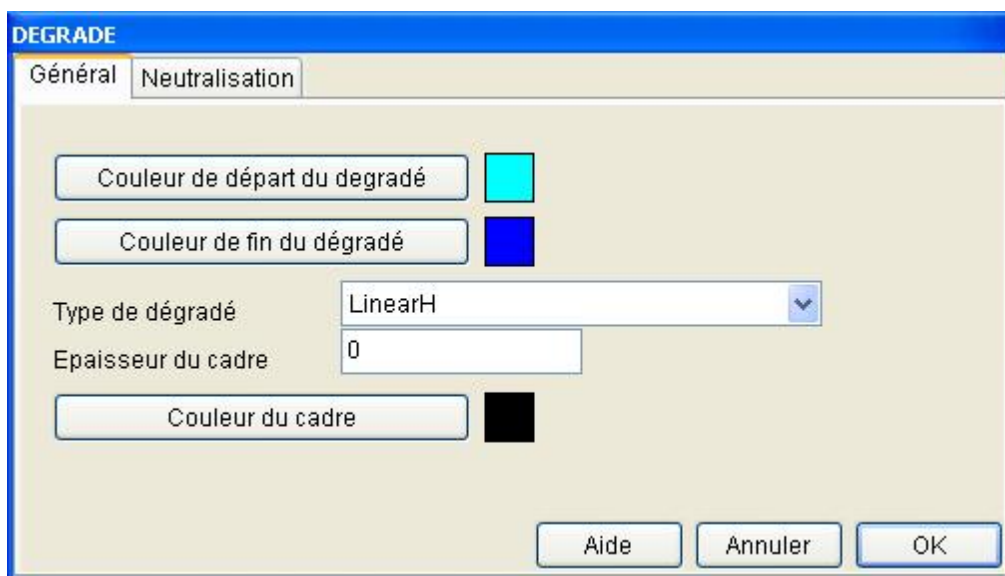
Si cette option n'est pas cochée, la ligne sera invisible quand elle sera neutralisée.

### 3.2.6.5.3 Dégradé

Cet objet graphique dessine un dégradé sur une page de supervision.



#### **Boîte de dialogue " Dégradé" - Général :**



#### **Couleur de départ dégradé et Couleur de fin dégradé :**

Ces deux boutons permettent de sélectionner les couleurs utilisées pour dessiner le dégradé

#### **Type de dégradé :**

Spécifie le type de dégradé : circulaire, horizontal, vertical , etc.

#### **Épaisseur du cadre :**

Il est possible d'afficher un cadre autour du dégradé et d'en spécifier l'épaisseur. Par défaut, il n'y a pas de cadre ( Épaisseur = 0 ).

#### **Couleur de cadre :**

Dans le cas où l'épaisseur du cadre est non nulle, il est possible de spécifier la couleur du cadre.

### 3.2.6.6 Objets de l'onglet GRAPHIQUE

#### 3.2.6.6.1 Graphe

L'objet graphe permet d'afficher des courbes ou Histogrammes à partir de tableaux de mesures présents sur OPTIMA PLC ou APIGRAF PLC. Un Objet graphe peut afficher jusqu'à 9 courbes.

En mode exécution, les courbes peuvent être imprimées. (Voir plus loin : Impression)

#### Boîte de dialogue graphe - Onglet "Général"

**Objet graphe**

Général | Axes et Echelles | Titre | Legendes | Fond | Neutralisation | Seuils | Impression | Zoom | 3D | Autres

Tableaux de mesures sur axe Y

Nom	Unité
TEST1	
TEST2	
TEST3	

Ajouter Supprimer

Tableau Axe X (facultatif)

Aucun

Tableau Axe X

☐ Type Facteur seconde

Format des dates/temps : dd/mm/yy "hh:nn"

Hint

Type d'affichage

☒ Traits

☐ Histogramme

Type de barre

☐ Rectangle

☐ Rectangle dégradé

☐ Pyramide

☐ Pyramide inversée

☐ Flèche

Echelle en Y

☒ Axe Y1

☐ Axe Y2

Options d'affichage

Couleur de la courbe

Aide Annuler OK

Les paramètres sont les suivants :

#### **Bouton « Ajouter »**

Permet d'ajouter une nouvelle courbe au graphe en spécifiant le nom du tableau de mesure correspondant.

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

#### **Bouton « Supprimer »**

Supprime du graphe, la courbe correspondant au tableau de mesure sélectionné dans la liste des tableaux de mesure.

#### **Tableau Axe X**

Ce champ est facultatif. Ce tableau permet de spécifier les valeurs en X des tableaux de mesures en Y.

Si celui-ci n'est pas renseigné, les valeurs en X des tableaux de mesure en Y seront les indices des tableaux en Y : 1,2,3 etc..

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

#### Option « Type facteur seconde »

Cocher cette option si les données du tableau en X sont de type facteur seconde (type APIGRAF pour l'horodatage). Cette option doit être cochée si le tableau des temps provient de l'application APIMONITOR (SURVEYTIME1, SURVEYTIME2, ...). Si cette option est cochée, les valeurs de l'axe des X apparaîtront sous forme de date/heure. Il est possible de modifier le format de cet affichage dans la fenêtre « Format des dates/temps »

#### Format des dates/temps

Il est possible de sélectionner un format d'horodatage dans la liste déroulante associée à la fenêtre de saisie ou de saisir son propre format.

#### Exemples de format :

dd/mm/yyyy"---"hh:nn:ss affiche 02/03/2002---18:23:15

hh:nn AM/PM affiche 6:23PM

dddd "d" "mm" "yyyy" affiche Samedi 2 Mars 2002

#### Type d'affichage

Permet de sélectionner le type d'affichage pour la courbe sélectionnée dans la liste des tableaux de mesure. Il y a deux possibilités

- Traits
- Histogramme

#### Type de barre

Cette option n'est accessible que si le type de graphe est un histogramme. Elle permet de sélectionner le type de barres du graphe.

#### Echelle en Y

Il y a 2 échelles verticales disponibles dans le graphe, une à gauche (Y1) et une à droite (Y2). Ce champ permet de spécifier quelle échelle verticale sera utilisée pour l'affichage de la courbe.

#### Couleur de la courbe

Ce bouton permet de spécifier la couleur de la courbe sélectionnée dans la liste des tableaux de mesure.

#### Bouton « Hint »

Ce bouton permet de définir les paramètres de « Hint » de l'objet. Le texte « Hint » que l'on définit est le texte qui apparaît sur fond jaune lorsque l'on passe le curseur de la souris sur l'objet.

Quand on clique sur le bouton « Hint », le fenêtre suivante s'ouvre :



Cette fenêtre présente deux contrôles :

#### Option « Afficher le Hint »

Si cette option n'est pas cochée, le texte du Hint n'apparaîtra pas quand le curseur de la souris passera sur l'objet même si celui-ci est défini. Il faut donc cocher cette option si l'on veut utiliser le « Hint » de l'objet.



### Fenêtre de saisie « Hint »

Cette fenêtre permet d'entrer le texte du Hint. Ce texte peut être un paramètre (exemple {1}).

### Boîte de dialogue graphe - Onglet "Axes et Echelles"

Objet graphe

Général Axes et Echelles Titre Legendes Fond Neutralisation Seuils Impression Zoom 3D Autres

Echelles

☒ Echelle X Automatique  
X Min 0  
X Max 10000  
Coefficient d'affichage en X 1

☒ Echelle Y1 Automatique  
Y Min 0  
Y Max 10000  
Coefficient d'affichage en Y1 1

☒ Echelle Y2 Automatique  
Y Min 0  
Y Max 10000  
Coefficient d'affichage en Y2 1

Aide Annuler OK

### Echelles

Permet de spécifier les échelles des trois axes suivants X, Y1 et Y2. L'axe Y2 peut être utilisé dans le cas où plusieurs courbes n'ont pas les mêmes unités.

Il y a deux modes d'échelles, le mode automatique et le mode manuel.

Dans le mode automatique, il faut cocher la case « Echelle X, Y1 ou Y2 Automatique » suivant l'axe concerné.

Dans ce cas, les valeurs Min et Max de l'échelle seront ajustées aux valeurs des tableaux et les champs Min et Max correspondants à l'échelle n'ont pas besoin d'être renseignés.

Si la case « Echelle automatique » n'est pas cochée. Les valeurs min et max de l'échelle seront les valeurs figurant dans les champs Min et Max de l'échelle.

### Coefficients d'affichage

Il est possible de spécifier pour chacune des trois échelles disponibles un coefficient d'affichage. Ce coefficient définit le rapport entre la valeur affichée et la valeur réelle dans le tableau.

### Option "Echelle X Automatique"

Cocher cette option pour que l'étendue de l'échelle des X s'adapte automatiquement aux valeurs des courbes. Si cette option n'est pas cochée, il est possible de donner les valeurs XMin et XMax de l'échelle des X.

### Coefficient d'affichage en X

Sa valeur par défaut est 1. Cette valeur est multipliée aux valeurs du tableau de mesure en X.

### Option "Echelle Y1 Automatique"

Cocher cette option pour que l'étendue de l'échelle des Y1 ( Echelle de gauche ) s'adapte automatiquement aux valeurs des courbes. Si cette option n'est pas cochée, il est possible de donner les valeurs Y1Min et Y1Max de l'échelle Y1.

### Coefficient d'affichage en Y

Sa valeur par défaut est 1. Cette valeur est multipliée aux valeurs des tableaux de mesure positionnés par rapport à l'axe Y1 (axe de gauche).

### Option "Echelle Y2 Automatique"

Cocher cette option pour que l'étendue de l'échelle des Y2 ( Echelle de droite ) s'adapte automatiquement aux valeurs des courbes. Si cette option n'est pas cochée, il est possible de donner les valeurs Y2Min et Y2Max de l'échelle Y2.

### Coefficient d'affichage en Y

Sa valeur par défaut est 1. Cette valeur est multipliée aux valeurs des tableaux de mesure positionnés par rapport à l'axe Y2 (axe de droite).

### Boîte de dialogue graphe - Onglet "Titre"

The screenshot shows a software dialog box titled "Objet graphe" with a blue title bar and a standard Windows-style close button. The dialog has a tabbed interface with the following tabs: "Général", "Axes et Echelles", "Titre" (which is the active tab), "Legendes", "Fond", "Neutralisation", "Seuils", "Impression", "Zoom", "3D", and "Autres". The "Titre" tab contains five text input fields arranged vertically, each with a label to its left: "Titre du graphe", "Expression à ajouter au titre", "Titre Axe X", "Titre Axe Y1", and "Titre Axe Y2". At the bottom right of the dialog, there are three buttons: "Aide", "Annuler", and "OK".

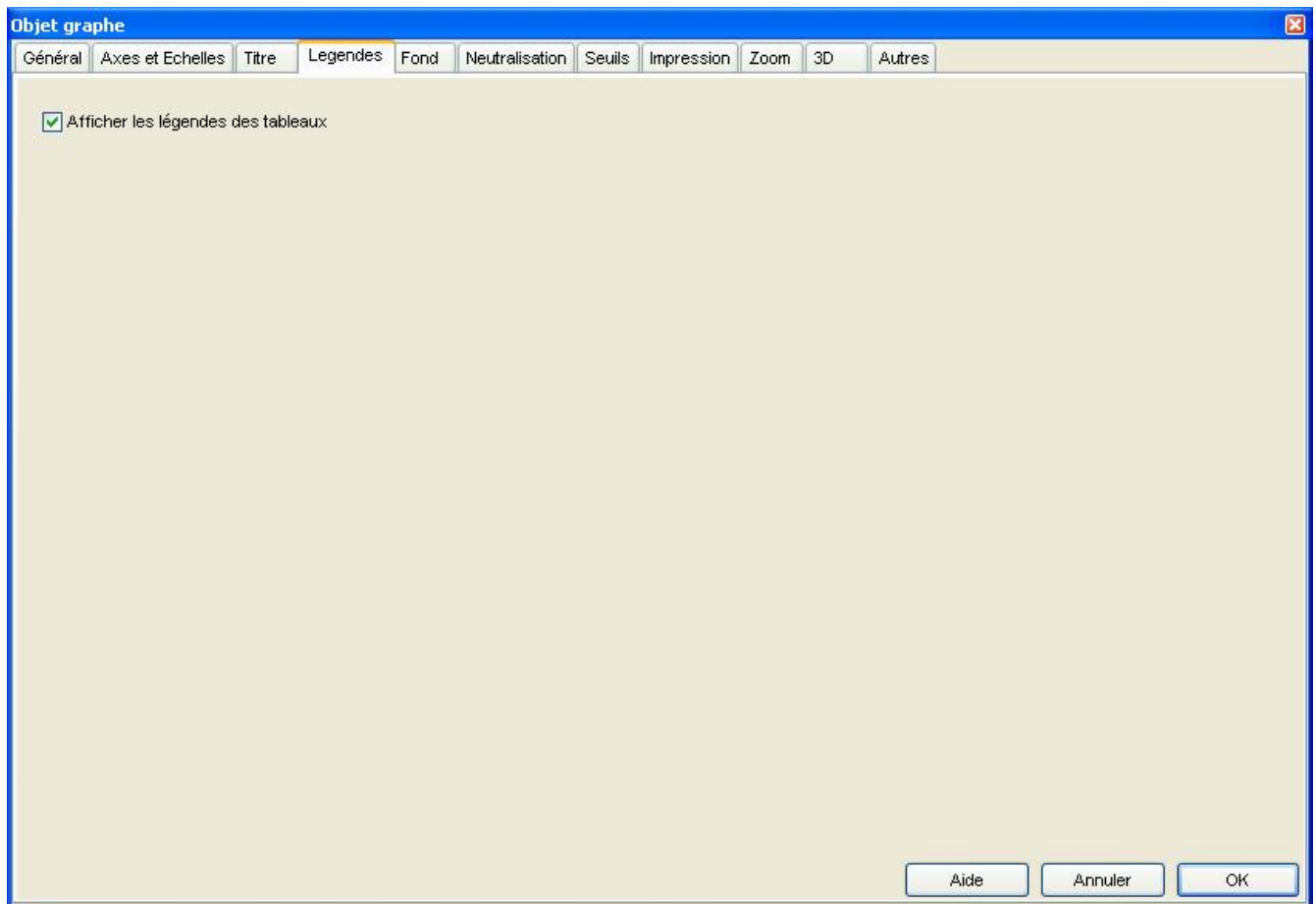
### Titres

Un titre général peut être donné à la courbe ainsi que des légendes pour chacun des trois axes disponibles.

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

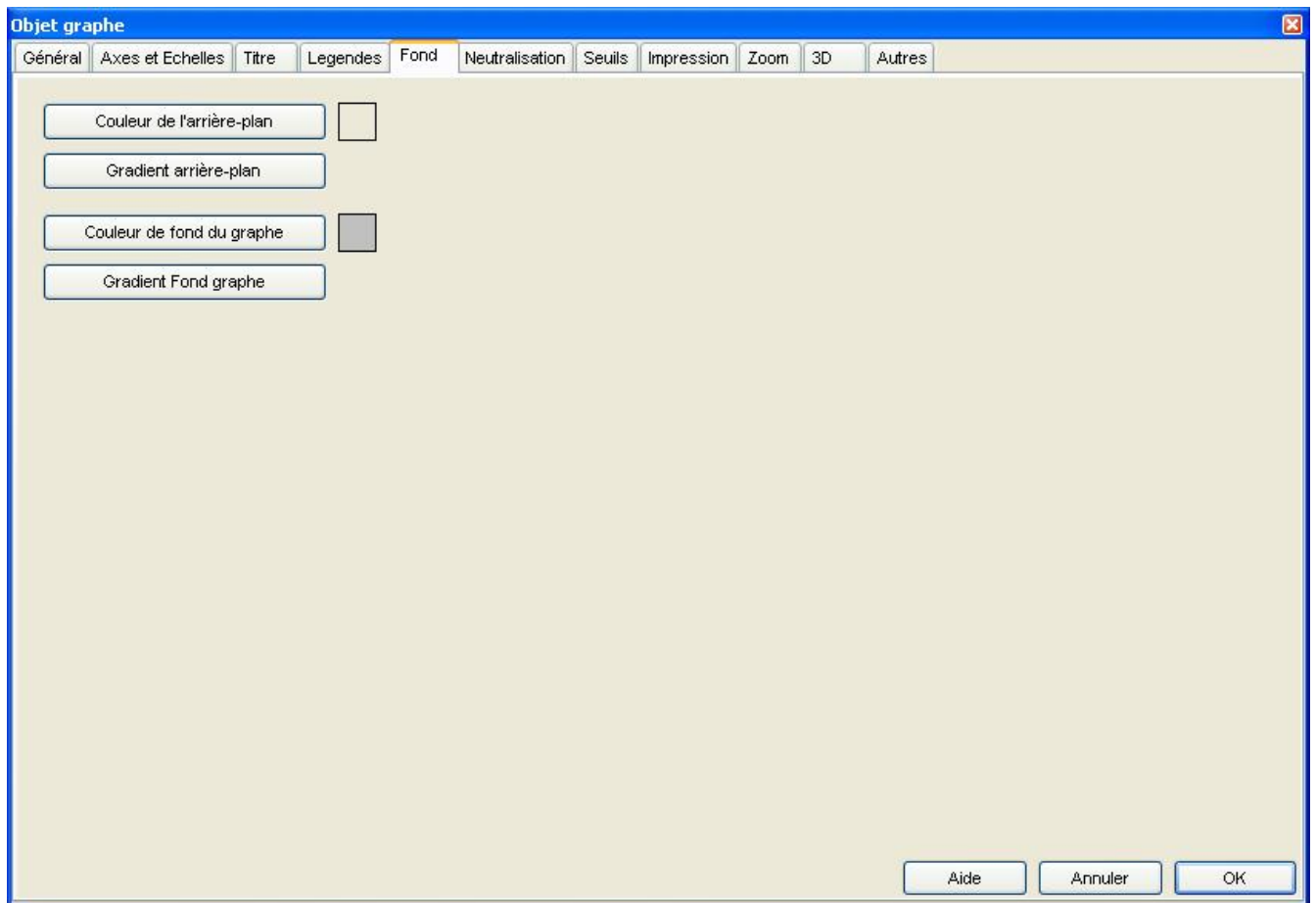
On peut également spécifier une expression à ajouter au titre. Par exemple un numéro de série contenu dans un registre : R100.

### Boîte de dialogue graphe - Onglet "Titre"



Cocher l'option "Afficher les légendes" pour afficher les noms des tableaux de mesures associés aux couleurs des courbes correspondantes.

#### **Boîte de dialogue graphe - Onglet "Fond"**

**Couleur du rectangle du graphe**

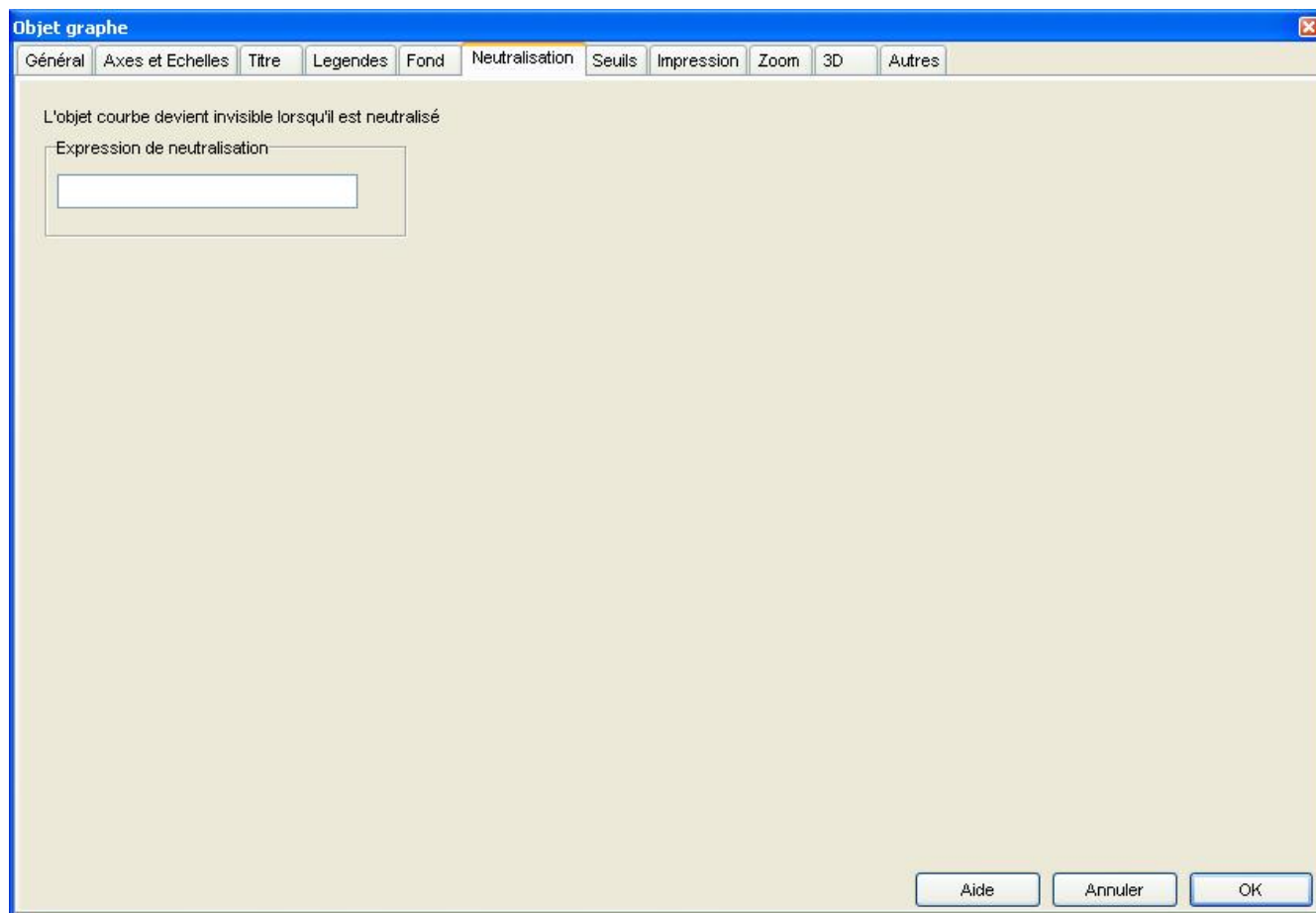
Ce bouton permet de choisir la couleur d'arrière plan du graphe.

**Couleur de l'arrière plan**

Ce bouton permet de choisir la couleur du contour de l'objet graphe.

Des boutons gradients permettent également de définir des dégradés de couleurs en fond de l'objet et en arrière plan de la courbe.

**Boîte de dialogue graphe - Onglet "Neutralisation"**



### Expression de neutralisation

Permet de spécifier la variable ou l'expression de neutralisation (Celle-ci doit obligatoirement être de type booléen). L'objet courbe sera invisible chaque fois que la variable ou l'expression sera égale à true.

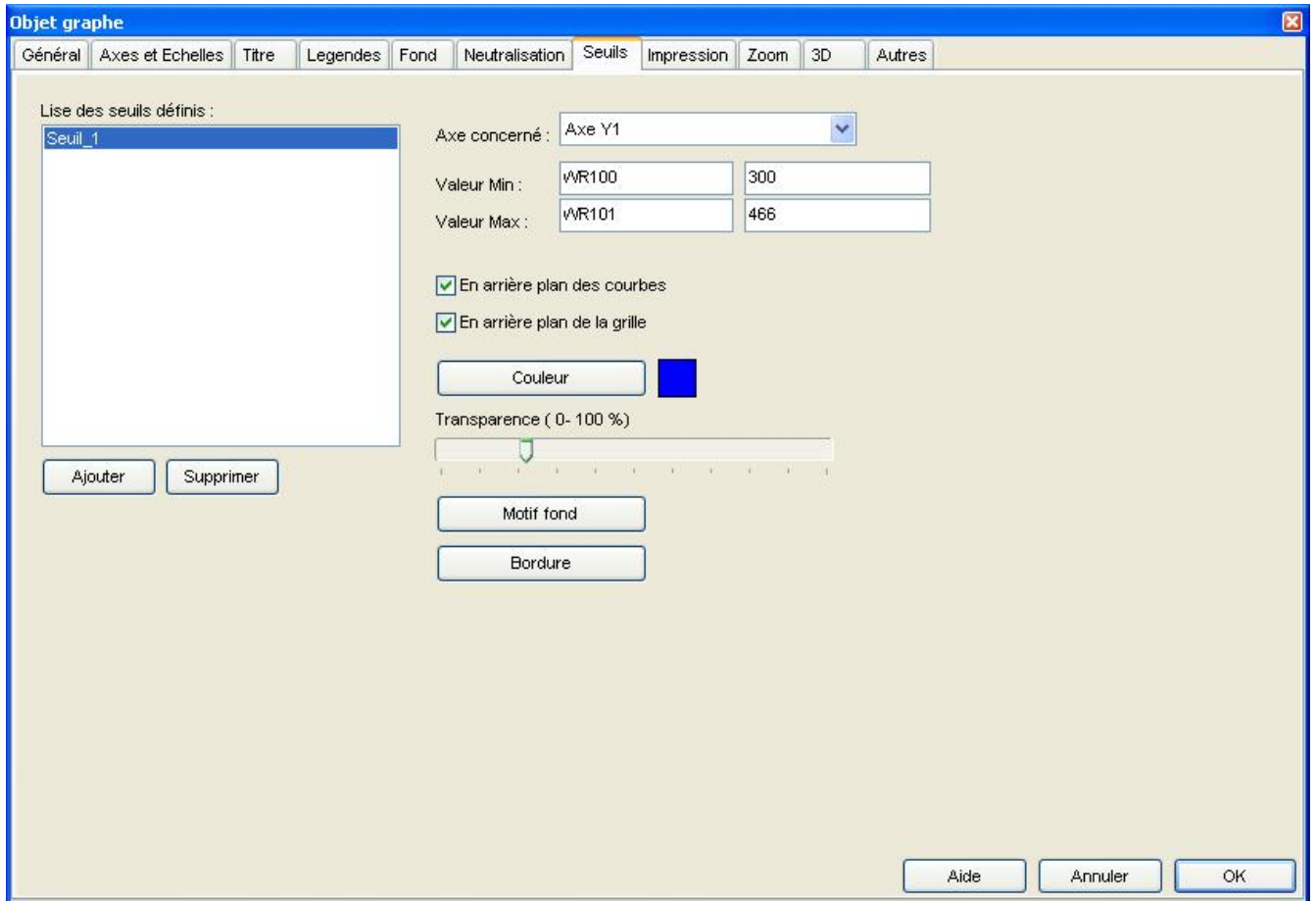
*Exemple : SA10*

Il est possible de spécifier l'adresse IP distante du noyau APIGRAF PLC dans l'expression. Si c'est le cas, c'est cette adresse qui sera prise en compte pour le rafraîchissement.

*Exemple : !193.0.1.1 !SA10*

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

### Boîte de dialogue graphe - Onglet "Seuils"



Les seuils sont des zones colorées qui s'affichent sur la courbes et qui permettent en général de visualiser des zones limites. Des seuils peuvent être ajoutés ou supprimés dans la liste déroulante à l'aide des boutons "Ajouter" et "Supprimer". Une fois un seuil ajouté, il est possible de paramétrer son aspect :

#### **Axe concerné :**

Définit l'axe auquel se rapporte le seuil, un seuil peut donc être horizontal si il se réfère à Y1 ou Y2 ou vertical si il se réfère à l'axe X.

#### **Valeur Min et Valeur Max :**

Ces valeurs définissent l'étendue de la zone seuil, on peut entrer des valeurs numériques constantes ou bien des expressions numériques.

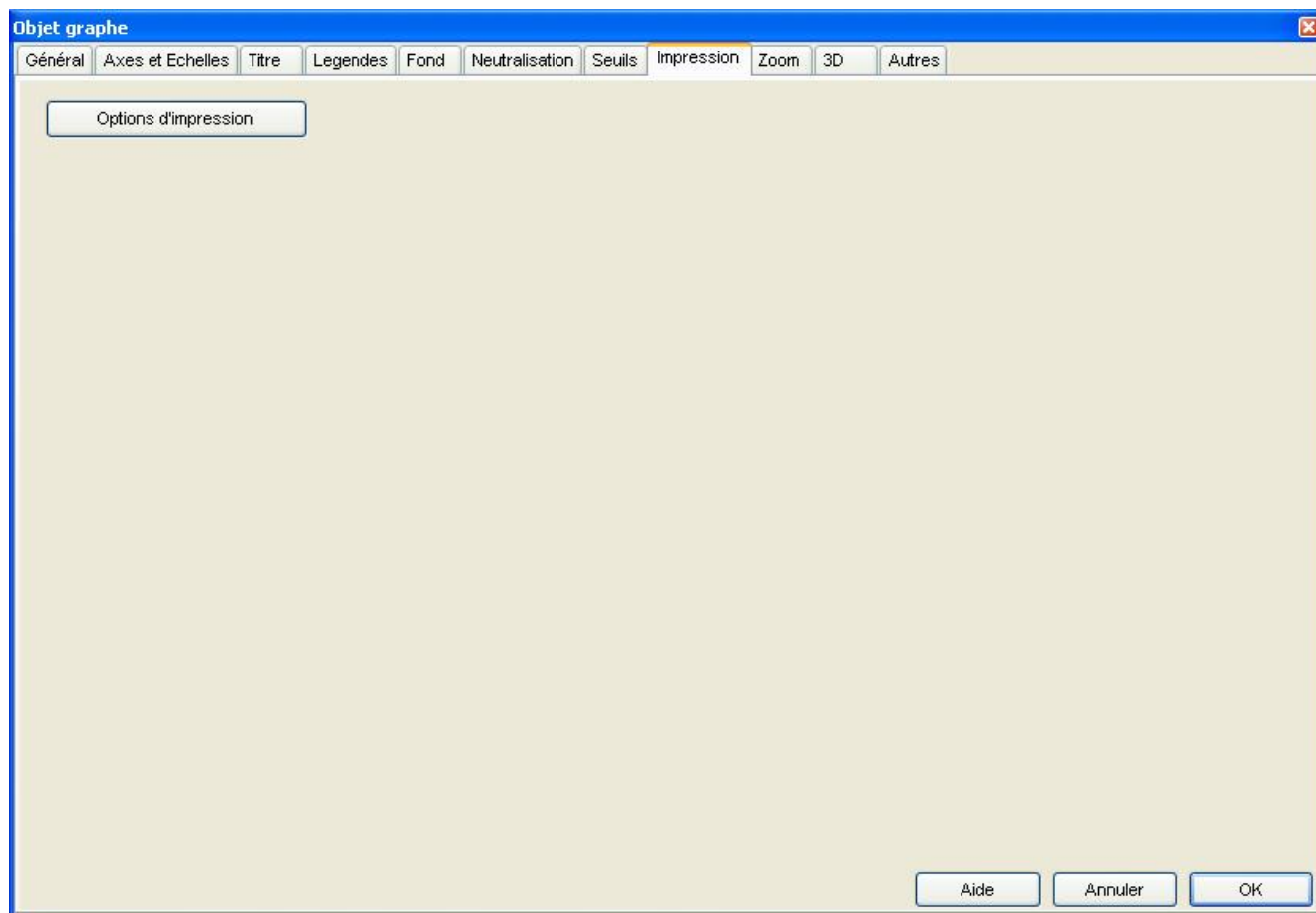
*Exemple : R10*

Il est possible de spécifier l'adresse IP distante du noyau OPTIMA PLC ou APIGRAF PLC dans l'expression. Si c'est le cas, c'est cette adresse qui sera prise en compte pour le rafraîchissement.

*Exemple : !193.0.1.1 !R10*

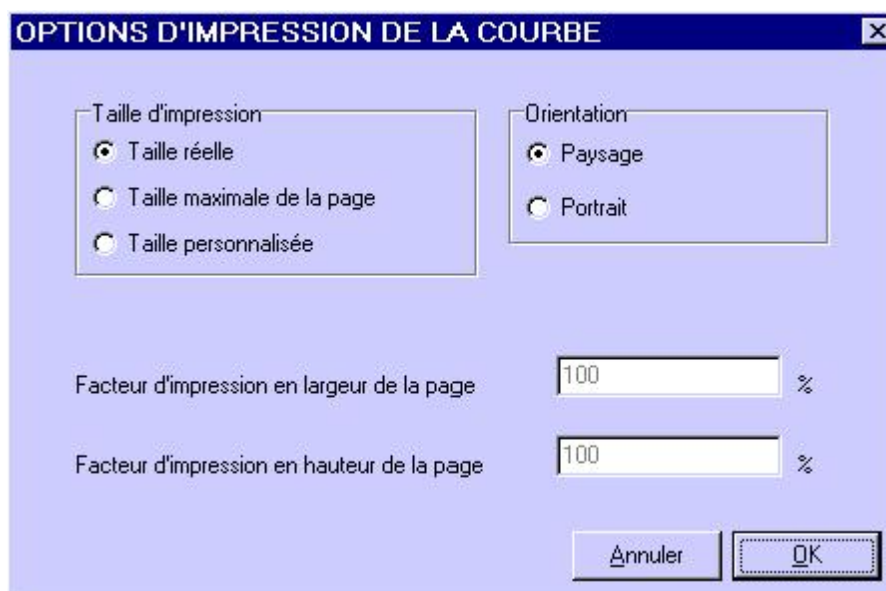
L'apparence de la zone seuil peut être également redéfinie: la transparence, le motif du fond, la couleur, ...

#### **Boîte de dialogue graphe - Onglet "Impression"**



### Options d'impression

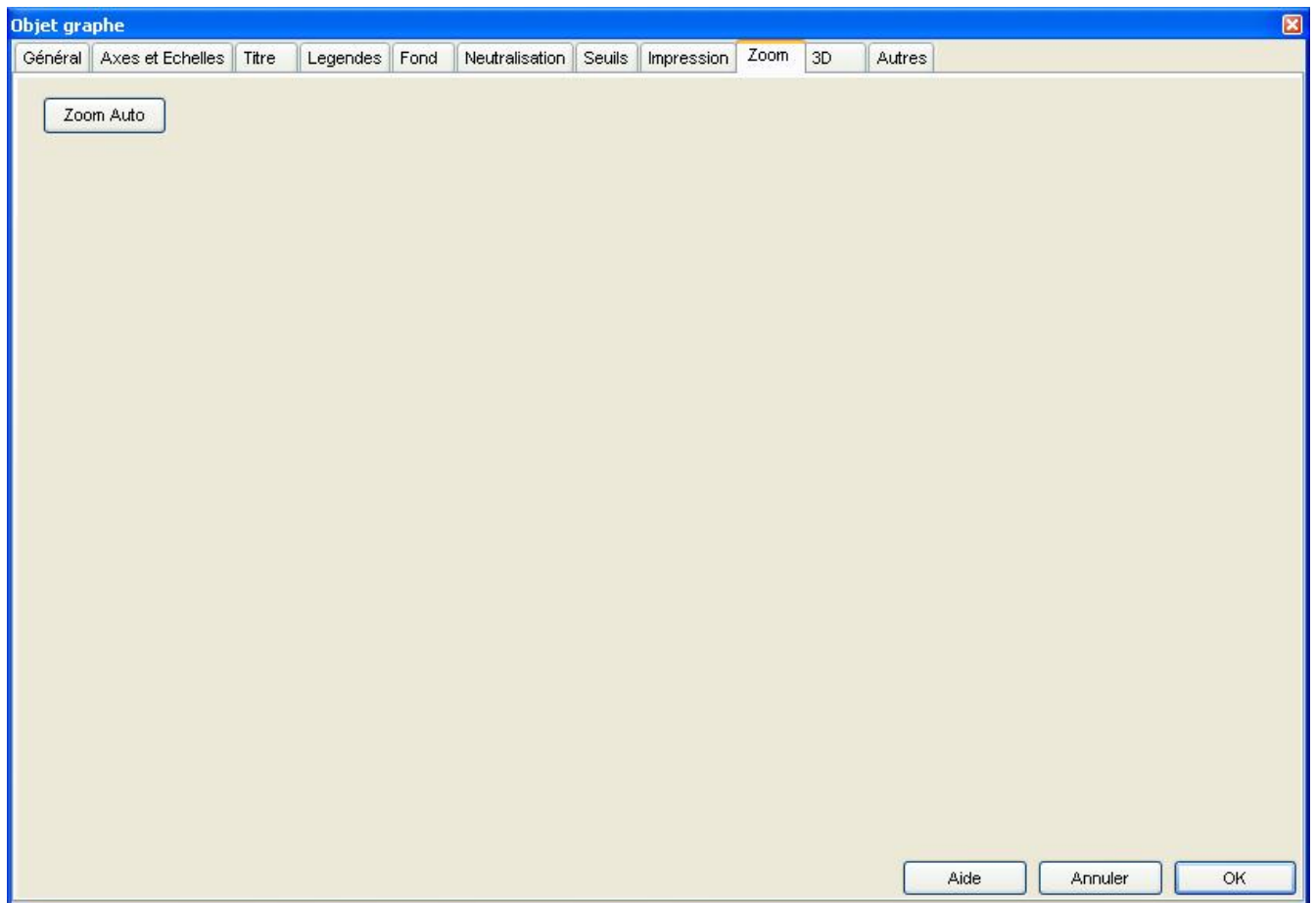
Les courbes peuvent être imprimées en mode exécution. Il faut pour cela double-cliquer sur la courbe à imprimer en mode exécution. Les options d'impression permettent de paramétrer l'impression de la courbe :



Si l'option « Taille personnalisée » est choisie, les facteurs d'impression en largeur et en hauteur de la page seront pris en compte pour l'impression. Ces facteurs seront appliqués à la taille d'affichage à l'écran de la courbe.

L'option « Taille maximale de la page » va étendre la courbe sur toute la taille du papier de sortie. Dans ce cas la définition de l'impression est moins bonne.

### Boîte de dialogue graphe - Onglet "Zoom"



Le bouton "Zoom" ouvre la fenêtre suivante :

Cette fenêtre permet d'associer des variables de contrôle des zooms de la courbe.

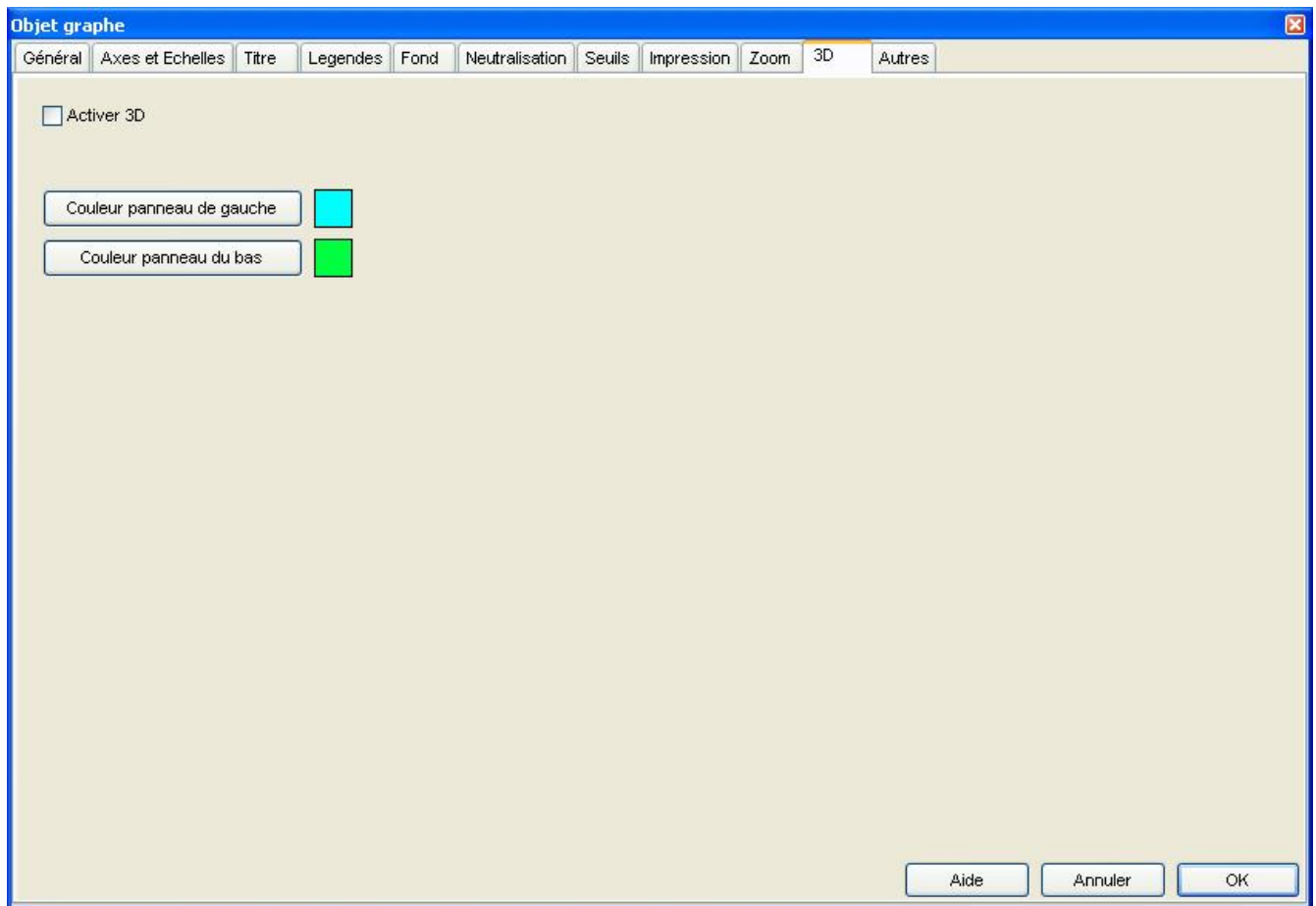
**Exemple** : Entrer SA10 dans la boîte de saisie "Expression zoom plus abscisses". Définir ensuite un bouton API monostable avec la valeur SA10. A chaque pression du bouton, la courbe va être zoomée en X avec le pas spécifié (2 par défaut).

On peut faire la même chose pour contrôler le déplacement ("Shift") de la courbe.



Une expression reset zoom permet d'annuler tous les zooms effectués sur la courbe.

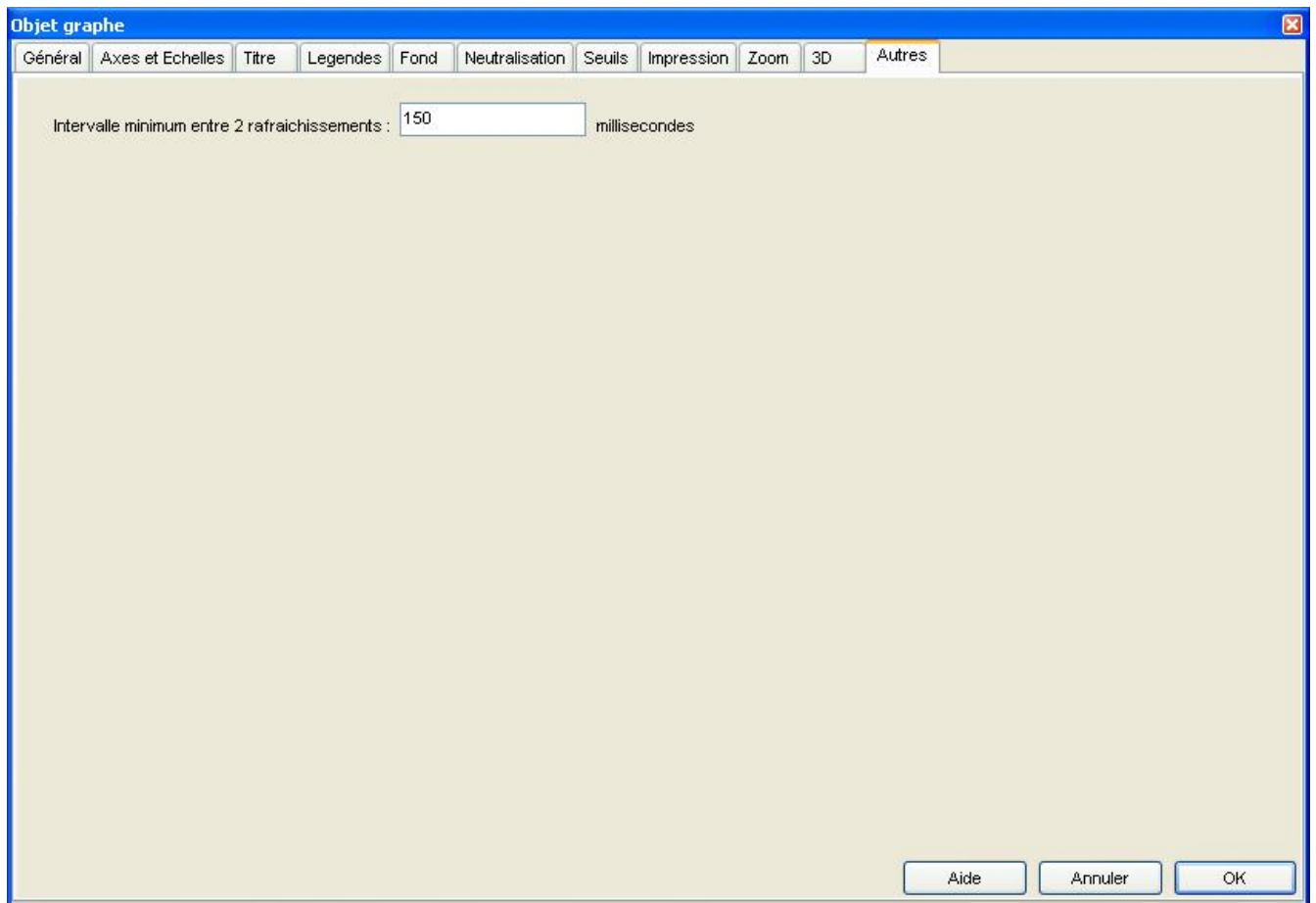
### **Boîte de dialogue graphe - Onglet "3D"**



#### **Option "Activer 3D"**

Permet d'afficher la fenêtre du graphe en 3 dimensions. On peut alors définir les couleurs des panneaux de gauche et du bas créés par la visualisation 3D.

### **Boîte de dialogue graphe - Onglet "Autres"**



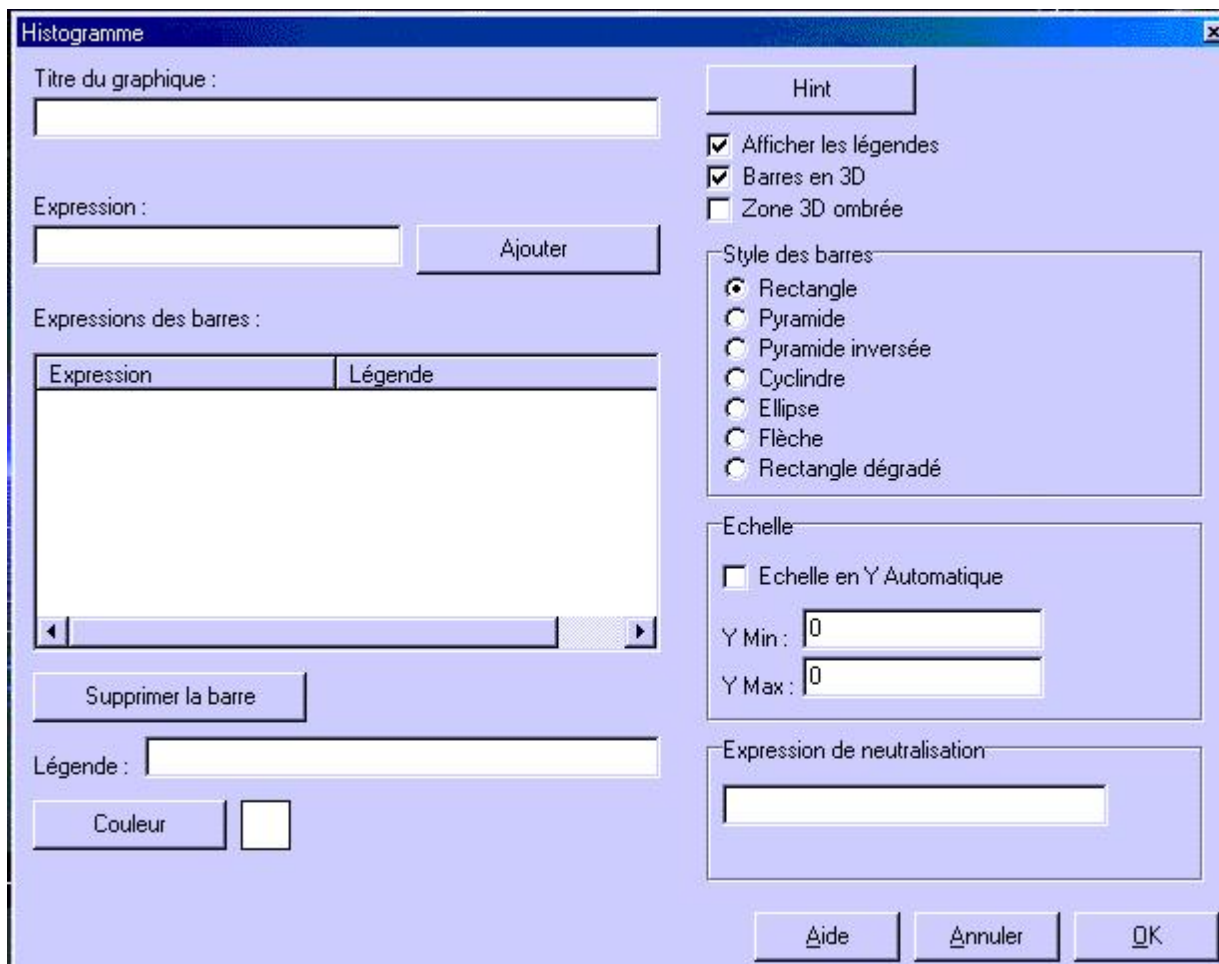
Les affichages complexes de courbes volumineuses demandent beaucoup de ressources. On peut donc limiter la fréquence de rafraichissement des affichages.

### 3.2.6.6.2 Histogramme

L'objet histogramme permet d'afficher un graphe de type histogramme sur une page de supervision. L'utilisateur doit définir la liste des variables automates qui vont composer les différentes barres de l'histogramme.

Ces variables doivent être de type numériques entiers ou décimaux ( Exemples : R100 ou RR100 )

**Boîte de dialogue « Histogramme » :**



Les champs sont les suivants :

### Titre du graphique

Permet de spécifier le titre qui apparaîtra en haut du graphique. Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

### Expression

Cette fenêtre peut contenir le nom d'une variable automate de type numérique. Cette variable peut être entière ou décimale. Une fois la saisie effectuée, il faut cliquer sur le bouton « Ajouter » pour ajouter l'expression dans la liste des barres.

Exemple : R10

Il est possible de spécifier l'adresse IP distante du noyau APIGRAF PLC dans l'expression. Si c'est le cas, c'est cette adresse qui sera prise en compte pour le rafraîchissement.

Exemple : !193.0.1.1 !R10

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

### Bouton « Ajouter »

Ajoute l'expression saisie dans la fenêtre « Expression » dans la liste des barres

### Bouton « Supprimer la barre »

Pour supprimer une barre, il faut d'abord sélectionner l'expression correspondante dans la liste et ensuite cliquer sur le bouton « Supprimer la barre »

### Fenêtre de saisie « Légende »

Permet de modifier la légende qui sera associée à une barre. Il faut d'abord sélectionner l'expression de la barre correspondante dans la liste pour modifier ensuite sa légende.

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1}) Voir Section « Appel d'une page de supervision avec passage de paramètres »)

#### Bouton « Couleur »

Permet de modifier la couleur d'une barre. Il faut d'abord sélectionner la barre dans la liste pour modifier ensuite sa couleur.

#### Bouton « Hint »

Ce bouton permet de définir les paramètres de « Hint » de l'objet. Le texte « Hint » que l'on définit est le texte qui apparaît sur fond jaune lorsque l'on passe le curseur de la souris sur l'objet.

Quand on clique sur le bouton « Hint », le fenêtre suivante s'ouvre :



Cette fenêtre présente deux contrôles :

#### Option « Afficher le Hint »

Si cette option n'est pas cochée, le texte du Hint n'apparaîtra pas quand le curseur de la souris passera sur l'objet même si celui-ci est défini. Il faut donc cocher cette option si l'on veut utiliser le « Hint » de l'objet.

#### Fenêtre de saisie « Hint »

Cette fenêtre permet d'entrer le texte du Hint. Ce texte peut être un paramètre (exemple {1}).

#### Option « Afficher les légendes »

Si cette option n'est pas cochée, le cadre contenant la liste des légendes ne sera pas affichée et l'histogramme occupera de ce fait une place plus importante dans l'objet.

#### Option « Barres en 3D »

Si cette option est cochée, les barres composant l'histogramme seront dessinées en 3 dimensions.

#### Option « Zone 3D Ombrée »

Si cette option est cochée, la partie 3D des barres apparaîtra plus foncée que le secteur.

#### Style des barres

Permet de choisir l'apparence des barres formant l'histogramme (Rectangle, flèche, ...)

#### Echelle en Y Automatique

Si cette option est cochée, les valeurs min et max de l'échelle verticale seront calculées automatiquement.

#### Ymin

Si l'option « Echelle en Y Automatique » n'est pas cochée, cette valeur fixera la valeur min de l'échelle verticale.

#### YMax

Si l'option « Echelle en Y Automatique » n'est pas cochée, cette valeur fixera la valeur max de l'échelle verticale.

#### Expression de neutralisation

Permet de spécifier la variable ou l'expression de neutralisation (Celle-ci doit obligatoirement être de type booléen). L'objet histogramme sera invisible chaque fois que la variable ou l'expression sera égale à true.

*Exemple : SA10*

Il est possible de spécifier l'adresse IP distante du noyau APIGRAF PLC dans l'expression. Si c'est le cas, c'est cette adresse qui sera prise en compte pour le rafraîchissement.

Exemple : !193.0.1.1 !SA10

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

### 3.2.6.6.3 Echelle

Cet objet permet d'afficher une échelle graduée sur une page de supervision, cet objet peut être utile pour ajouter une graduation sous un objet Jauge, curseur etc..

#### Exemple d'objet échelle :



#### Boîte de dialogue « Echelle » :

Les paramètres sont les suivants :

#### Valeur minimale

Ce paramètre indique la valeur initiale de graduation de l'échelle. Ce paramètre peut être entier ou décimal.

#### Valeur maximale

Indique la valeur finale de graduation de l'échelle. Ce paramètre peut être entier ou décimal.

#### Intervalle

Ce paramètre indique la valeur séparant deux graduations. Ce paramètre peut être entier ou décimal.

#### Nombre de décimales

Les graduations peuvent être affichées avec des décimales. Ce paramètre indique le nombre de décimales affichées. Ce paramètre est de type entier.

**Préfixe**

Permet de définir un texte destiné à être ajouté à gauche des valeurs numériques de graduation.

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

**Suffixe**

Permet de définir un texte destiné à être ajouté à droite des valeurs numériques de graduation (par exemple, le suffixe peut être utilisé pour afficher un % à droite de chaque valeur).

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

**Bouton « Couleur de l'échelle »**

Ce contrôle permet de modifier la couleur de l'échelle.

**Bouton « Caractères »**

Permet de définir la police de caractères utilisée pour afficher les valeurs numériques des graduations.

**Orientation**

Permet d'orienter la graduation de façon verticale ou horizontale.

**Position**

Si la graduation est horizontale, ce contrôle permet de placer les valeurs numériques de graduation en haut ou en bas de l'échelle.

Si la graduation est verticale, ce contrôle permet de placer les valeurs numériques de graduation à gauche ou à droite de l'échelle.

**Expression de neutralisation**

Permet de spécifier la variable ou l'expression de neutralisation (Celle-ci doit obligatoirement être de type booléen). L'objet échelle sera invisible chaque fois que la variable ou l'expression sera égale à true.

*Exemple : SA10*

Il est possible de spécifier l'adresse IP distante du noyau APIGRAF PLC dans l'expression. Si c'est le cas, c'est cette adresse qui sera prise en compte pour le rafraîchissement.

*Exemple : !193.0.1.1 !SA10*

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

**Bouton « Hint »**

Ce bouton permet de définir les paramètres de « Hint » de l'objet. Le texte « Hint » que l'on définit est le texte qui apparaît sur fond jaune lorsque l'on passe le curseur de la souris sur l'objet.

Quand on clique sur le bouton « Hint », la fenêtre suivante s'ouvre :



Cette fenêtre présente deux contrôles :

**Option « Afficher le Hint »**

Si cette option n'est pas cochée, le texte du Hint n'apparaîtra pas quand le curseur de la souris passera sur l'objet même si celui-ci est défini. Il faut donc cocher cette option si l'on veut utiliser le « Hint » de l'objet.

**Fenêtre de saisie « Hint »**

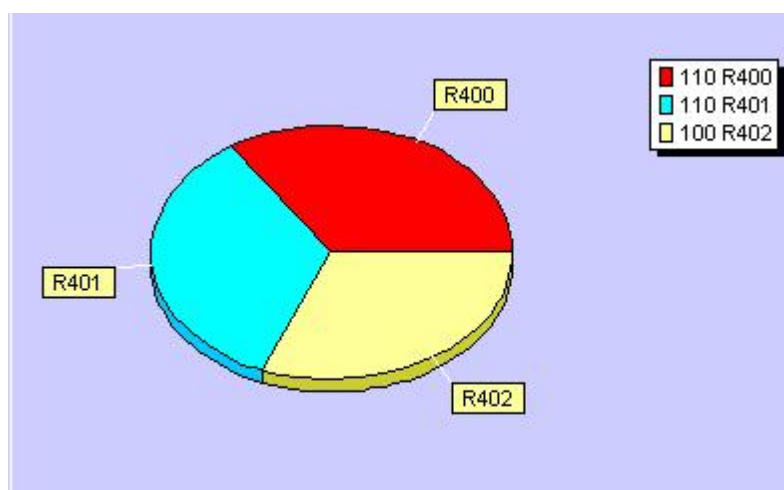
Cette fenêtre permet d'entrer le texte du Hint. Ce texte peut être un paramètre (exemple {1}).

**3.2.6.6.4 Camembert**

L'objet camembert permet d'afficher un graphe de type camembert sur une page de supervision. L'utilisateur doit définir la liste des variables automatiques qui vont composer les différents secteurs du camembert.

Ces variables doivent être de type numériques entiers ou décimaux ( Exemples : R100 ou RR100 )

Exemple de camembert :

**Boîte de dialogue « Camembert » :**

**Secteurs**

Titre du graphique :

Expression du secteur

Expressions des secteurs :

Expression	Légende
R400	R400
R401	R401
R402	R402

Légende :

☒ Afficher les légendes

☒ Zone3D ombrée

☒ Cercle parfait

Expression de neutralisation

Les champs sont les suivants :

**Titre du graphique**

Permet de spécifier le titre qui apparaîtra en haut du graphique. Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

**Expression du secteur**

Cette fenêtre peut contenir le nom d'une variable automate de type numérique. Cette variable peut être entière ou décimale. Une fois la saisie effectuée, il faut cliquer sur le bouton « Ajouter » pour ajouter l'expression dans la liste des secteurs.

*Exemple* : R10

Il est possible de spécifier l'adresse IP distante du noyau APIGRAF PLC dans l'expression. Si c'est le cas, c'est cette adresse qui sera prise en compte pour le rafraîchissement.

*Exemple* : !193.0.1.1 !R10

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

**Bouton « Ajouter »**

Ajoute l'expression saisie dans la fenêtre « Expression du secteur » dans la liste des secteurs.

**Bouton « Supprimer »**

Pour supprimer un secteur, il faut d'abord le sélectionner dans la liste et ensuite cliquer sur le bouton « Supprimer »

**Fenêtre de saisie « Légende »**

Permet de modifier la légende qui sera associé à un secteur. Il faut d'abord sélectionner le secteur dans la liste pour modifier ensuite sa légende.

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

**Bouton « Couleur »**

Permet de modifier la couleur d'un secteur. Il faut d'abord sélectionner le secteur dans la liste pour modifier ensuite sa couleur.

**Bouton « Hint »**

Ce bouton permet de définir les paramètres de « Hint » de l'objet. Le texte « Hint » que l'on définit est le texte qui apparaît sur fond jaune lorsque l'on passe le curseur de la souris sur l'objet.

Quand on clique sur le bouton « Hint », le fenêtre suivante s'ouvre :



Cette fenêtre présente deux contrôles :

**Option « Afficher le Hint »**

Si cette option n'est pas cochée, le texte du Hint n'apparaîtra pas quand le curseur de la souris passera sur l'objet même si celui-ci est défini. Il faut donc cocher cette option si l'on veut utiliser le « Hint » de l'objet.

**Fenêtre de saisie « Hint »**

Cette fenêtre permet d'entrer le texte du Hint. Ce texte peut être un paramètre (exemple {1}).



**Option « Afficher les légendes »**

Si cette option n'est pas cochée, le cadre contenant la liste des légendes ne sera pas affichée et le camembert occupera de ce fait une place plus importante dans l'objet.

**Option « Zone 3D Ombrée »**

Si cette option est cochée, la partie 3D des secteur (tranche du secteur) apparaîtra plus foncée que le secteur.

**Option « Cercle parfait »**

Si cette option est cochée, le camembert formera un cercle parfait. Sinon la forme du camembert sera elliptique.

**Expression de neutralisation**

Permet de spécifier la variable ou l'expression de neutralisation (Celle-ci doit obligatoirement être de type booléen). L'objet camembert sera invisible chaque fois que la variable ou l'expression sera égale à true.

*Exemple : SA10*

Il est possible de spécifier l'adresse IP distante du noyau APIGRAF PLC dans l'expression. Si c'est le cas, c'est cette adresse qui sera prise en compte pour le rafraîchissement.

*Exemple : !193.0.1.1 !SA10*

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

### 3.2.6.7 Objets de l'onglet ANIMATIONS / IMAGES

#### 3.2.6.7.1 Animation

Cet objet est généralement employé pour réaliser une animation, des boutons à états multiples, ou encore afficher des messages contextuels...L'apparence de cet objet varie en fonction de la valeur de sa variable de contrôle.

Cette variable peut être de type booléen (flag L, Interrupteur logique SA...) ou entier (Registre R...).

**Boîte de dialogue Objet Animation**

Objet Animation

Expression

Neutralisation

Expression de neutralisation :

Cet objet est invisible quand il est neutralisé

☐ Incrémentation Auto

Valeur Début 0

Valeur Fin 0

Hint

Etats

Ajouter

Monter

Descendre

Parametrer

Supprimer

Aide Annuler OK

Les paramètres sont les suivants :

### Expression

Permet de définir le nom de la variable de contrôle de l'objet (variable associée à l'objet animation). Cette variable sert d'index pour chaque état de l'animation (un état correspond à une valeur de la variable).

Indiquer dans ce champ la variable automate à utiliser (de type numérique entier)

*Exemple : R20*

Il est possible de spécifier l'adresse IP distante du noyau APIGRAF PLC dans l'expression. Si c'est le cas, c'est cette adresse qui sera prise en compte pour le rafraîchissement.

*Exemple : !193.0.1.1 !R20*

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

### Valeur Début et Valeur fin

Ces champs permettent de définir la plage de valeurs de la variable de contrôle, lesquelles définiront les différents aspects de l'objet.

Par exemple, si la variable de contrôle est de type booléen la valeur de début sera 0 et celle de fin sera 1, nous obtiendrons donc un objet à deux états comme, par exemple, un bouton poussoir bistable.

### Incrémentation Auto

Si la case « Incrémentation » est cochée la variable est automatiquement incrémentée lors d'un clic de souris sur l'objet animation. Lorsque la valeur de fin est dépassée, la valeur de début est remplacée dans la variable de contrôle.

### Etats de la variable

Cette zone permet de visualiser la liste des différents états de la variable à tester et l'affichage correspondant. Cette liste est affichée avec la valeur de début et la valeur de fin pour chaque état.

### Ajouter, Supprimer, Paramétrer

Le bouton « Ajouter » permet de créer un nouvel état. Le bouton « Supprimer » supprime l'état en surbrillance dans la liste. Enfin, le bouton « Paramétrer » permet de modifier l'état en surbrillance (un double clic sur l'un des états provoque le même effet), il permet l'appel à la boîte de dialogue « Etat d'animation » (cf. ci-dessous).

### Monter, Descendre

Ces boutons permettent de décaler la barre en surbrillance dans la liste : vers le haut avec « monter » et vers le bas avec « descendre ».

### Expression de neutralisation

Permet de spécifier la variable ou l'expression de neutralisation (Celle-ci doit obligatoirement être de type booléen). L'objet animation sera invisible chaque fois que la variable ou l'expression sera égale à true.

*Exemple : SA10*

Il est possible de spécifier l'adresse IP distante du noyau APIGRAF PLC dans l'expression. Si c'est le cas, c'est cette adresse qui sera prise en compte pour le rafraîchissement.

*Exemple : !193.0.1.1 !SA10*

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

### Bouton « Hint »

Ce bouton permet de définir les paramètres de « Hint » de l'objet. Le texte « Hint » que l'on définit est le texte qui apparaît sur fond jaune lorsque l'on passe le curseur de la souris sur l'objet.

Quand on clique sur le bouton « Hint », le fenêtre suivante s'ouvre :



Cette fenêtre présente deux contrôles :

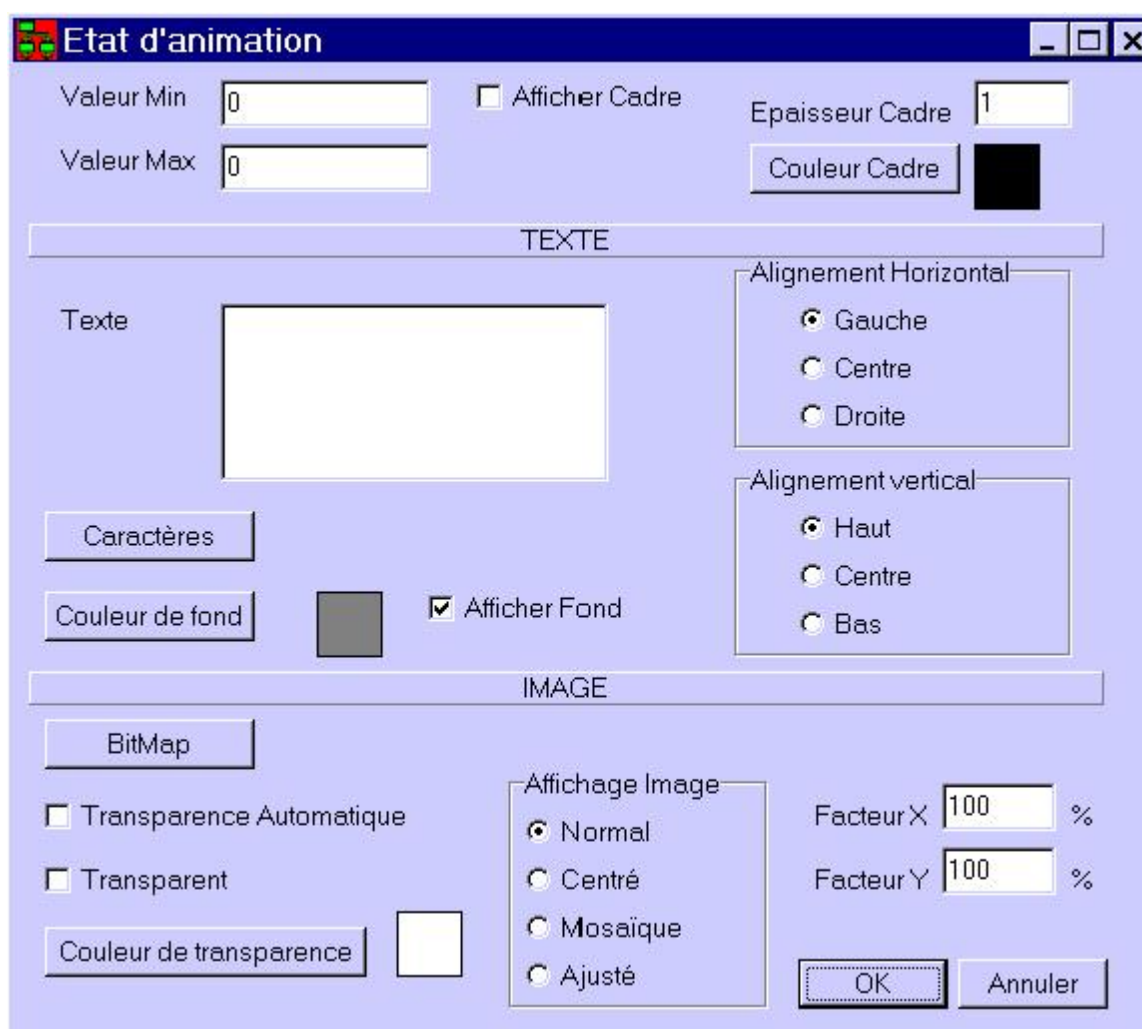
#### Option « Afficher le Hint »

Si cette option n'est pas cochée, le texte du Hint n'apparaîtra pas quand le curseur de la souris passera sur l'objet même si celui-ci est défini. Il faut donc cocher cette option si l'on veut utiliser le « Hint » de l'objet.

#### Fenêtre de saisie « Hint »

Cette fenêtre permet d'entrer le texte du Hint. Ce texte peut être un paramètre (exemple {1}).

#### Boîte de dialogue Etat d'animation



Cette boîte de dialogue permet de configurer un affichage donné pour une valeur de la variable associée. Les champs suivants doivent être définis :

#### Valeur min et valeur max

Ces deux champs permettent de définir la première et la dernière valeur de la variable pour lesquelles l'affichage sera réalisé avec les paramètres qui suivent.

Autrement dit, ces valeurs définissent l'intervalle de valeurs, bornes comprises, correspondant à l'état de l'objet décrit dans cette boîte.

**Afficher cadre**

Cette option permet d'afficher un cadre autour de l'image de l'objet.

**Couleur cadre**

Permet de choisir la couleur du cadre à l'aide de la boîte de dialogue couleur.

**Epaisseur cadre**

Cette option permet de choisir l'épaisseur du cadre de l'image. La valeur par défaut est 1 pixel.

**Texte**

Saisir dans la zone correspondante le texte à afficher.

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

**Groupes Alignement vertical et Alignement horizontal**

Permet de définir la manière dont le texte doit figurer dans l'objet.

**Caractères**

Permet l'accès à la boîte de dialogue concernant le type d'affichage du texte (couleur, police...).

**Couleur de fond**

Sélection de la couleur de fond de l'objet au moyen d'une boîte de dialogue « couleur ».

**Afficher fond**

Si cette option n'est pas cochée, l'objet est transparent, sinon le fond prend la couleur définie plus haut.

**Bitmap**

Permet l'importation d'une image de format bitmap (BMP). Pour importer l'image il suffit d'indiquer le chemin d'accès au fichier. Le bouton « parcourir » facilite la recherche en affichant la liste des fichiers BMP.

L'affichage de l'image peut également être

- Normal : l'image est affichée en haut à gauche de l'objet,
- Centré : l'image est affichée au centre de l'objet,
- Mosaïque : l'image est répétée de manière à couvrir toute la surface de l'objet.
- Ajusté : permet d'ajuster la taille de l'image à celle de l'objet.

**Transparence Automatique**

Si cette option est cochée, la couleur de transparence est déterminée automatiquement. C'est en principe la couleur de fond de l'image qui est prise en compte.

**Transparent**

Permet de définir si l'on veut que l'image ait une couleur de transparence. Si tel est le cas, la couleur de l'image définie comme couleur de transparence ne sera pas affichée. L'image laisse alors apparaître ce qui se trouve en arrière plan au niveau de la couleur de transparence.

**Couleur de transparence**

Cette option permet dans le cas où l'option « Couleur de transparence Automatique » n'est pas cochée de choisir la couleur de transparence de l'image.

**Facteurs X et Y**

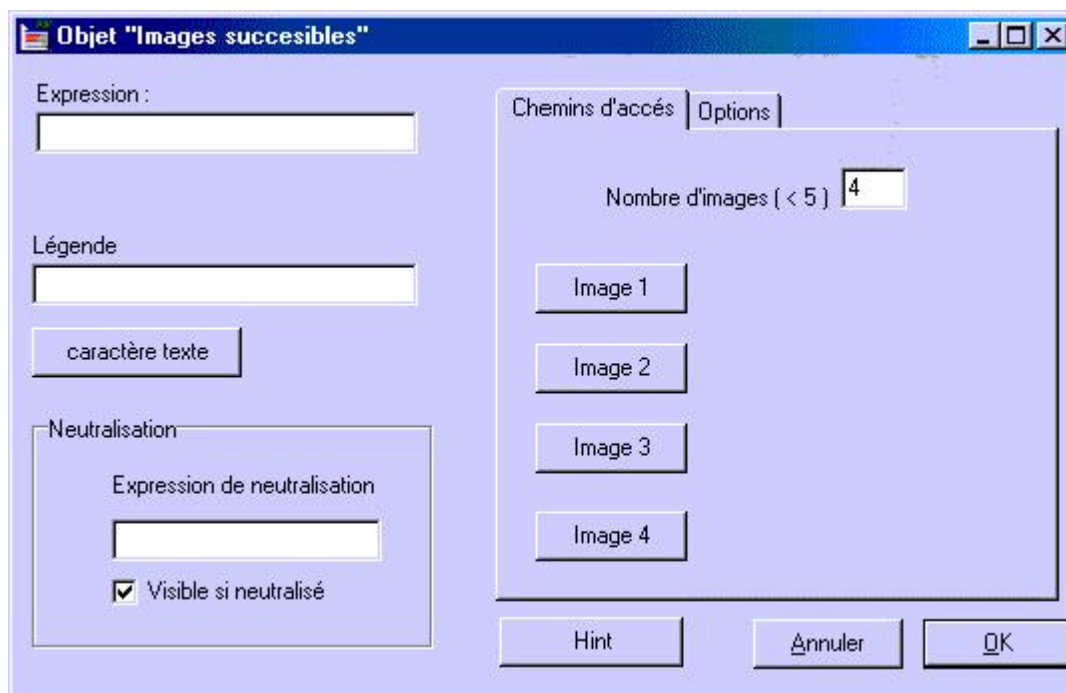
Permet de modifier la taille de l'image suivant deux échelles (verticale et horizontale) pouvant varier de 0 à 1000%.

### 3.2.6.7.2 Insertion images succesives

Cet objet est généralement employé pour réaliser une animation, des boutons à états multiples, ou encore afficher des messages contextuels...

L'apparence de cet objet varie avec un clic sur le bouton gauche de la souris au-dessus de l'objet ou en fonction de la valeur de sa variable de contrôle. Cette variable peut être de type booléen (flag L, Interrupteur logique SA...) ou entier (Registre R...).

**Boîte de dialogue « Insertion images succesives » :**



Les paramètres sont les suivants :

### Expression

Permet de définir le nom de la variable de contrôle de l'objet (variable associée à l'objet insertion images successives). Cette variable sert d'index pour chaque état de l'animation (un état correspond à une valeur de la variable). Indiquer dans ce champ la variable automate à utiliser (de type numérique entier)

*Exemple : R20*

Il est possible de spécifier l'adresse IP distante du noyau APIGRAF PLC dans l'expression. Si c'est le cas, c'est cette adresse qui sera prise en compte pour le rafraîchissement.

*Exemple : !193.0.1.1 !R20*

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

### Légende

Permet de mettre une légende au-dessus de l'objet. Ce paramètre n'accepte que du texte.

### Bouton « Caractères texte »

Permet l'accès à la boîte de dialogue concernant le type d'affichage de la légende (couleur, police...).

### Expression de neutralisation

Permet de spécifier la variable ou l'expression de neutralisation (Celle-ci doit obligatoirement être de type booléen).

*Exemple : SA10*

Il est possible de spécifier l'adresse IP distante du noyau APIGRAF PLC dans l'expression. Si c'est le cas, c'est cette adresse qui sera prise en compte pour le rafraîchissement.

*Exemple : !193.0.1.1 !SA10*

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

### Option « visible si neutralisé »

Si cette option n'est pas cochée, l'objet deviendra invisible lorsqu'il sera neutralisé.

### Bouton « Hint »

Ce bouton permet de définir les paramètres de « Hint » de l'objet. Le texte « Hint » que l'on définit est le texte qui apparaît sur fond jaune lorsque l'on passe le curseur de la souris sur l'objet.

Quand on clique sur le bouton « Hint », la fenêtre suivante s'ouvre :



Cette fenêtre présente deux contrôles :

**Option « Afficher le Hint »**

Si cette option n'est pas cochée, le texte du Hint n'apparaîtra pas quand le curseur de la souris passera sur l'objet même si celui-ci est défini. Il faut donc cocher cette option si l'on veut utiliser le « Hint » de l'objet.

**Fenêtre de saisie « Hint »**

Cette fenêtre permet d'entrer le texte du Hint. Ce texte peut être un paramètre (exemple {1}).

**Onglet « Chemins d'accès »**

Les paramètres sont les suivants :

**Nombre d'images**

Permet de spécifier le nombre d'image qui se succéderont durant l'animation. Ce paramètre doit être un chiffres compris entre 1 et 4.

**Image 1, Image 2, Image 3, Image 4**

Permet de spécifier le fichier correspondant à l'image. Les formats suivants sont admis : \*.BMP ; \*.EMF ; \*.WMF ; \*.ICO

Remarque : Une collection d'images au format BITMAP est disponible dans le sous-répertoire APIGRAFIP \GRAPHIQUES\BITMAPS

**Onglet « Options »**



Les paramètres sont les suivants :

#### **Option Centrer l'image**

Cette option permet de centrer l'image dans l'objet.

#### **Option Ajuster la taille de l'objet à l'image**

Cette option ajuste automatiquement la taille de l'objet crée à la taille réelle de l'image.

#### **Option Ajuster la taille de l'image à l'objet**

Cette option ajuste automatiquement la taille de l'image à la taille réelle de l'objet. Le fait de redimensionner l'objet aura donc une influence sur la taille de l'image.

#### **Transparent**

Permet de définir si l'on veut que l'image ait une couleur de transparence. Si tel est le cas, la couleur de l'image définie comme couleur de transparence ne sera pas affichée. L'image laisse alors apparaître ce qui se trouve en arrière plan au niveau de la couleur de transparence.

#### **Couleur de transparence Automatique**

Si cette option est cochée, la couleur de transparence est déterminée automatiquement. C'est en principe la couleur de fond de l'image qui est prise en compte.

#### **Couleur de transparence**

Cette option permet dans le cas ou l'option « Couleur de transparence Automatique » n'est pas cochée de choisir la couleur de transparence de l'image.

### **3.2.6.7.3 Texte défilant**

Cet objet permet d'afficher un texte qui défile de façon continue. Deux aspects sont possibles pour cet objet :

- l'aspect texte normal
- l'aspect Panneau à leds

#### **Boîte de dialogue texte défilant**

Les paramètres sont :

« **Texte** »

Définit le texte à faire défiler.

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

« **Bouton caractères** »

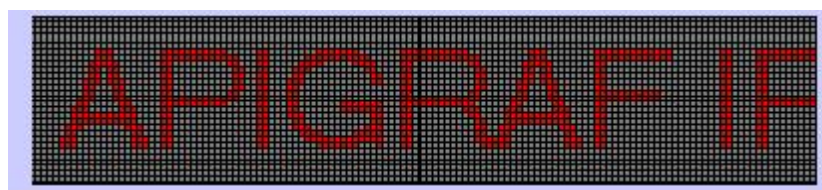
Ce bouton permet de définir la police du texte.

« **Type d'affichage** »

2 types d'affichage sont disponibles :

- normal
- panneau à leds

Exemple d'affichage avec le type panneau à leds :



« **Bouton Couleur des leds off** »

Définit la couleur des leds qui sont éteintes.

« **Bouton Couleur des leds on** »

Définit la couleur des leds allumées.

« **Bouton Couleur Fond** »

Définit la couleur de fond de la zone de défilement de texte.

**Expression de neutralisation**

Permet de spécifier la variable ou l'expression de neutralisation (Celle-ci doit obligatoirement être de type booléen). L'objet Texte défilant sera invisible chaque fois que la variable ou l'expression sera égale à true.



Exemple : SA10

Il est possible de spécifier l'adresse IP distante du noyau APIGRAF PLC dans l'expression. Si c'est le cas, c'est cette adresse qui sera prise en compte pour le rafraîchissement.

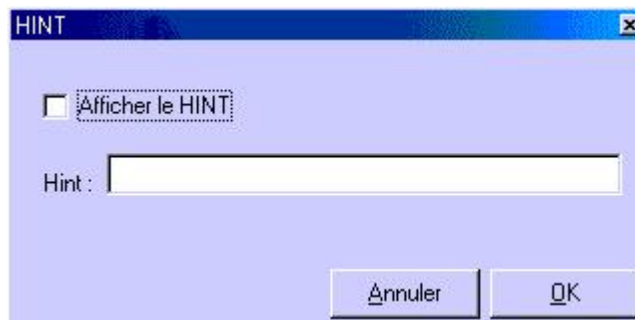
Exemple : !193.0.1.1 !SA10

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

#### Bouton « Hint »

Ce bouton permet de définir les paramètres de « Hint » de l'objet. Le texte « Hint » que l'on définit est le texte qui apparaît sur fond jaune lorsque l'on passe le curseur de la souris sur l'objet.

Quand on clique sur le bouton « Hint », la fenêtre suivante s'ouvre :



Cette fenêtre présente deux contrôles :

#### Option « Afficher le Hint »

Si cette option n'est pas cochée, le texte du Hint n'apparaîtra pas quand le curseur de la souris passera sur l'objet même si celui-ci est défini. Il faut donc cocher cette option si l'on veut utiliser le « Hint » de l'objet.

#### Fenêtre de saisie « Hint »

Cette fenêtre permet d'entrer le texte du Hint. Ce texte peut être un paramètre (exemple {1}).

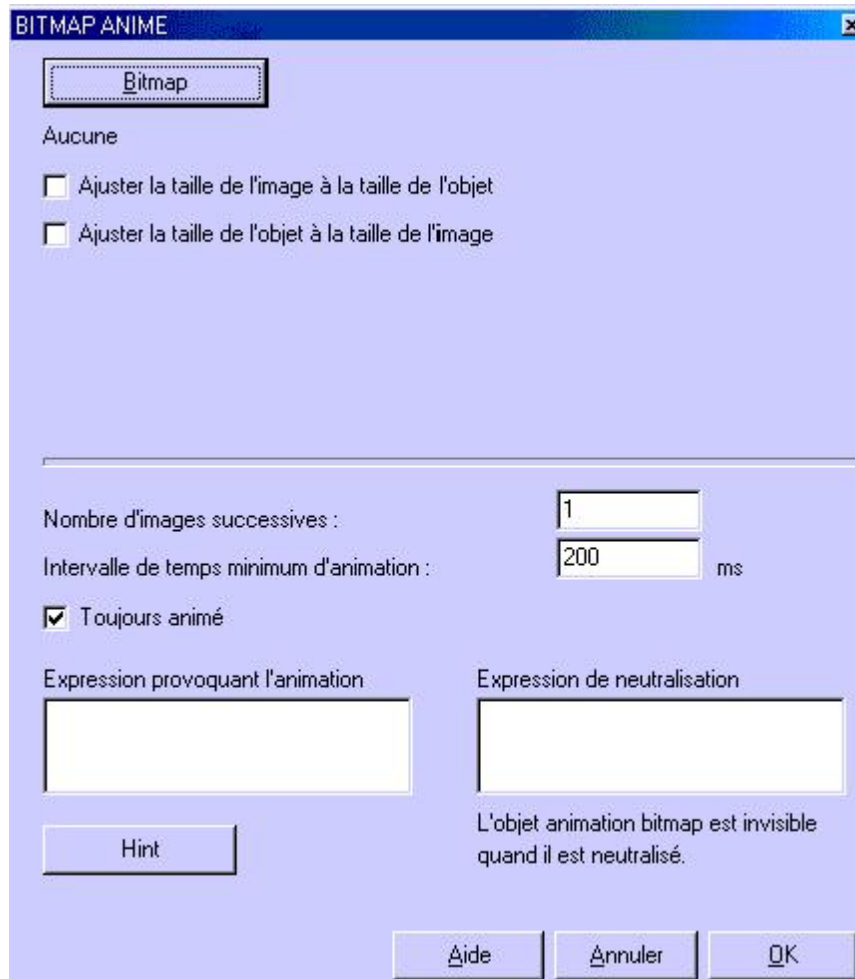
### 3.2.6.7.4 Bitmap animé

Cet Objet permet de créer des animations autonomes sur des pages de supervision. Contrairement à l'objet animation, il n'y a pas de notion d'état dans cet objet.

Il faut pour cet objet créer un bitmap qui contiendra toutes les images de l'animation collées les unes aux autres de façon horizontale. Il est possible de consulter le projet d'exemple « Bitmap animé ». Ce projet est accessible dans le répertoire « \Program Files\OptimaLog\Apigraf IP\Exemples ».

#### Boîte de dialogue Objet Bitmap animé

.



Les paramètres sont les suivants :

« **Bouton Bitmap** »

Ce contrôle permet de sélectionner l'image bitmap de l'animation, ce bitmap contient les différentes images de l'animation collées les unes aux autres de façon horizontale. Le projet d'exemple « Bitmap animé » illustre la décomposition de l'image bitmap.

« **Ajuster la taille de l'image à la taille de l'objet** »

Lorsque cette option est activée, la taille de l'image est toujours la même que celle de l'objet. Le fait d'agrandir l'objet aura donc pour effet de zoomer l'image.

« **Ajuster la taille de l'objet à la taille de l'image** »

Permet d'avoir l'image en taille réelle avec la taille de l'objet ajustée à celle de l'image

« **Nombres d'images successives** »

Spécifie le nombre total d'images de l'animation présentes dans l'image bitmap. Si le nombre entré n'est pas correct, l'animation ne sera pas correcte. C'est ce paramètre qui permet de découper les images de l'animation dans l'image bitmap.

« **Option Toujours animé** »

Si cette option est cochée, l'objet sera toujours en animation lors de l'exécution de la page de supervision contenant l'objet.

« **Expression provoquant l'animation** »

Permet de spécifier une expression booléenne qui déclenchera l'animation lorsque celle-ci sera égale à TRUE. Dans ce cas, il faut décocher l'option « Toujours animé ».

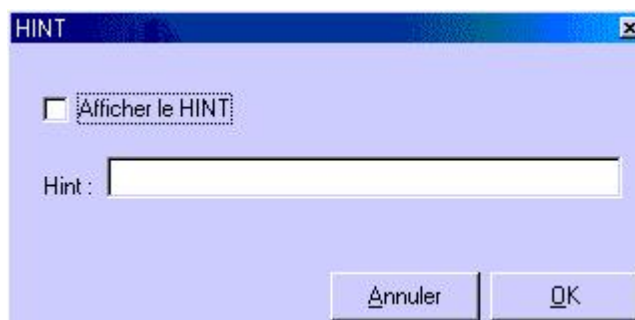
Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

**Bouton « Hint »**

Ce bouton permet de définir les paramètres de « Hint » de l'objet. Le texte « Hint » que l'on définit est le texte qui

apparaît sur fond jaune lorsque l'on passe le curseur de la souris sur l'objet.

Quand on clique sur le bouton « Hint », la fenêtre suivante s'ouvre :



Cette fenêtre présente deux contrôles :

#### Option « Afficher le Hint »

Si cette option n'est pas cochée, le texte du Hint n'apparaîtra pas quand le curseur de la souris passera sur l'objet même si celui-ci est défini. Il faut donc cocher cette option si l'on veut utiliser le « Hint » de l'objet.

#### Fenêtre de saisie « Hint »

Cette fenêtre permet d'entrer le texte du Hint. Ce texte peut être un paramètre (exemple {1}).

#### « Expression de neutralisation »

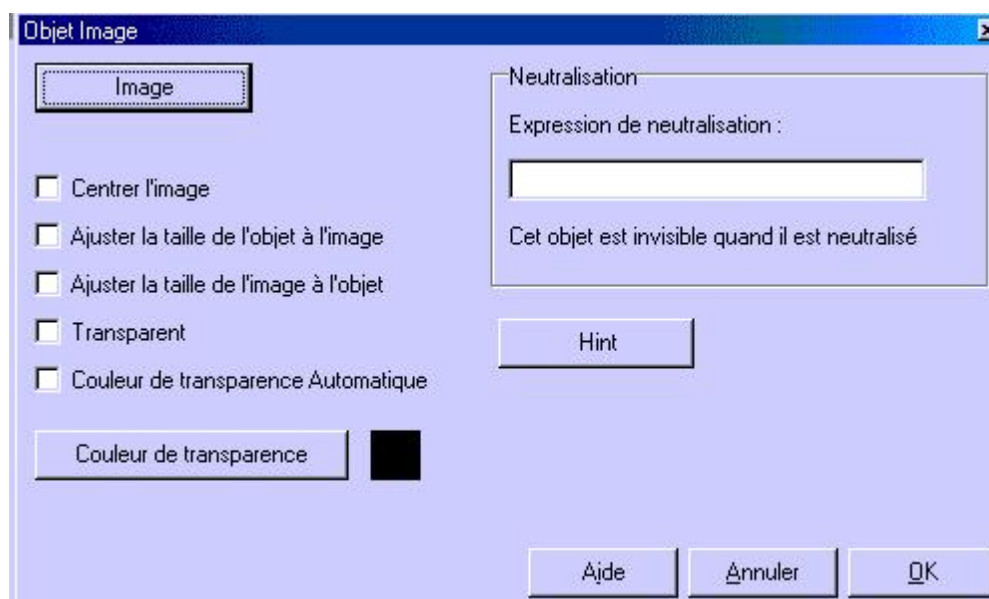
Permet de spécifier une expression booléenne permettant de rendre invisible l'objet animation lorsque celle-ci sera égale à TRUE.

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1}) Voir Section « Appel d'une page de supervision avec passage de paramètres »)

### 3.2.6.7.5 Image

Cet objet permet d'afficher une image.

#### Boîte de dialogue paramètre Image



Les options suivantes sont disponibles :

#### Image

Permet de spécifier le fichier correspondant à l'image. Les formats suivants sont admis : \*.BMP ; \*.EMF ; \*.WMF ; \*.ICO

Remarque : Une collection d'images au format BITMAP est disponible dans le sous-répertoire APIGRAFIP \GRAPHIQUES\BITMAPS

### Option Centrer l'image

Cette option permet de centrer l'image dans l'objet.

### Option Ajuster la taille de l'objet à l'image

Cette option ajuste automatiquement la taille de l'objet créée à la taille réelle de l'image.

### Option Ajuster la taille de l'image à l'objet

Cette option ajuste automatiquement la taille de l'image à la taille réelle de l'objet. Le fait de redimensionner l'objet aura donc une influence sur la taille de l'image.

### Transparent

Permet de définir si l'on veut que l'image ait une couleur de transparence. Si tel est le cas, la couleur de l'image définie comme couleur de transparence ne sera pas affichée. L'image laisse alors apparaître ce qui se trouve en arrière plan au niveau de la couleur de transparence.

### Couleur de transparence Automatique

Si cette option est cochée, la couleur de transparence est déterminée automatiquement. C'est en principe la couleur de fond de l'image qui est prise en compte.

### Couleur de transparence

Cette option permet dans le cas où l'option « Couleur de transparence Automatique » n'est pas cochée de choisir la couleur de transparence de l'image.

### Expression de neutralisation

Permet de spécifier la variable ou l'expression de neutralisation (Celle-ci doit obligatoirement être de type booléen). L'image sera invisible chaque fois que la variable ou l'expression sera égale à true.

*Exemple* : SA10

Il est possible de spécifier l'adresse IP distante du noyau APIGRAF PLC dans l'expression. Si c'est le cas, c'est cette adresse qui sera prise en compte pour le rafraîchissement.

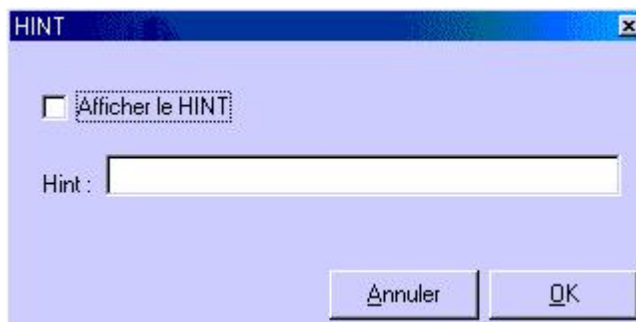
*Exemple* : !193.0.1.1 !SA10

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

### Bouton « Hint »

Ce bouton permet de définir les paramètres de « Hint » de l'objet. Le texte « Hint » que l'on définit est le texte qui apparaît sur fond jaune lorsque l'on passe le curseur de la souris sur l'objet.

Quand on clique sur le bouton « Hint », le fenêtre suivante s'ouvre :



Cette fenêtre présente deux contrôles :

### Option « Afficher le Hint »

Si cette option n'est pas cochée, le texte du Hint n'apparaîtra pas quand le curseur de la souris passera sur l'objet même si celui-ci est défini. Il faut donc cocher cette option si l'on veut utiliser le « Hint » de l'objet.

### Fenêtre de saisie « Hint »

Cette fenêtre permet d'entrer le texte du Hint. Ce texte peut être un paramètre (exemple {1}).

## 3.2.6.8 Objets de l'onglet JAUGES

### 3.2.6.8.1 Jauge

L'objet jauge permet de visualiser graphiquement l'état d'une variable de type numérique. Il ne permet cependant pas la modification de cette variable.

#### Boîte de dialogue paramètre Objet Jauge

Les options sont les suivantes :

#### Expression

Indiquer dans ce champ la variable automate à utiliser (de type numérique entier Exemple : R3)

*Exemple : R12*

Il est possible de spécifier l'adresse IP distante du noyau APIGRAF PLC dans l'expression. Si c'est le cas, c'est cette adresse qui sera prise en compte pour le rafraîchissement

*Exemple : !193.0.1.1 !R12*

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

#### Bouton « Hint »

Ce bouton permet de définir les paramètres de « Hint » de l'objet. Le texte « Hint » que l'on définit est le texte qui apparaît sur fond jaune lorsque l'on passe le curseur de la souris sur l'objet.

Quand on clique sur le bouton « Hint », le fenêtre suivante s'ouvre :



Cette fenêtre présente deux contrôles :

#### Option « Afficher le Hint »

Si cette option n'est pas cochée, le texte du Hint n'apparaîtra pas quand le curseur de la souris passera sur l'objet même si celui-ci est défini. Il faut donc cocher cette option si l'on veut utiliser le « Hint » de l'objet.

#### Fenêtre de saisie « Hint »

Cette fenêtre permet d'entrer le texte du Hint. Ce texte peut être un paramètre (exemple {1}).

#### Expression de neutralisation

Permet de spécifier la variable ou l'expression de neutralisation (Celle-ci doit obligatoirement être de type booléen). La jauge sera invisible chaque fois que la variable ou l'expression sera égale à true.

*Exemple : SA10*

Il est possible de spécifier l'adresse IP distante du noyau APIGRAF PLC dans l'expression. Si c'est le cas, c'est cette adresse qui sera prise en compte pour le rafraîchissement.

*Exemple : !193.0.1.1 !SA10*

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

#### Couleur Barre

Permet de choisir la couleur de la barre de la jauge.

#### Couleur Fond

Permet de choisir la couleur de fond de la jauge.

#### Type Forme

Cette option permet de choisir la forme de la jauge ( Barre horizontale ou verticale, camembert ou cadran).

#### Option Affichage pourcentage

Cette option permet d'afficher le pourcentage de remplissage de la jauge par rapport aux valeurs min et max.

#### Valeur min

Définit la valeur en dessous de laquelle la jauge est à sa position minimale.

Il est possible de spécifier dans ce champ une valeur constante ou l'expression d'une variable automate :

*Exemple : R20*

Pour les variables automates, on peut de plus spécifier l'adresse IP distante du noyau APIGRAF PLC dans l'expression. Si c'est le cas, c'est cette adresse qui sera prise en compte pour le rafraîchissement de la valeur Min.

*Exemple : !193.0.1.1 !R20*

Dans ce cas, la borne min de la jauge sera variable.

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

#### Valeur max

Définit la valeur au-dessus de laquelle la jauge est à sa position maximale.

Ce champ offre les mêmes options d'utilisation que le champ « Valeur Min » ci dessus.

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

### 3.2.6.8.2 Vu mètre

Cet objet a des propriétés similaires à un objet jauge.

Seul son aspect de vu mètre à leds le différencie de cet objet. Il permet de visualiser une variable numérique (Exemple un registre : R12) :

#### Aspect de l'objet vu mètre



#### Boîte de dialogue objet vu mètre

Les paramètres sont :

#### « Expression »

Indiquer dans ce champ le nom de la variable à contrôler. Cette variable doit être de type numérique entier (Registre).

*Exemple : R12*

Il est possible de spécifier l'adresse IP distante du noyau APIGRAF PLC dans l'expression. Si c'est le cas, c'est cette adresse qui sera prise en compte pour le rafraîchissement.

*Exemple : !193.0.1.1 !R12*

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

#### « Valeur Min »

Cette option définit la valeur en dessous de laquelle le curseur est à sa position minimale.

Il est possible de spécifier dans ce champ une valeur constante ou l'expression d'une variable automate :

*Exemple : R20*

Pour les variables automates, on peut de plus spécifier l'adresse IP distante du noyau APIGRAF PLC dans l'expression. Si c'est le cas, c'est cette adresse qui sera prise en compte pour le rafraîchissement de la valeur Min.

*Exemple : !193.0.1.1 !R12*

Dans ce cas, la borne min de la barre de défilement sera variable.

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

#### « Valeur Max »

Cette option définit la valeur au-dessus de laquelle le curseur est à sa position maximale.

Ce champ offre les mêmes options d'utilisation que le champ « Valeur Min » ci dessus.

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

#### « Orientation »

Cette option permet de choisir le sens d'affichage du curseur, vertical ou horizontal.

#### « Largeur des leds »

Permet de définir la largeur de chacune des leds composant le vu mètre.

#### Expression de neutralisation

Permet de spécifier la variable ou l'expression de neutralisation (Celle-ci doit obligatoirement être de type booléen). Le vu mètre sera invisible chaque fois que la variable ou l'expression sera égale à true.

*Exemple : SA10*

Il est possible de spécifier l'adresse IP distante du noyau APIGRAF PLC dans l'expression. Si c'est le cas, c'est cette adresse qui sera prise en compte pour le rafraîchissement.

*Exemple : !193.0.1.1 !SA10*

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

#### Bouton « Hint »

Ce bouton permet de définir les paramètres de « Hint » de l'objet. Le texte « Hint » que l'on définit est le texte qui apparaît sur fond jaune lorsque l'on passe le curseur de la souris sur l'objet.

Quand on clique sur le bouton « Hint », la fenêtre suivante s'ouvre :



Cette fenêtre présente deux contrôles :

#### Option « Afficher le Hint »

Si cette option n'est pas cochée, le texte du Hint n'apparaîtra pas quand le curseur de la souris passera sur l'objet même si celui-ci est défini. Il faut donc cocher cette option si l'on veut utiliser le « Hint » de l'objet.



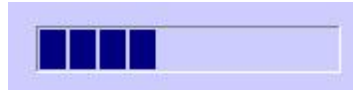
**Fenêtre de saisie « Hint »**

Cette fenêtre permet d'entrer le texte du Hint. Ce texte peut être un paramètre (exemple {1}).

**3.2.6.8.3 Barre de progression**

Cet objet permet de contrôler une barre de progression avec une variable automate.

*Aperçu d'une barre de progression :*

**Boîte de dialogue « Barre de progression »**

La boîte de dialogue « Barre de progression » contient les éléments suivants :

- Un champ de saisie « Expression ».
- Deux champs de saisie pour « Valeur Min » (contenant 0) et « Valeur Max » (contenant 100).
- Un bouton « Hint ».
- Une section « Orientation » avec deux boutons radio : « Horizontale » (sélectionné) et « Verticale ».
- Une section « Neutralisation » avec un champ de saisie « Expression de neutralisation : » et le texte « L'objet est invisible quand il est neutralisé ».
- Des boutons « Aide », « Annuler » et « OK » en bas à droite.

Les champs sont les suivants :

**« Expression »**

Indiquer dans ce champ le nom de la variable automate qui doit contrôler la barre de progression. Cette variable doit être de type numérique entier (Registre).

*Exemple : R12*

Il est possible de spécifier l'adresse IP distante du noyau APIGRAF PLC dans l'expression. Si c'est le cas, c'est cette adresse qui sera prise en compte pour le rafraîchissement.

*Exemple : !193.0.1.1 !R12*

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

**« Orientation »**

Cette option permet de choisir le sens d'affichage de la barre de progression, vertical ou horizontal.

**« Valeur Min »**

Cette option définit la valeur pour laquelle la barre de progression est à son minimum.

Il est possible de spécifier dans ce champ une valeur constante ou l'expression d'une variable automate de type entier :

*Exemple : R20*

Pour les variables automates, on peut de plus spécifier l'adresse IP distante du noyau APIGRAF PLC dans l'expression. Si c'est le cas, c'est cette adresse qui sera prise en compte pour le rafraîchissement de la valeur Min.

Exemple : !193.0.1.1 !R12

Dans ce cas, la borne min de la barre de progression sera variable.

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

#### « Valeur Max »

Cette option définit la valeur pour laquelle la barre de progression est maximale.

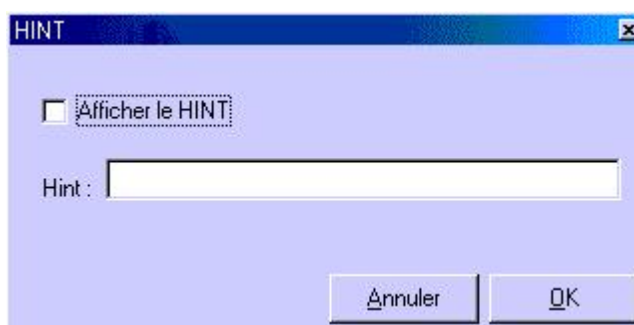
Ce champ offre les mêmes options d'utilisation que le champ « Valeur Min » ci dessus.

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

#### Bouton « Hint »

Ce bouton permet de définir les paramètres de « Hint » de l'objet. Le texte « Hint » que l'on définit est le texte qui apparaît sur fond jaune lorsque l'on passe le curseur de la souris sur l'objet.

Quand on clique sur le bouton « Hint », le fenêtre suivante s'ouvre :



Cette fenêtre présente deux contrôles :

#### Option « Afficher le Hint »

Si cette option n'est pas cochée, le texte du Hint n'apparaîtra pas quand le curseur de la souris passera sur l'objet même si celui-ci est défini. Il faut donc cocher cette option si l'on veut utiliser le « Hint » de l'objet.

#### Fenêtre de saisie « Hint »

Cette fenêtre permet d'entrer le texte du Hint. Ce texte peut être un paramètre (exemple {1}).

#### Expression de neutralisation

Permet de spécifier la variable ou l'expression de neutralisation (Celle-ci doit obligatoirement être de type booléen). La barre de progression sera invisible chaque fois que la variable ou l'expression sera égale à true.

Exemple : SA10

Il est possible de spécifier l'adresse IP distante du noyau APIGRAF PLC dans l'expression. Si c'est le cas, c'est cette adresse qui sera prise en compte pour le rafraîchissement.

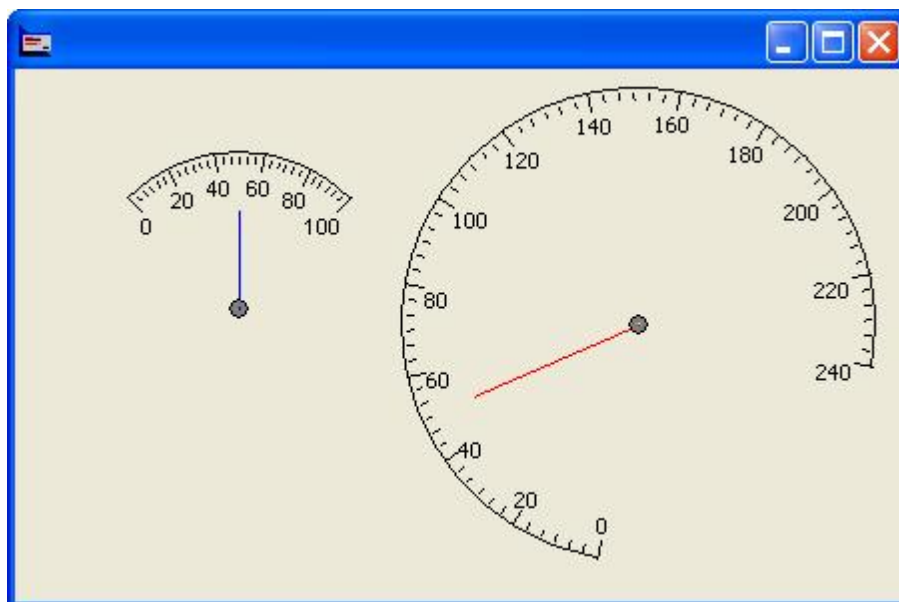
Exemple : !193.0.1.1 !SA10

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

### 3.2.6.8.4 Jauge circulaire simple

L'objet "Jauge circulaire simple" permet de visualiser graphiquement l'état d'une variable de type numérique. Il ne permet pas la modification de cette variable.

Exemple de jauge circulaire simple:



**Boîte de dialogue « jauge circulaire simple » - Onglet Général:**

**Jauge circulaire simple**

Général Neutralisation Graduations

Expression :

Valeur Min :

Valeur Max :

Couleur de l'aiguille

Hint

Angle de départ du cadran :  degrés

Angle total du cadran :  degrés

Aide Annuler OK

Les paramètres sont les suivants:

**Expression**

Indiquer dans ce champ le nom de la variable à visualiser dans le Vu mètre à aiguille. Cette variable doit être de type numérique.

*Exemple :* R12

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1}) Voir Section « Appel d'une page de supervision avec passage de paramètres »)

**Valeur Min**

Ce paramètre indique la valeur de la borne inférieure des graduations. Ce paramètre doit être de type numérique.

*Exemple :* 100

R13

**Valeur Max**

Ce paramètre indique la valeur de la borne supérieur des graduations. Ce paramètre doit être de type numérique.

Exemple : 1000  
R14

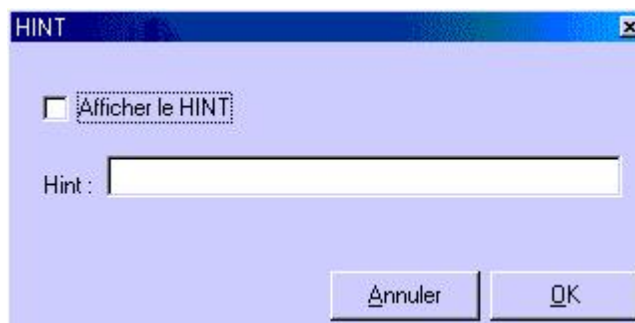
**Couleur de l'aiguille**

Permet de sélectionner la couleur de l'aiguille de la jauge circulaire.

**Bouton « Hint »**

Ce bouton permet de définir les paramètres de « Hint » de l'objet. Le texte « Hint » que l'on définit est le texte qui apparaît sur fond jaune lorsque l'on passe le curseur de la souris sur l'objet.

Quand on clique sur le bouton « Hint », le fenêtre suivante s'ouvre :



Cette fenêtre présente deux contrôles :

**Option « Afficher le Hint »**

Si cette option n'est pas cochée, le texte du Hint n'apparaîtra pas quand le curseur de la souris passera sur l'objet même si celui-ci est défini. Il faut donc cocher cette option si l'on veut utiliser le « Hint » de l'objet.

**Fenêtre de saisie « Hint »**

Cette fenêtre permet d'entrer le texte du Hint. Ce texte peut être un paramètre (exemple {1}).

**Angle de départ du cadran**

Spécifie à quel valeur d'angle commence le cadran. Ceci permet de positionner la cadran en bas par exemple.

**Angle total du cadran**

Spécifie la valeur totale angulaire en degrés du cadran. Pour avoir un cadran sur un demi-cercle, entrer la valeur 180

**Boîte de dialogue « jauge circulaire simple» - Onglet Neutralisation :****Expression de neutralisation**

Permet de spécifier la variable ou l'expression de neutralisation (Celle-ci doit obligatoirement être de type booléen). L'objet sera invisible chaque fois que la variable ou l'expression sera égale à true.

Exemple : SA10

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

**Boîte de dialogue « jauge circulaire simple» - Onglet Graduations :**

**Jauge circulaire simple**

Général Neutralisation **Graduations**

Intervalle de graduation : 20

Nb de sous graduations entre deux graduations : 5

Taille des graduations : 10

Couleur des graduations [Black swatch]

Taille des sous graduations : 4

Couleur des sous graduations [Black swatch]

☒ Afficher les valeurs de graduation

Caractères Graduations

Aide Annuler OK

Cet onglet permet de définir l'affichage des graduations sur la jauge circulaire :

**Intervalle de graduation :**

Définit l'intervalle entre 2 graduations. Cette valeur doit être de type numérique.

**Nb de sous graduations entre 2 graduations**

Définit le nombre de repères affichés entre 2 graduations

**Taille des graduations**

Définit la longueur des graduations en pixels

**Bouton "Couleur des graduations"**

Définit la couleur des tirets qui symbolisent les graduations.

**Taille des sous-graduations**

Définit la longueur des sous-graduations en pixels

**Bouton "Couleur des sous-graduations"**

Définit la couleur des tirets qui symbolisent les sous-graduations.

**Option "Afficher les valeurs de graduations"**

Si cette option est cochée, les valeurs numériques correspondantes aux graduations seront affichées.

**Bouton "Caractères graduations"**

Ce bouton permet de définir la police d'affichage des valeurs de graduations.

### 3.2.6.8.5 Jauge circulaire

L'objet "Jauge circulaire" permet de visualiser graphiquement l'état d'une variable de type numérique. Il ne permet pas la modification de cette variable.

Exemple de jauge circulaire :



**Boîte de dialogue « jauge circulaire » - Onglet Général:**

**JAUGE CIRCULAIRE**

Général Neutralisation Fond Seuils Orientation Graduations

Expression :

Valeur Min :

Valeur Max :

Les paramètres sont les suivants:

**Expression**

Indiquer dans ce champ le nom de la variable à visualiser dans le Vu mètre à aiguille. Cette variable doit être de type numérique.

*Exemple :* R12

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

**Valeur Min**

Ce paramètre indique la valeur de la borne inférieure des graduations. Ce paramètre doit être de type numérique.

*Exemple :* 100  
R13

**Valeur Max**

Ce paramètre indique la valeur de la borne supérieur des graduations. Ce paramètre doit être de type numérique.

Exemple : 1000  
R14

### Bouton « Hint »

Ce bouton permet de définir les paramètres de « Hint » de l'objet. Le texte « Hint » que l'on définit est le texte qui apparaît sur fond jaune lorsque l'on passe le curseur de la souris sur l'objet.

Quand on clique sur le bouton « Hint », le fenêtre suivante s'ouvre :



Cette fenêtre présente deux contrôles :

### Option « Afficher le Hint »

Si cette option n'est pas cochée, le texte du Hint n'apparaîtra pas quand le curseur de la souris passera sur l'objet même si celui-ci est défini. Il faut donc cocher cette option si l'on veut utiliser le « Hint » de l'objet.

### Fenêtre de saisie « Hint »

Cette fenêtre permet d'entrer le texte du Hint. Ce texte peut être un paramètre (exemple {1}).

### Boîte de dialogue « jauge circulaire » - Onglet Neutralisation :

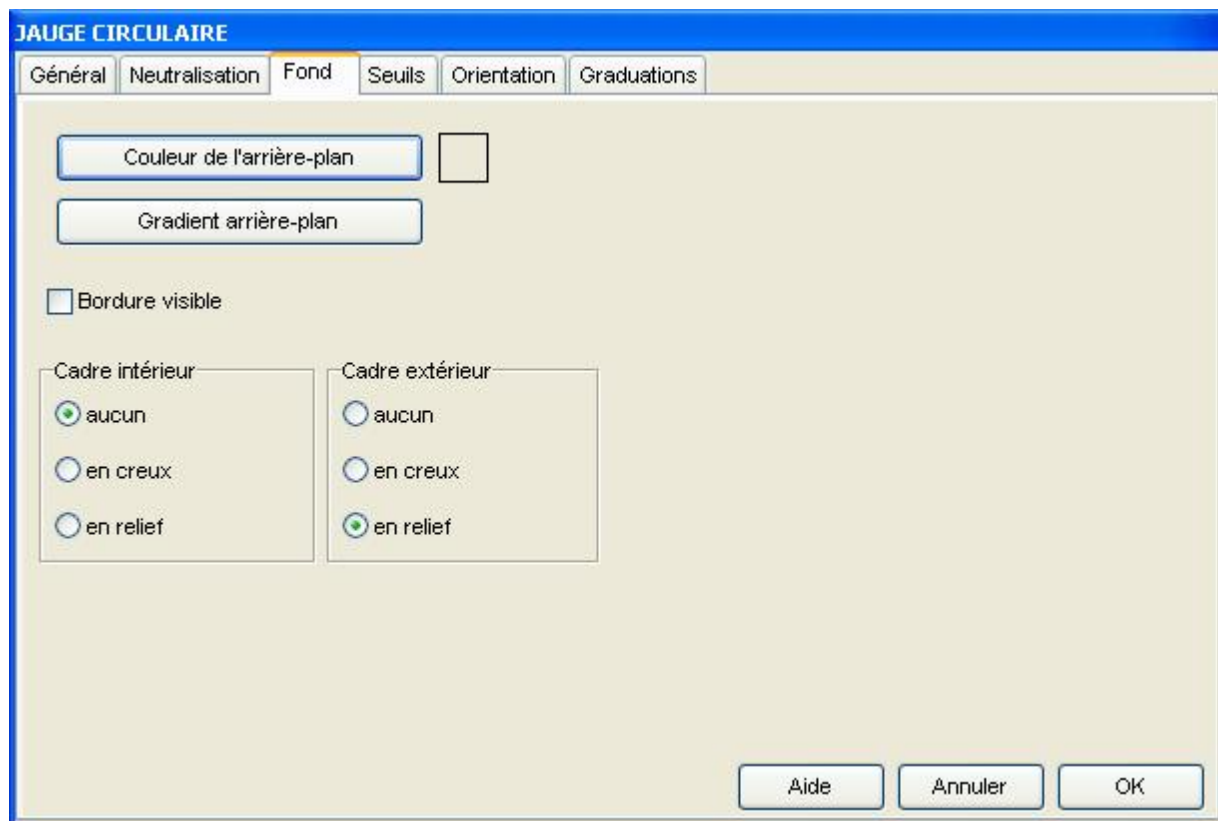
#### Expression de neutralisation

Permet de spécifier la variable ou l'expression de neutralisation (Celle-ci doit obligatoirement être de type booléen). L'objet sera invisible chaque fois que la variable ou l'expression sera égale à true.

Exemple : SA10

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

### Boîte de dialogue « jauge circulaire » - Onglet Fond :



**JAUGE CIRCULAIRE**

Général Neutralisation **Fond** Seuils Orientation Graduations

Couleur de l'arrière-plan

Gradient arrière-plan

☐ Bordure visible

Cadre intérieur

- ☒ aucun
- ☐ en creux
- ☐ en relief

Cadre extérieur

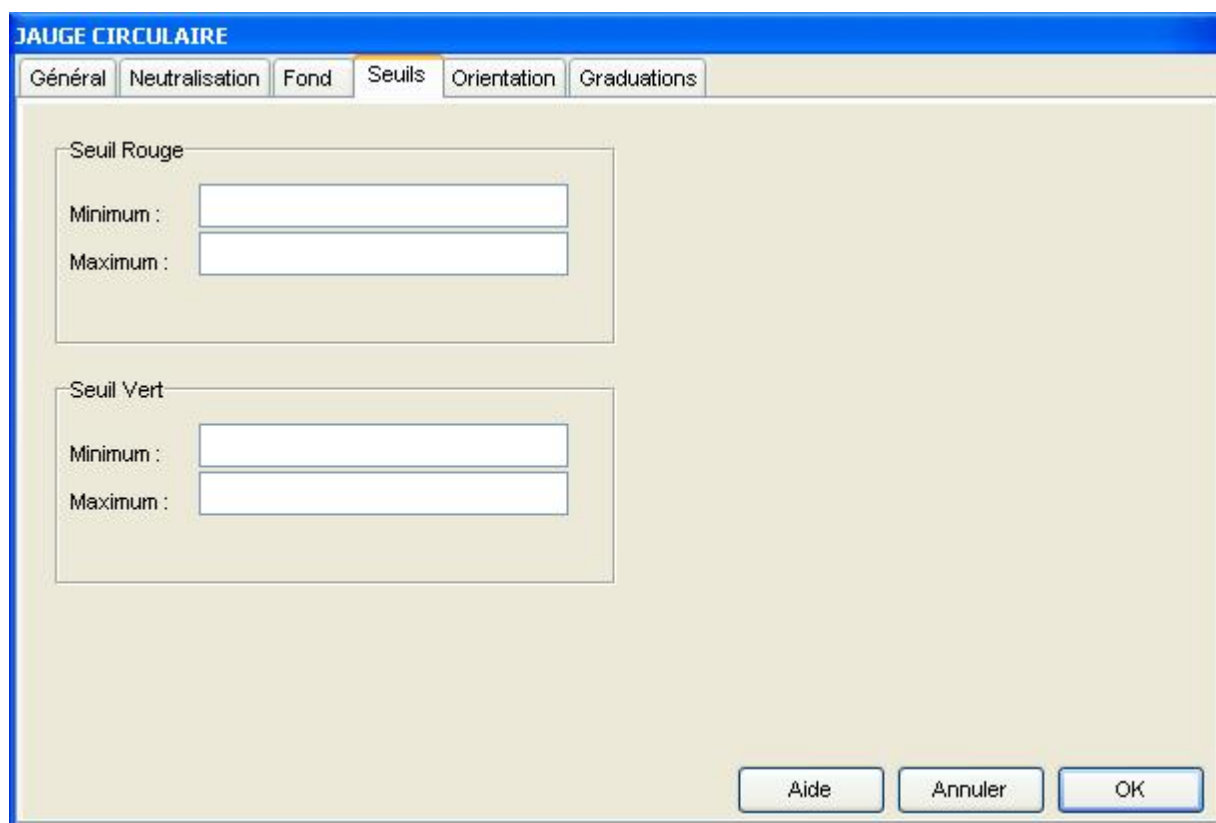
- ☐ aucun
- ☐ en creux
- ☒ en relief

Aide Annuler OK

Cet onglet permet de définir la couleur et l'aspect du cadre qui contient la jauge circulaire.

**Boîte de dialogue « jauge circulaire » - Onglet Seuils :**

Cet onglet permet de paramétrer deux zones repérées sur la jauge, une zone verte et une zone rouge.



**JAUGE CIRCULAIRE**

Général Neutralisation Fond **Seuils** Orientation Graduations

Seuil Rouge

Minimum :

Maximum :

Seuil Vert

Minimum :

Maximum :

Aide Annuler OK

Pour chaque seuil rouge et vert, les valeurs min et max sont paramétrables. Entrer des valeurs de types



numériques constantes ou variables.

Exemples : R10

100

0

#### **Boîte de dialogue « jauge circulaire » - Onglet Orientation :**

##### **Angle de départ :**

Permet de définir l'angle de départ du cadran de la jauge

##### **Angle total de la jauge :**

Permet de définir la plage angulaire d'affichage de la jauge.

#### **Boîte de dialogue « jauge circulaire » - Onglet Graduations :**

Cet onglet permet de définir l'affichage des graduations sur la jauge circulaire :

##### **Intervalle de graduation :**

Définit l'intervalle entre 2 graduations. Cette valeur doit être de type numérique.

##### **Nb de sous graduations entre 2 graduations**

Définit le nombre de repères affichés entre 2 graduations

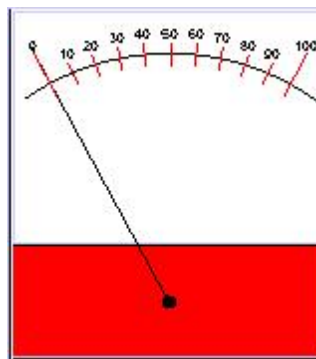
##### **Distance des sous graduations**

Définit la distance entre les sous graduations et le cadran. La valeur de type numérique entière est donnée en nombre de pixels.

### **3.2.6.8.6 Vu mètre à aiguille**

L'objet Vu mètre à aiguille permet de visualiser graphiquement l'état d'une variable de type numérique. Il ne permet pas cependant la modification de cette variable.

Exemple de Vu mètre à aiguille :



#### **Boîte de dialogue « Vu mètre à aiguille » :**

Les paramètres sont les suivants:

### Expression

Indiquer dans ce champ le nom de la variable à visualiser dans le Vu mètre à aiguille. Cette variable doit être de type numérique entier (Registre).

*Exemple* : R12

Il est possible de spécifier l'adresse IP distante du noyau APIGRAF PLC dans l'expression. Si c'est le cas, c'est cette adresse qui sera prise en compte pour le rafraîchissement.

*Exemple* : !193.0.1.1 !R12

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

### Légende

Permet de mettre une légende au-dessus de l'objet. Ce paramètre n'accepte que du texte.

### Bouton « Caractères texte »

Permet l'accès à la boîte de dialogue concernant le type d'affichage de la légende (couleur, police...).

### Couleur Fond

Spécifie la couleur de fond du Vu mètre à aiguille.

### Couleur Base

Spécifie la couleur de la partie basse du Vu mètre à aiguille.

### Couleur Aiguille

Spécifie la couleur de l'aiguille.

### Echelle Min

Ce paramètre indique la valeur de la borne inférieure des graduations du Vu mètre à aiguille. Ce paramètre constant peut être de type entier ou décimal

*Exemple* : 10  
0,5

### Echelle Max

Ce paramètre indique la valeur de la borne supérieur des graduations du Vu mètre à aiguille. Ce paramètre constant peut être de type entier ou décimal

*Exemple* : 100  
10,05

### Intervalle graduation

Ce paramètre indique la valeur séparant deux graduations. Ce paramètre constant peut être entier ou décimal.

Exemple : 10  
1,5

### Nombre de décimales

Les graduations peuvent être affichées avec des décimales. Ce paramètre indique le nombre de décimales affichées. Ce paramètre est de type entier.

### Bouton Caractère Graduations

Permet l'accès à la boîte de dialogue concernant le type d'affichage des graduations (couleur, police...).

### Expression de neutralisation

Permet de spécifier la variable ou l'expression de neutralisation (Celle-ci doit obligatoirement être de type booléen). L'objet sera invisible chaque fois que la variable ou l'expression sera égale à true.

*Exemple* : SA10

Il est possible de spécifier l'adresse IP distante du noyau APIGRAF PLC dans l'expression. Si c'est le cas, c'est cette adresse qui sera prise en compte pour le rafraîchissement.

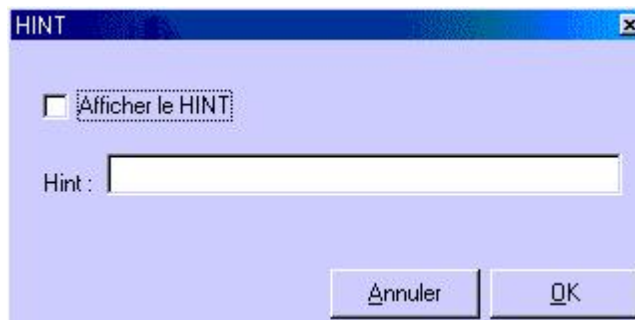
*Exemple* : !193.0.1.1 !SA10

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

### Bouton « Hint »

Ce bouton permet de définir les paramètres de « Hint » de l'objet. Le texte « Hint » que l'on définit est le texte qui apparaît sur fond jaune lorsque l'on passe le curseur de la souris sur l'objet.

Quand on clique sur le bouton « Hint », la fenêtre suivante s'ouvre :



Cette fenêtre présente deux contrôles :

#### Option « Afficher le Hint »

Si cette option n'est pas cochée, le texte du Hint n'apparaîtra pas quand le curseur de la souris passera sur l'objet même si celui-ci est défini. Il faut donc cocher cette option si l'on veut utiliser le « Hint » de l'objet.

#### Fenêtre de saisie « Hint »

Cette fenêtre permet d'entrer le texte du Hint. Ce texte peut être un paramètre (exemple {1}).

## 3.2.6.9 Objets de l'onglet CURSEURS / TRACKBARS

### 3.2.6.9.1 Curseur

Cet objet permet de contrôler une variable numérique à l'aide d'un curseur, dont le déplacement provoque une variation proportionnelle de la variable.

**Boîte de dialogue paramètre Curseur**

Les champs de saisie de cette boîte sont les suivants :

**Expression**

Indiquer dans ce champ le nom de la variable à contrôler. Cette variable doit être de type numérique entier (Registre).

*Exemple : R12*

Il est possible de spécifier l'adresse IP distante du noyau APIGRAF PLC dans l'expression. Si c'est le cas, c'est cette adresse qui sera prise en compte pour le rafraîchissement.

*Exemple : !193.0.1.1 !R12*

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

**Orientation**

Cette option permet de définir si le curseur doit être orienté horizontalement ou verticalement.

**Progression**

Permet de spécifier le sens de progression du curseur, c'est à dire le sens dans lequel la valeur associée au curseur grandit.

Si le curseur est orienté horizontalement, on peut choisir une progression de droite à gauche ou de gauche à droite.

Si le curseur est orienté verticalement, on peut choisir une progression de haut en bas ou de bas en haut.

**Valeur Min**

Cette option définit la valeur en dessous de laquelle le curseur est à sa position minimale.

Il est possible de spécifier dans ce champ une valeur constante ou l'expression d'une variable automate :

*Exemple : R20*

Pour les variables automates, on peut de plus spécifier l'adresse IP distante du noyau APIGRAF PLC dans l'expression. Si c'est le cas, c'est cette adresse qui sera prise en compte pour le rafraîchissement de la valeur Min.

*Exemple : !193.0.1.1 !R12*

Dans ce cas, la borne min du curseur sera variable.

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

### Valeur Max

Cette option définit la valeur au-dessus de laquelle le curseur est à sa position maximale.

Ce champ offre les mêmes options d'utilisation que le champ « Valeur Min » ci-dessus.

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

### Graduation

Spécifie la résolution de la graduation qui est liée au curseur au niveau de l'affichage.

### Zone sélectionnée : Début et Fin

Cette option permet d'afficher une certaine zone du curseur en bleu.

### Position graduations

Permet de spécifier l'endroit où est affichée la graduation associée au curseur. Si l'orientation est verticale, on peut choisir une position à droite ou à gauche. Si l'orientation est horizontale, on peut choisir une position en haut ou en bas.

### Expression de neutralisation

Permet de spécifier la variable ou l'expression de neutralisation (Celle-ci doit obligatoirement être de type booléen). Le curseur sera neutralisé chaque fois que la variable ou l'expression sera égale à true.

*Exemple : SA10*

Il est possible de spécifier l'adresse IP distante du noyau APIGRAF PLC dans l'expression. Si c'est le cas, c'est cette adresse qui sera prise en compte pour le rafraîchissement.

*Exemple : !193.0.1.1 !SA10*

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

### Option « Visible si neutralisé »

Si cette option est cochée, le curseur sera visible quand il sera neutralisé. Cependant, l'opérateur ne pourra pas agir dessus.

Si cette option n'est pas cochée, le curseur sera invisible quand il sera neutralisé.

### Bouton « Hint »

Ce bouton permet de définir les paramètres de « Hint » de l'objet. Le texte « Hint » que l'on définit est le texte qui apparaît sur fond jaune lorsque l'on passe le curseur de la souris sur l'objet.

Quand on clique sur le bouton « Hint », la fenêtre suivante s'ouvre :



Cette fenêtre présente deux contrôles :

### Option « Afficher le Hint »

Si cette option n'est pas cochée, le texte du Hint n'apparaîtra pas quand le curseur de la souris passera sur l'objet même si celui-ci est défini. Il faut donc cocher cette option si l'on veut utiliser le « Hint » de l'objet.

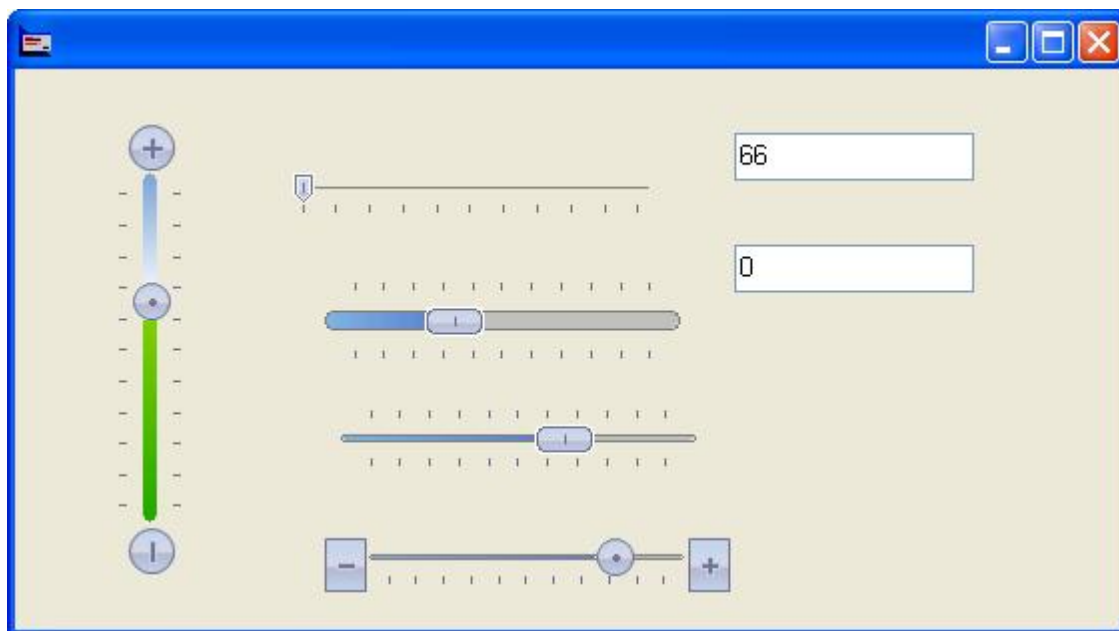
**Fenêtre de saisie « Hint »**

Cette fenêtre permet d'entrer le texte du Hint. Ce texte peut être un paramètre (exemple {1}).

**3.2.6.9.2 Advanced TrackBar**

Cet objet permet de contrôler une variable numérique à l'aide d'un curseur, dont le déplacement provoque une variation proportionnelle de la variable.

Le dessin de ce curseur est plus évolué que celui du curseur classique et est de plus paramétrable.

**Boîte de dialogue paramètre Curseur - Onglet GENERAL**

**CURSEUR / TRACKBAR**

Général Neutralisation Boutons Curseur Glissière Fond

Expression :

Valeur Min :

Valeur Max :

Intervalle graduations :

Position graduations :  ▼

Progression :  ▼

Orientation

☒ Horizontale

☐ Verticale

Les champs de saisie de cette boîte sont les suivants :

**Expression**

Indiquer dans ce champ le nom de la variable à contrôler. Cette variable doit être de type numérique.

*Exemple : R12*

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

### **Orientation**

Cette option permet de définir si le curseur doit être orienté horizontalement ou verticalement.

### **Progression**

Permet de spécifier le sens de progression du curseur, c'est à dire le sens dans lequel la valeur associée au curseur grandit.

Si le curseur est orienté horizontalement, on peut choisir une progression de droite à gauche ou de gauche à droite.

Si le curseur est orienté verticalement, on peut choisir une progression de haut en bas ou de bas en haut.

### **Valeur Min**

Cette option définit la valeur en dessous de laquelle le curseur est à sa position minimale.

Il est possible de spécifier dans ce champ une valeur constante ou l'expression d'une variable automate :

*Exemple : R20*

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

### **Valeur Max**

Cette option définit la valeur au-dessus de laquelle le curseur est à sa position maximale.

Ce champ offre les mêmes options d'utilisation que le champ « Valeur Min » ci dessus.

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

### **Intervalle Graduation**

Spécifie la résolution de la graduation qui est liée au curseur au niveau de l'affichage.

### **Position graduations**

Permet de spécifier l'endroit où est affichée la graduation associée au curseur. Si l'orientation est verticale, on peut choisir une position à droite ou à gauche. Si l'orientation est horizontale, on peut choisir une position en haut ou en bas.

### **Bouton « Hint »**

Ce bouton permet de définir les paramètres de « Hint » de l'objet. Le texte « Hint » que l'on définit est le texte qui apparaît sur fond jaune lorsque l'on passe le curseur de la souris sur l'objet.

Quand on clique sur le bouton « Hint », la fenêtre suivante s'ouvre :



Cette fenêtre présente deux contrôles :

### **Option « Afficher le Hint »**

Si cette option n'est pas cochée, le texte du Hint n'apparaîtra pas quand le curseur de la souris passera sur l'objet même si celui-ci est défini. Il faut donc cocher cette option si l'on veut utiliser le « Hint » de l'objet.

### **Fenêtre de saisie « Hint »**

Cette fenêtre permet d'entrer le texte du Hint. Ce texte peut être un paramètre (exemple {1}).

### **Boîte de dialogue paramètre Curseur - Onglet NEUTRALISATION**

#### **Expression de neutralisation**

Permet de spécifier la variable ou l'expression de neutralisation (Celle-ci doit obligatoirement être de type booléen). Le curseur sera neutralisé chaque fois que la variable ou l'expression sera égale à true.

*Exemple : SA10*

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

#### **Option « Visible si neutralisé »**

Si cette option est cochée, le curseur sera visible quand il sera neutralisé. Cependant, l'opérateur ne pourra pas agir dessus.

Si cette option n'est pas cochée, le curseur sera invisible quand il sera neutralisé.

### **Boîte de dialogue paramètre Curseur - Onglet BOUTONS**

Ce curseur offre la possibilité de positionner des boutons à chaque extrémité du curseur. Ces boutons agiront sur le curseur dans le sens plus et dans le sens moins.

#### **Option « Boutons d'extrémité visibles »**

Si cette option est cochée, les boutons seront visibles à chaque extrémité.

#### **Forme des boutons :**

Choisir ici la forme des boutons (Ronds ou Carrés)

### **Boîte de dialogue paramètre Curseur - Onglet CURSEUR**

#### **Forme du curseur**

Ce champ permet de sélectionner la forme du curseur : Pointeur, Rectangle, Rond, Rectangle arrondi,.....

#### **Taille et largeur du curseur**

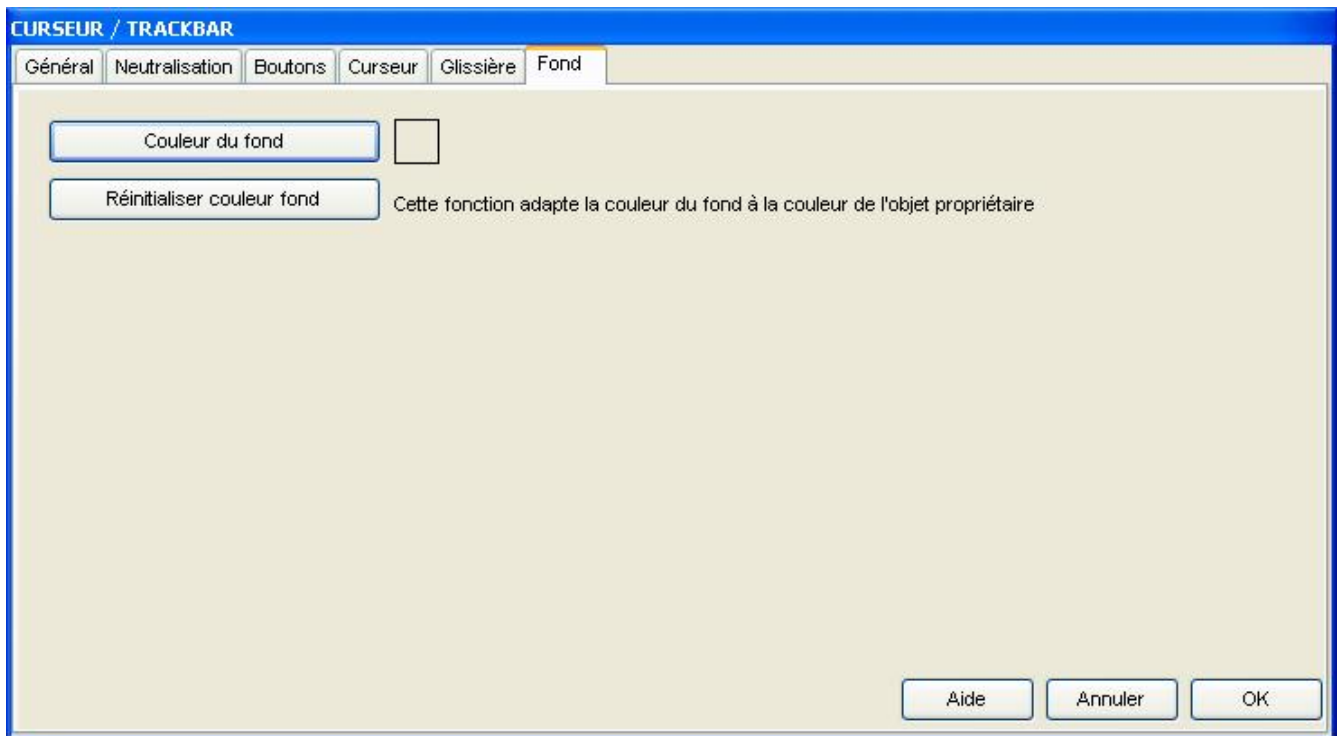
Ces 2 champs permettent de dimensionner le curseur.

### **Boîte de dialogue paramètre Curseur - Onglet GLISSIERE**

Cet onglet permet de paramétrer l'aspect de la glissière du curseur.

### **Boîte de dialogue paramètre Curseur - Onglet FOND**



**Bouton "Couleur de fond"**

Permet de choisir la couleur de fond du curseur.

**Bouton "Réinitialiser la couleur de fond"**

Permet d'initialiser la couleur de fond avec la couleur de fond de l'objet propriétaire.

### 3.2.6.9.3 Barre de défilement

L'objet barre de défilement est une variante de l'objet curseur. Son aspect est celui des barres de défilement standard de Windows®. Cet objet permet de faire varier une variable automate de type numérique entière (registre).

**Boîte de dialogue paramètre barre de défilement**

Objet barre de défilement

Expression

Progression: Gauche à droite

Valeur Min: 0

Valeur Max: 100

Orientation: ☒ Horizontale ☐ Verticale

Neutralisation

Expression de neutralisation :

☒ visible si neutralisé

Hint

Aide Annuler OK

Les champs sont les suivants :

### Expression

Indiquer dans ce champ le nom de la variable à contrôler. Cette variable doit être de type numérique entier (Registre).

*Exemple : R12*

Il est possible de spécifier l'adresse IP distante du noyau APIGRAF PLC dans l'expression. Si c'est le cas, c'est cette adresse qui sera prise en compte pour le rafraîchissement.

*Exemple : !193.0.1.1 !R12*

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

### Orientation

Cette option permet de choisir le sens d'affichage du curseur, vertical ou horizontal.

### Progression

Permet d'indiquer dans quel sens la valeur de la variable augmente.

### Valeur min

Cette option définit la valeur en dessous de laquelle le curseur est à sa position minimale.

Il est possible de spécifier dans ce champ une valeur constante ou l'expression d'une variable automate :

*Exemple : R20*

Pour les variables automates, on peut de plus spécifier l'adresse IP distante du noyau APIGRAF PLC dans l'expression. Si c'est le cas, c'est cette adresse qui sera prise en compte pour le rafraîchissement de la valeur Min.

*Exemple : !193.0.1.1 !R12*

Dans ce cas, la borne min de la barre de défilement sera variable.

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

### Valeur max

Cette option définit la valeur au-dessus de laquelle le curseur est à sa position maximale.

Ce champ offre les mêmes options d'utilisation que le champ « Valeur Min » ci dessus.

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

#### Bouton « Hint »

Ce bouton permet de définir les paramètres de « Hint » de l'objet. Le texte « Hint » que l'on définit est le texte qui apparaît sur fond jaune lorsque l'on passe le curseur de la souris sur l'objet.

Quand on clique sur le bouton « Hint », la fenêtre suivante s'ouvre :



Cette fenêtre présente deux contrôles :

#### Option « Afficher le Hint »

Si cette option n'est pas cochée, le texte du Hint n'apparaîtra pas quand le curseur de la souris passera sur l'objet même si celui-ci est défini. Il faut donc cocher cette option si l'on veut utiliser le « Hint » de l'objet.

#### Fenêtre de saisie « Hint »

Cette fenêtre permet d'entrer le texte du Hint. Ce texte peut être un paramètre (exemple {1}).

#### Expression de neutralisation

Permet de spécifier la variable ou l'expression de neutralisation (Celle-ci doit obligatoirement être de type booléen). La barre de défilement sera neutralisée chaque fois que la variable ou l'expression sera égale à true.

*Exemple : SA10*

Il est possible de spécifier l'adresse IP distante du noyau APIGRAF PLC dans l'expression. Si c'est le cas, c'est cette adresse qui sera prise en compte pour le rafraîchissement.

*Exemple : !193.0.1.1 !SA10*

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

#### Option « Visible si neutralisé »

Si cette option est cochée, la barre de défilement sera visible quand elle sera neutralisée. Cependant, l'opérateur ne pourra pas agir sur la barre et celle-ci sera grisée.

Si cette option n'est pas cochée, la barre de défilement sera invisible quand elle sera neutralisée.

### 3.2.6.10 Objets de l'onglet INTERRUPTEURS

#### 3.2.6.10.1 Bouton Marche/Arrêt

L'objet Bouton Marche / Arrêt simule un interrupteur. L'apparence de cet objet varie avec un clic sur le bouton gauche de la souris au-dessus de l'objet ou en fonction de la valeur de sa variable de contrôle.

Cette variable peut être de type booléen (flag L, Interrupteur logique SA...) ou entier (Registre R...).

Image du bouton :



### Boîte de dialogue « Bouton Marche Arrêt » :

Les paramètres sont les suivants :

#### Expression

Permet de définir le nom de la variable de contrôle de l'objet (variable associée au bouton Marche/Arrêt). Cette variable sert d'index pour chaque état de l'animation (un état correspond à une valeur de la variable).

Indiquer dans ce champ la variable automate à utiliser (de type numérique entier )

*Exemple* : R20

Il est possible de spécifier l'adresse IP distante du noyau APIGRAF PLC dans l'expression. Si c'est le cas, c'est cette adresse qui sera prise en compte pour le rafraîchissement.

*Exemple* : !193.0.1.1 !R20

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

#### Légende

Permet de mettre une légende au-dessus de l'objet. Ce paramètre n'accepte que du texte.

#### Bouton « Caractères texte »

Permet l'accès à la boîte de dialogue concernant le type d'affichage de la légende (couleur, police...).

#### Expression de neutralisation

Permet de spécifier la variable ou l'expression de neutralisation (Celle-ci doit obligatoirement être de type booléen)

*Exemple* : SA10

Il est possible de spécifier l'adresse IP distante du noyau APIGRAF PLC dans l'expression. Si c'est le cas, c'est cette adresse qui sera prise en compte pour le rafraîchissement.

*Exemple* : !193.0.1.1 !SA10

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

#### Option « visible si neutralisé »

Si cette option n'est pas cochée, l'interrupteur deviendra invisible lorsqu'il sera neutralisé.

#### Bouton « Hint »

Ce bouton permet de définir les paramètres de « Hint » de l'objet. Le texte « Hint » que l'on définit est le texte qui apparaît sur fond jaune lorsque l'on passe le curseur de la souris sur l'objet.

Quand on clique sur le bouton « Hint », la fenêtre suivante s'ouvre :



Cette fenêtre présente deux contrôles :

#### Option « Afficher le Hint »

Si cette option n'est pas cochée, le texte du Hint n'apparaîtra pas quand le curseur de la souris passera sur l'objet même si celui-ci est défini. Il faut donc cocher cette option si l'on veut utiliser le « Hint » de l'objet.

#### Fenêtre de saisie « Hint »

Cette fenêtre permet d'entrer le texte du Hint. Ce texte peut être un paramètre (exemple {1}).

### 3.2.6.10.2 Levier sans couleur

L'objet Levier sans couleur simule un interrupteur. L'apparence de cet objet varie avec un clic sur le bouton gauche de la souris au-dessus de l'objet ou en fonction de la valeur de sa variable de contrôle. Cette variable peut être de type booléen (flag L, Interrupteur logique SA...) ou entier (Registre R...).

Image du levier :



La boîte de dialogue est commune à tous les interrupteurs, veuillez vous référer à « Boîte de dialogue [Bouton Marche/Arrêt](#) <sup>233</sup> »

### 3.2.6.10.3 Bouton d'arrêt d'urgence

L'objet Bouton type arrêt d'urgence simule un interrupteur.

L'apparence de cet objet varie avec un clic sur le bouton gauche de la souris au-dessus de l'objet ou en fonction de la valeur de sa variable de contrôle. Cette variable peut être de type booléen (flag L, Interrupteur logique SA...) ou entier (Registre R...).

Image du bouton type arrêt d'urgence :



La boîte de dialogue est commune à tous les interrupteurs, veuillez vous référer à « Boîte de dialogue [Bouton Marche/Arrêt](#) <sup>233</sup> »

### 3.2.6.11 Objets de l'onglet SELECTEURS

#### 3.2.6.11.1 Sélecteur 2/3 positions

L'objet Sélecteur 2 / 3 positions simule un sélecteur pouvant avoir 2 ou 3 états.

L'apparence de cet objet varie avec un clic sur le bouton gauche de la souris au-dessus de l'objet ou en fonction de la valeur de sa variable de contrôle. Il permet de faire évoluer la variable de 0 à 2 avec un cycle de positions : 0 – 1 – 2 – 1 – 0.

*Exemple de Sélecteur 3 positions avec étiquette:*



*Exemple de Sélecteur 3 positions sans étiquette:*



**Boîte de dialogue « Sélecteur 2 / 3 positions »**

Les paramètres sont les suivants :

### Expression

Permet de définir le nom de la variable de contrôle de l'objet (variable associée à l'objet Sélecteur 3 positions). Cette variable sert d'index pour chaque état de l'animation (un état correspond à une valeur de la variable).

Indiquer dans ce champ la variable automate à utiliser (de type numérique entier)

*Exemple : R20*

Il est possible de spécifier l'adresse IP distante du noyau APIGRAF PLC dans l'expression. Si c'est le cas, c'est cette adresse qui sera prise en compte pour le rafraîchissement.

*Exemple : !193.0.1.1 !R20*

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

### Etiquette

Permet de mettre une légende au-dessus de l'objet sur une ou deux lignes. Il est possible de spécifier une police de caractères pour chaque ligne/

### Option 2 positions

Si cette options est coché le sélecteur aura 2 positions.

### Option 3 positions

Si cette options est coché le sélecteur aura 3 positions.

### Expression de neutralisation

Permet de spécifier la variable ou l'expression de neutralisation (Celle-ci doit obligatoirement être de type booléen).

*Exemple : SA10*

Il est possible de spécifier l'adresse IP distante du noyau APIGRAF PLC dans l'expression. Si c'est le cas, c'est cette adresse qui sera prise en compte pour le rafraîchissement.

*Exemple : !193.0.1.1 !SA10*

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

#### Option « visible si neutralisé »

Si cette option n'est pas cochée, le sélecteur deviendra invisible lorsqu'il sera neutralisé.

#### Bouton « Hint »

Ce bouton permet de définir les paramètres de « Hint » de l'objet. Le texte « Hint » que l'on définit est le texte qui apparaît sur fond jaune lorsque l'on passe le curseur de la souris sur l'objet.

Quand on clique sur le bouton « Hint », le fenêtre suivante s'ouvre :



Cette fenêtre présente deux contrôles :

#### Option « Afficher le Hint »

Si cette option n'est pas cochée, le texte du Hint n'apparaîtra pas quand le curseur de la souris passera sur l'objet même si celui-ci est défini. Il faut donc cocher cette option si l'on veut utiliser le « Hint » de l'objet.

#### Fenêtre de saisie « Hint »

Cette fenêtre permet d'entrer le texte du Hint. Ce texte peut être un paramètre (exemple {1}).

### 3.2.6.11.2 Sélecteur 2/3 positions sans étiquette

L'objet Sélecteur sans étiquette 2 / 3 positions simule un sélecteur pouvant avoir 2 ou 3 états. C'est le même objet que le Sélecteur 2 / 3 positions mais il ne possède pas d'étiquette.

Voir : [Sélecteur 2/3 positions](#) 

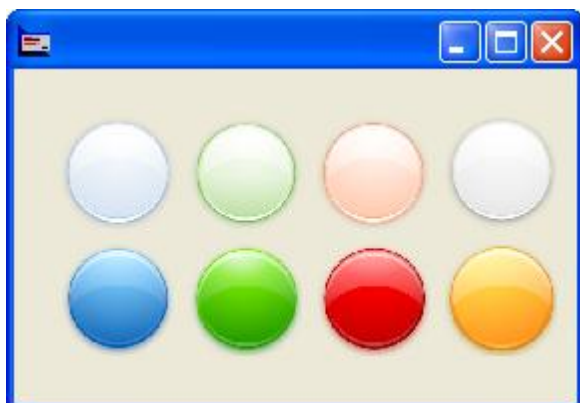
### 3.2.6.12 Objets de l'onglet VOYANTS

#### 3.2.6.12.1 Voyant type 1

Cet objet simule un voyant 2 états : blanc au repos, couleur vive en fonctionnement (couleur à définir : bleu, vert, rouge ou jaune).

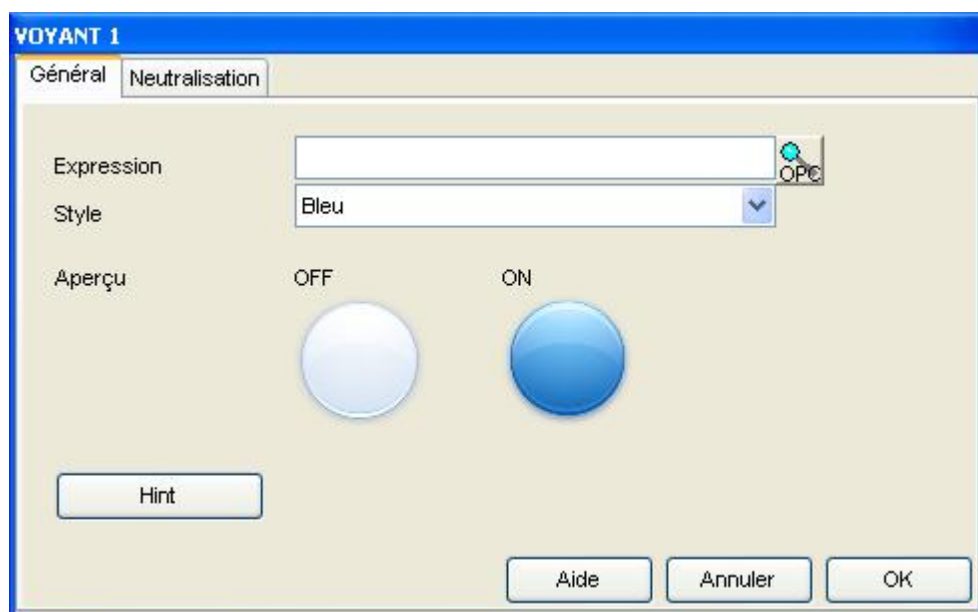
Son état varie en fonction de sa variable de contrôle ( 0 / 1 ).

Exemple de Voyant type 1 :



**Boîte de dialogue « Voyant type 1 » - Onglet général :**





Les paramètres sont les suivants:

### Expression

Indiquer dans ce champ le nom de la variable à visualiser avec le voyant. Cette variable doit être de type booléen (c'est à dire à deux états 0 et 1).

*Exemple* : SA10

Il est possible de spécifier l'adresse IP distante du noyau OPTIMA PLC ou APIGRAF PLC dans l'expression. Si c'est le cas, c'est cette adresse qui sera prise en compte pour le rafraîchissement.

*Exemple* : !193.0.1.1 !R12

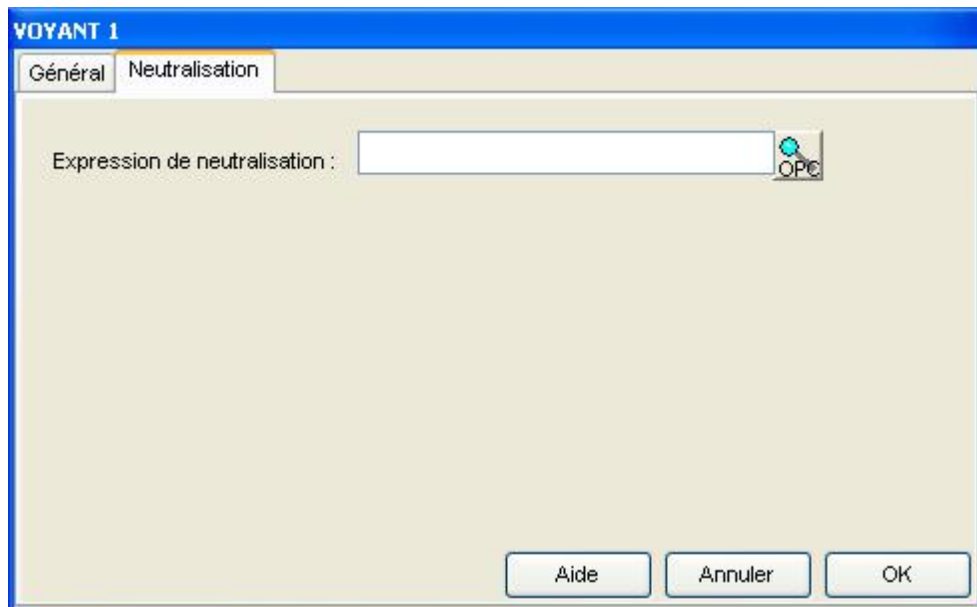
Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

Si la variable provient d'un serveur OPC, le bouton "OPC" permet d'explorer le serveur OPC et de trouver la variable à superviser dans un explorateur.

### Style

Permet de définir l'apparence du voyant : Bleu, vert, rouge ou jaune.

**Boîte de dialogue « Voyant type 1 » - Onglet Neutralisation:**



### Expression de neutralisation

Permet de spécifier la variable ou l'expression de neutralisation (Celle-ci doit obligatoirement être de type booléen).

*Exemple : SA10*

Il est possible de spécifier l'adresse IP distante du noyau OPTIMA PLC ou APIGRAF PLC dans l'expression. Si c'est le cas, c'est cette adresse qui sera prise en compte pour le rafraîchissement.

*Exemple : !193.0.1.1 !SA10*

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

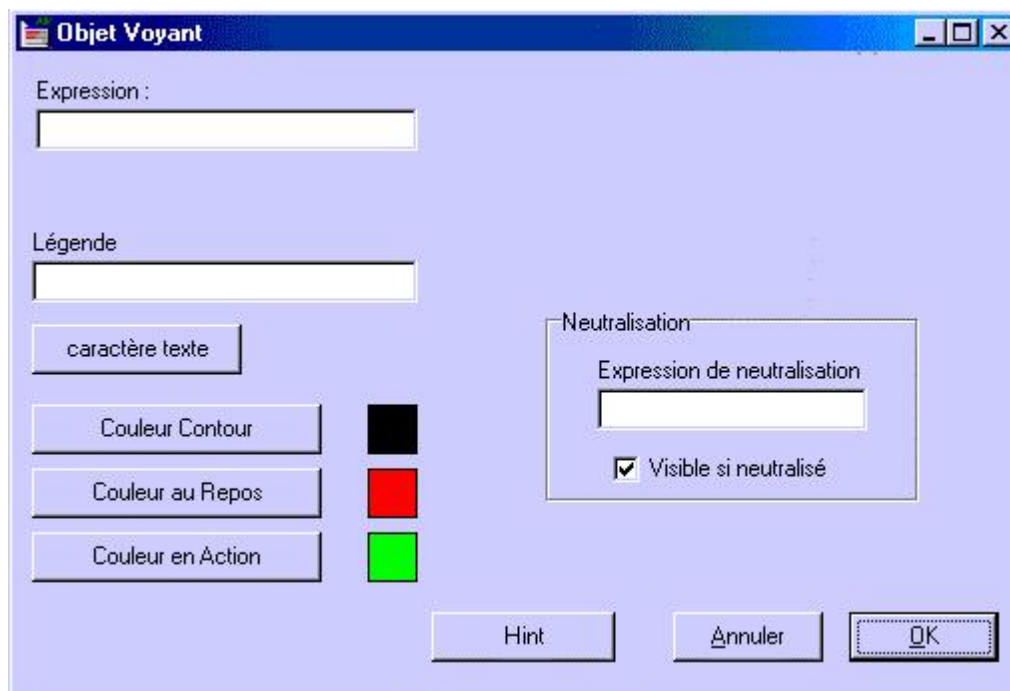
### 3.2.6.12.2 Voyant simple

Cet objet simule un voyant 2 états : rouge au repos, vert en fonctionnement (couleur par défaut). Son état varie en fonction de sa variable de contrôle (ayant 0 / 1 ou Faux / Vrai comme valeur).

Exemple de Voyant simple :



**Boîte de dialogue « Voyant simple » :**



Les paramètres sont les suivants:

### Expression

Indiquer dans ce champ le nom de la variable à visualiser avec le voyant. Cette variable doit être de type booléen (c'est à dire à deux états 0 et 1).

*Exemple : SA10*

Il est possible de spécifier l'adresse IP distante du noyau OPTIMA PLC ou APIGRAF PLC dans l'expression. Si c'est le cas, c'est cette adresse qui sera prise en compte pour le rafraîchissement.

*Exemple : !193.0.1.1 !R12*

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

Si la variable provient d'un serveur OPC, le bouton "OPC" permet d'explorer le serveur OPC et de trouver la variable à superviser dans un explorateur.

### Légende

Permet de mettre une légende au-dessus de l'objet. Ce paramètre n'accepte que du texte.

### Bouton « Caractères texte »

Permet l'accès à la boîte de dialogue concernant le type d'affichage de la légende (couleur, police...).

### Couleur Contour

Spécifie la couleur du contour du voyant.

### Couleur au Repos

Spécifie la couleur du voyant au repos (variable = 0 ou false).

### Couleur en Action

Spécifie la couleur du voyant en action (variable = 1 ou True).

### Expression de neutralisation

Permet de spécifier la variable ou l'expression de neutralisation (Celle-ci doit obligatoirement être de type booléen).

*Exemple : SA10*

Il est possible de spécifier l'adresse IP distante du noyau APIGRAF PLC dans l'expression. Si c'est le cas, c'est cette adresse qui sera prise en compte pour le rafraîchissement.

Exemple : !193.0.1.1 !SA10

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

#### Option « visible si neutralisé »

Si cette option n'est pas cochée, le voyant deviendra invisible lorsqu'il sera neutralisé.

#### Bouton « Hint »

Ce bouton permet de définir les paramètres de « Hint » de l'objet. Le texte « Hint » que l'on définit est le texte qui apparaît sur fond jaune lorsque l'on passe le curseur de la souris sur l'objet.

Quand on clique sur le bouton « Hint », la fenêtre suivante s'ouvre :



Cette fenêtre présente deux contrôles :

#### Option « Afficher le Hint »

Si cette option n'est pas cochée, le texte du Hint n'apparaîtra pas quand le curseur de la souris passera sur l'objet même si celui-ci est défini. Il faut donc cocher cette option si l'on veut utiliser le « Hint » de l'objet.

#### Fenêtre de saisie « Hint »

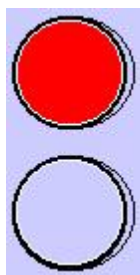
Cette fenêtre permet d'entrer le texte du Hint. Ce texte peut être un paramètre (exemple {1}).

### 3.2.6.12.3 Voyant double

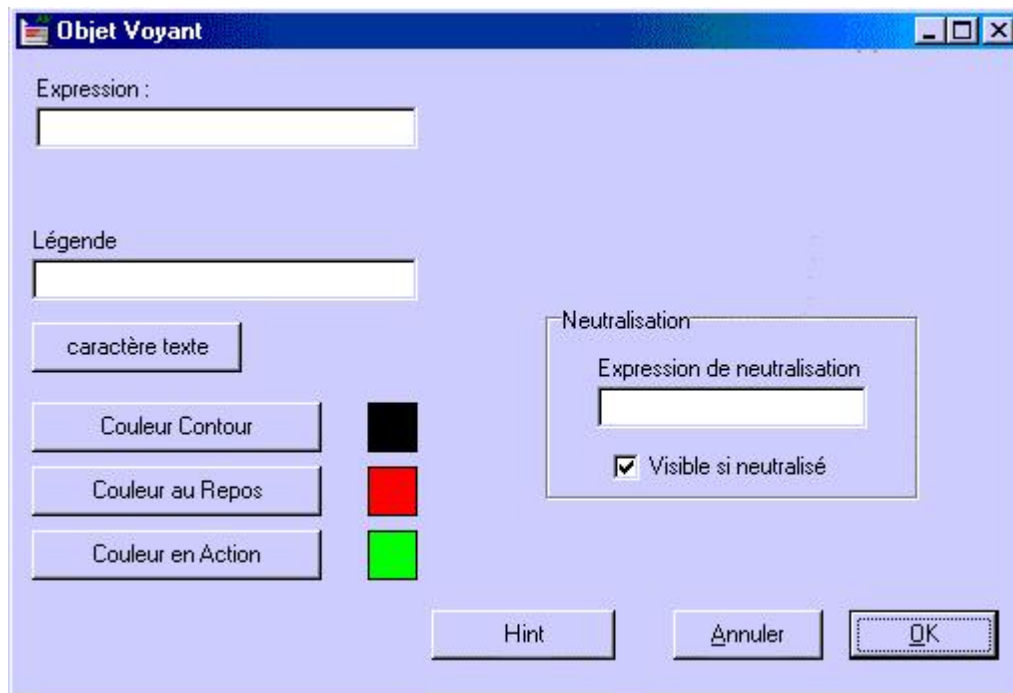
Cet objet simule deux voyant avec une seule variable : le premier est de couleur rouge au repos et gris en action, le second est de couleur grise au repos et vert en action (Couleurs par défaut).

Son état varie en fonction de sa variable de contrôle (ayant 0 / 1 ou Faux / Vrai comme valeur).

Exemple de Voyant double :



#### Boite de dialogue « Voyant double » :



Les paramètres sont les suivants:

#### **Expression**

Indiquer dans ce champ le nom de la variable à visualiser avec le voyant. Cette variable doit être de type numérique entier (Registre).

*Exemple* : R12

Il est possible de spécifier l'adresse IP distante du noyau APIGRAF PLC dans l'expression. Si c'est le cas, c'est cette adresse qui sera prise en compte pour le rafraîchissement.

*Exemple* : !193.0.1.1 !R12

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

#### **Légende**

Permet de mettre une légende au-dessus de l'objet. Ce paramètre n'accepte que du texte.

#### **Bouton « Caractères texte »**

Permet l'accès à la boîte de dialogue concernant le type d'affichage de la légende (couleur, police...).

#### **Couleur Contour**

Spécifie la couleur du contour du voyant.

#### **Couleur au Repos**

Spécifie la couleur du premier voyant au repos (variable = 0 ou false).

#### **Couleur en Action**

Spécifie la couleur du second voyant en action (variable = 1 ou True).

#### **Expression de neutralisation**

Permet de spécifier la variable ou l'expression de neutralisation (Celle-ci doit obligatoirement être de type booléen).

*Exemple* : SA10

Il est possible de spécifier l'adresse IP distante du noyau APIGRAF PLC dans l'expression. Si c'est le cas, c'est cette adresse qui sera prise en compte pour le rafraîchissement.

*Exemple* : !193.0.1.1 !SA10

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

#### Option « visible si neutralisé »

Si cette option n'est pas cochée, le voyant deviendra invisible lorsqu'il sera neutralisé.

#### Bouton « Hint »

Ce bouton permet de définir les paramètres de « Hint » de l'objet. Le texte « Hint » que l'on définit est le texte qui apparaît sur fond jaune lorsque l'on passe le curseur de la souris sur l'objet.

Quand on clique sur le bouton « Hint », la fenêtre suivante s'ouvre :



Cette fenêtre présente deux contrôles :

#### Option « Afficher le Hint »

Si cette option n'est pas cochée, le texte du Hint n'apparaîtra pas quand le curseur de la souris passera sur l'objet même si celui-ci est défini. Il faut donc cocher cette option si l'on veut utiliser le « Hint » de l'objet.

#### Fenêtre de saisie « Hint »

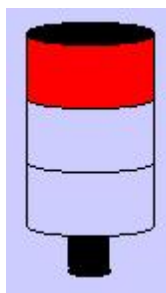
Cette fenêtre permet d'entrer le texte du Hint. Ce texte peut être un paramètre (exemple {1}).

### 3.2.6.12.4 Colonne 3 couleurs

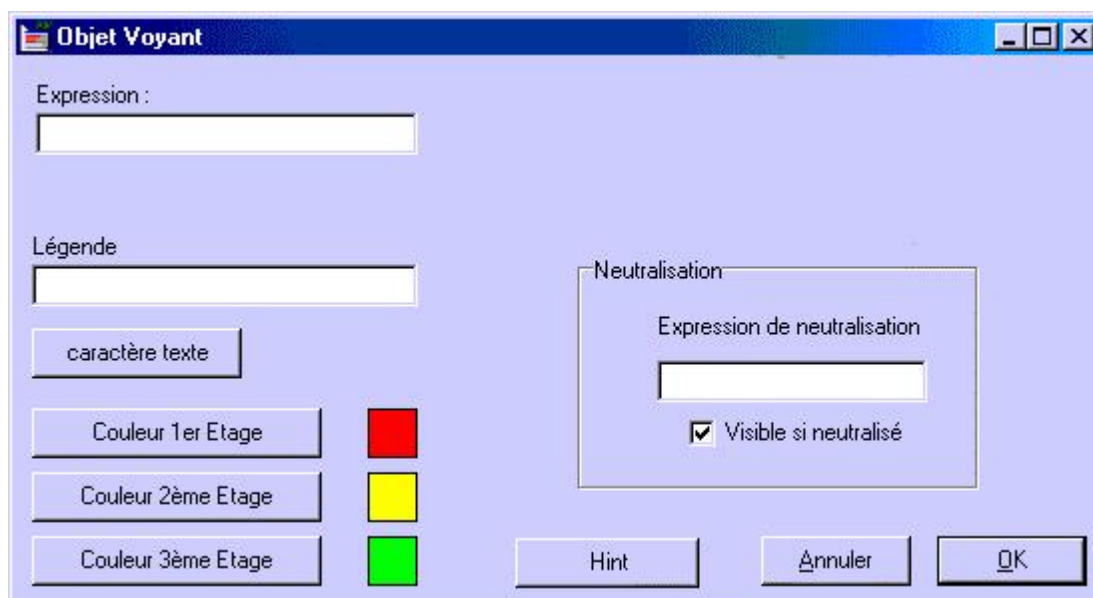
Cet objet simule une colonne constitué de trois niveaux contrôlés avec une seule variable : le premier étage est actif pour une valeur nulle de la variable, sa couleur par défaut est le rouge, le second est actif pour une valeur égal à 1, sa couleur est le jaune par défaut et le troisième étage est actif pour une valeur égal à 2, sa couleur est le vert par défaut.

Les étages non actif prennent sont grisés.

Exemple de Colonne 3 couleurs :



#### Boîte de dialogue de « Voyant double » :



Les paramètres sont les suivants:

### Expression

Indiquer dans ce champ le nom de la variable à visualiser avec la colonne. Cette variable doit être de type numérique entier (Registre).

*Exemple* : R12

Il est possible de spécifier l'adresse IP distante du noyau APIGRAF PLC dans l'expression. Si c'est le cas, c'est cette adresse qui sera prise en compte pour le rafraîchissement.

*Exemple* : !193.0.1.1 !R12

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

### Légende

Permet de mettre une légende au-dessus de l'objet. Ce paramètre n'accepte que du texte.

### Bouton « Caractères texte »

Permet l'accès à la boîte de dialogue concernant le type d'affichage de la légende (couleur, police...).

### Couleur 1er étage

Spécifie la couleur de la partie haute de la colonne, active quand la variable est nulle

### Couleur 2ème étage

Spécifie la couleur de la partie centrale de la colonne, active quand la variable est égale à 1.

### Couleur 3ème étage

Spécifie la couleur de la partie basse de la colonne, active quand la variable est égale à 2.

### Expression de neutralisation

Permet de spécifier la variable ou l'expression de neutralisation (Celle-ci doit obligatoirement être de type booléen).

*Exemple* : SA10

Il est possible de spécifier l'adresse IP distante du noyau APIGRAF PLC dans l'expression. Si c'est le cas, c'est cette adresse qui sera prise en compte pour le rafraîchissement.

*Exemple* : !193.0.1.1 !SA10

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

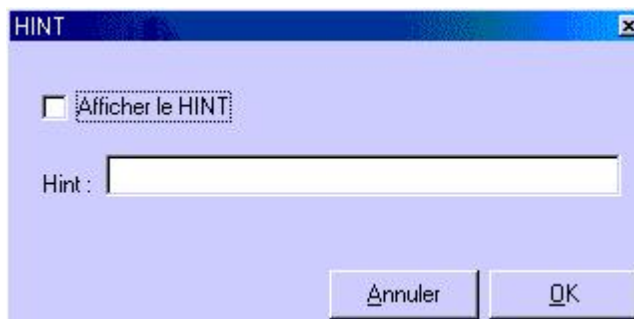
**Option « visible si neutralisé »**

Si cette option n'est pas cochée, la colonne deviendra invisible lorsqu'il sera neutralisé.

**Bouton « Hint »**

Ce bouton permet de définir les paramètres de « Hint » de l'objet. Le texte « Hint » que l'on définit est le texte qui apparaît sur fond jaune lorsque l'on passe le curseur de la souris sur l'objet.

Quand on clique sur le bouton « Hint », la fenêtre suivante s'ouvre :



Cette fenêtre présente deux contrôles :

**Option « Afficher le Hint »**

Si cette option n'est pas cochée, le texte du Hint n'apparaîtra pas quand le curseur de la souris passera sur l'objet même si celui-ci est défini. Il faut donc cocher cette option si l'on veut utiliser le « Hint » de l'objet.

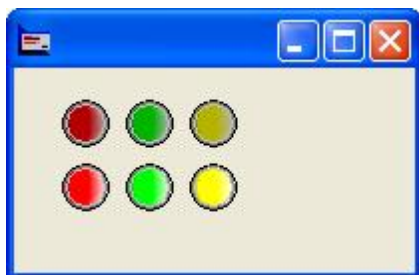
**Fenêtre de saisie « Hint »**

Cette fenêtre permet d'entrer le texte du Hint. Ce texte peut être un paramètre (exemple {1}).

**3.2.6.12.5 LED**

Cet objet est équivalent à un voyant mais avec l'apparence d'une diode électroluminescente éteinte ou allumé.

Exemples de Leds :

**Boîte de dialogue de « Led » :**



Les paramètres sont les suivants:

### Expression

Indiquer dans ce champ le nom de la variable à visualiser avec le voyant. Cette variable doit être de type booléen (c'est à dire à deux états 0 et 1).

*Exemple : SA10*

Il est possible de spécifier l'adresse IP distante du noyau OPTIMA PLC ou APIGRAF PLC dans l'expression. Si c'est le cas, c'est cette adresse qui sera prise en compte pour le rafraîchissement.

*Exemple : !193.0.1.1 !R12*

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

Si la variable provient d'un serveur OPC, le bouton "OPC" permet d'explorer le serveur OPC et de trouver la variable à superviser dans un explorateur.

### Légende

Permet de mettre une légende au-dessus de l'objet. Ce paramètre n'accepte que du texte.

### Bouton « Caractères texte »

Permet l'accès à la boîte de dialogue concernant le type d'affichage de la légende (couleur, police...).

### Expression de neutralisation

Permet de spécifier la variable ou l'expression de neutralisation (Celle-ci doit obligatoirement être de type booléen).

*Exemple : SA10*

Il est possible de spécifier l'adresse IP distante du noyau APIGRAF PLC dans l'expression. Si c'est le cas, c'est cette adresse qui sera prise en compte pour le rafraîchissement.

*Exemple : !193.0.1.1 !SA10*

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

### Option « visible si neutralisé »

Si cette option n'est pas cochée, la Led deviendra invisible lorsqu'il sera neutralisé.

### Bouton « Hint »

Ce bouton permet de définir les paramètres de « Hint » de l'objet. Le texte « Hint » que l'on définit est le texte qui

apparaît sur fond jaune lorsque l'on passe le curseur de la souris sur l'objet.

Quand on clique sur le bouton « Hint », la fenêtre suivante s'ouvre :



Cette fenêtre présente deux contrôles :

#### Option « Afficher le Hint »

Si cette option n'est pas cochée, le texte du Hint n'apparaîtra pas quand le curseur de la souris passera sur l'objet même si celui-ci est défini. Il faut donc cocher cette option si l'on veut utiliser le « Hint » de l'objet.

#### Fenêtre de saisie « Hint »

Cette fenêtre permet d'entrer le texte du Hint. Ce texte peut être un paramètre (exemple {1}).

### 3.2.6.13 Objets de l'onglet EXCEL

#### 3.2.6.13.1 Liaison EXCEL

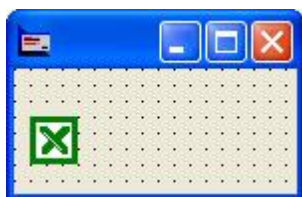
L'objet Liaison EXCEL permet de lire et d'écrire des variables automates depuis/vers un classeur EXCEL.

La mise à jour est déclenché par une variable automate booléenne. Quand la valeur de cette variable passe à 1, le fichier EXCEL est ouvert, le feuille de calcul est sélectionnée et les données sont mise à jour en lecture ou en écriture. La valeur de la variable de déclenchement est ensuite remise à 0.

Le principe est le suivant : On associe des variables automates provenant d'Optima PLC/ APIGRAF ou d'une liaison OPC à une cellule d'un formulaire EXCEL et on spécifie pour chaque lien si celui-ci est en écriture ou en lecture. Si le lien est en lecture alors la valeur sera lue dans le cellule EXCEL et sa valeur sera affectée à la variable automate correspondante. Si le lien est en écriture alors la valeur sera lue dans l'automate et cette valeur sera affectée à la cellule EXCEL correspondante.

Toutes les liaisons en écriture ou en lecture sont exécutées une seule fois dès que la valeur de la variable de déclenchement passe à 1.

L'objet liaison EXCEL n'est visible qu'en mode conception sur la page de supervision :



#### Boîte de dialogue « Liaison EXCEL » :

**Liaison Excel**

Nom ou expression du fichier EXCEL :

( Si la valeur entrée n'est pas une variable, entrer le nom du fichier entre côtes ' ' : Exemple : 'C:\TEST.XLS' )

Nom ou expression du classeur EXCEL :

( Si la valeur entrée n'est pas une variable, entrer le nom du fichier entre côtes ' ' : Exemple : 'Feuil1' )

Expression déclenchement liaison EXCEL :

☒ Application EXCEL visible :

☐ Fermer EXCEL à la fin du transfert

Les paramètres sont :

- **Nom ou expression du fichier EXCEL**

Spécifie le nom du fichier EXCEL à utiliser. Si le nom du fichier est fixé, entrer le nom du fichier entre cotes :

Exemple : 'C:\REPEXCEL\TEST.XLS'

Si le nom du fichier est variable, entrer le nom de la variable ( de type string/chaîne) dont la valeur représente le nom du fichier EXCEL à utiliser

Exemple : C10

- **Nom ou expression du classeur EXCEL**

Spécifie le nom du classeur EXCEL à utiliser dans le fichier EXCEL courant. Si le nom du classeur est fixé, entrer le nom du classeur entre cotes :

Exemple : 'Feuil1'

Si le nom du classeur est variable, entrer le nom de la variable ( de type string/chaîne) dont la valeur représente le nom du classeur EXCEL à utiliser

Exemple : C10

La valeur par défaut de ce champ est : 'Feuil1'

- **Expression déclenchement liaison EXCEL**

Spécifie la variable automate qui va déclencher la liaison EXCEL et donc la mise à jour des données en lecture ou écriture.

La variable doit être booléenne et le déclenchement sera effectué lors du passage à 1 de la valeur de la variable. La valeur de la variable sera remise à 0 à la fin des mises à jour.

- **Application EXCEL visible**

Spécifie si l'application EXCEL doit être visible à l'écran lors de la liaison.

- **Fermer EXCEL à la fin du transfert**

Spécifie si l'application EXCEL doit être fermée une fois les mise à jour effectuées.

### • Variable Index Ligne Cellules

Ce champ est facultatif, Il spécifie une variable numérique qui sera utilisé pour mettre un numéro de ligne à une case EXCEL définie dans la liste des liens en lecture/écriture.

Si une cellule est définie dans la liste des liens de manière complète, Exemple : A11 alors la variable index ne sera pas prise en compte. Par contre, si la cellule est définie uniquement avec une lettre, exemple : B alors la valeur de la variable sera ajouté pour former le nom complet de la cellule. Si la variable vaut 2 alors la cellule sera B2.

Il est possible de spécifier des noms de cellules complets (B2) et des noms de cellule incomplets ( B ) dans la liste des liens en lecture/écriture.

### • Bouton "Liste des liens en lecture/écriture"

Permet de définir la liste des liaisons. Le bouton ouvre la fenêtre suivante :

Variable Automate	Cellule EXCEL	Lecture/Ecriture
WR100	A4	Ecriture
WR100+10	A5	Ecriture
WR100+20	B2	Ecriture

La fenêtre affiche la liste des liens.

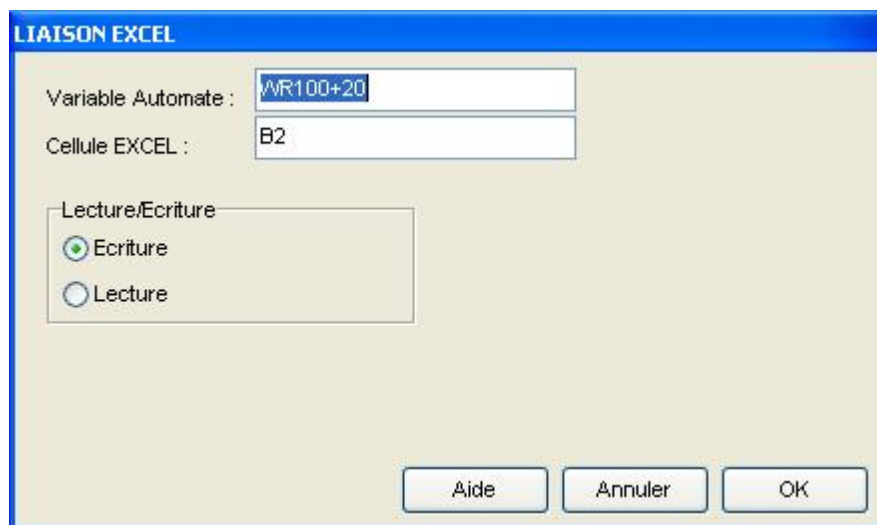
**Pour ajouter un lien**, cliquer sur le bouton "Ajouter", le fenêtre de paramétrage suivante apparaît :

Renseigner les champs suivants : Variable automate ( Exemple : R100 ), Cellule EXCEL ( Exemple : A4 ) et choisir le type de liaison ( Ecriture ou Lecture).

Valider ensuite avec "OK"

**Pour supprimer un lien**, sélectionner le lien à supprimer dans la liste à l'aide de la souris, celui-ci doit alors apparaître en surbrillance dans la liste et cliquer ensuite sur le bouton "Supprimer"

**Pour modifier un lien**, sélectionner le lien à modifier dans la liste à l'aide de la souris, celui-ci doit alors apparaître en surbrillance dans la liste et cliquer ensuite sur le bouton "Modifier". La fenêtre de paramétrage s'ouvre alors :



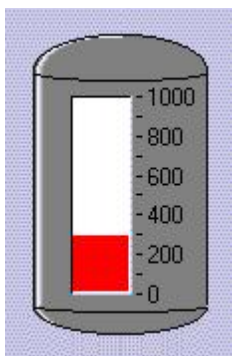
Modifier les champs et valider avec "OK".

### 3.2.6.14 Objets de l'onglet CUBE / VANNE

#### 3.2.6.14.1 Cuve

L'objet cuve permet d'afficher une cuve avec une jauge associée qui symbolise le niveau de la cuve, il est de plus possible d'afficher sur la cuve une échelle graduée.

Exemple de cuve :



**Boîte de dialogue « Cuve » :**

Les paramètres sont :

#### Expression

Indiquer dans ce champ le nom de la variable à visualiser dans la cuve. Cette variable doit être de type numérique entier (Registre).

*Exemple* : R12

Il est possible de spécifier l'adresse IP distante du noyau APIGRAF PLC dans l'expression. Si c'est le cas, c'est cette adresse qui sera prise en compte pour le rafraîchissement.

*Exemple* : !193.0.1.1 !R12

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

#### Valeur Min

Cette option définit la valeur pour laquelle la cuve est vide.

Il est possible de spécifier dans ce champ une valeur constante ou l'expression d'une variable automate :

*Exemple* : 1000  
R20

Pour les variables automates, on peut de plus spécifier l'adresse IP distante du noyau APIGRAF PLC dans l'expression. Si c'est le cas, c'est cette adresse qui sera prise en compte pour le rafraîchissement de la valeur Min.

*Exemple* : !193.0.1.1 !R12

Dans ce cas, la borne min de la cuve sera variable.

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

#### Valeur Max

Cette option définit la valeur pour laquelle la cuve est pleine.

Ce champ offre les mêmes options d'utilisation que le champ « Valeur Min » ci dessus.

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

**Couleur de la cuve**

Spécifie la couleur de la cuve.

**Couleur de la jauge**

Spécifie la couleur de la jauge associée à la cuve. Il s'agit de la couleur indiquant le niveau de la cuve.

**Couleur de fond jauge**

Spécifie la couleur de fond de la jauge associée à la cuve.

**Option « Afficher l'échelle »**

Si cette option est cochée, une échelle graduée sera visible à côté de la jauge associée à la cuve.

**Echelle Min**

Ce paramètre indique la valeur de la borne inférieure de l'échelle graduée de la jauge associée à la cuve. Ce paramètre constant peut être de type entier ou décimal

*Exemple :* 10  
0,5

**Echelle Max**

Ce paramètre indique la valeur de la borne supérieure de l'échelle graduée de la jauge associée à la cuve. Ce paramètre constant peut être de type entier ou décimal.

*Exemple :* 100  
10,05

**Intervalle graduation**

Ce paramètre indique la valeur séparant deux graduations. Ce paramètre constant peut être entier ou décimal.

Exemple : 10  
1,5

**Nombre de décimales**

Les graduations peuvent être affichées avec des décimales. Ce paramètre indique le nombre de décimales affichées. Ce paramètre est de type entier.

**Expression de neutralisation**

Permet de spécifier la variable ou l'expression de neutralisation (Celle-ci doit obligatoirement être de type booléen). L'objet cuve sera invisible chaque fois que la variable ou l'expression sera égale à true.

*Exemple :* SA10

Il est possible de spécifier l'adresse IP distante du noyau APIGRAF PLC dans l'expression. Si c'est le cas, c'est cette adresse qui sera prise en compte pour le rafraîchissement.

*Exemple :* !193.0.1.1 !SA10

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

**Bouton « Hint »**

Ce bouton permet de définir les paramètres de « Hint » de l'objet. Le texte « Hint » que l'on définit est le texte qui apparaît sur fond jaune lorsque l'on passe le curseur de la souris sur l'objet.

Quand on clique sur le bouton « Hint », la fenêtre suivante s'ouvre :



Cette fenêtre présente deux contrôles :

#### Option « Afficher le Hint »

Si cette option n'est pas cochée, le texte du Hint n'apparaîtra pas quand le curseur de la souris passera sur l'objet même si celui-ci est défini. Il faut donc cocher cette option si l'on veut utiliser le « Hint » de l'objet.

#### Fenêtre de saisie « Hint »

Cette fenêtre permet d'entrer le texte du Hint. Ce texte peut être un paramètre (exemple {1}).

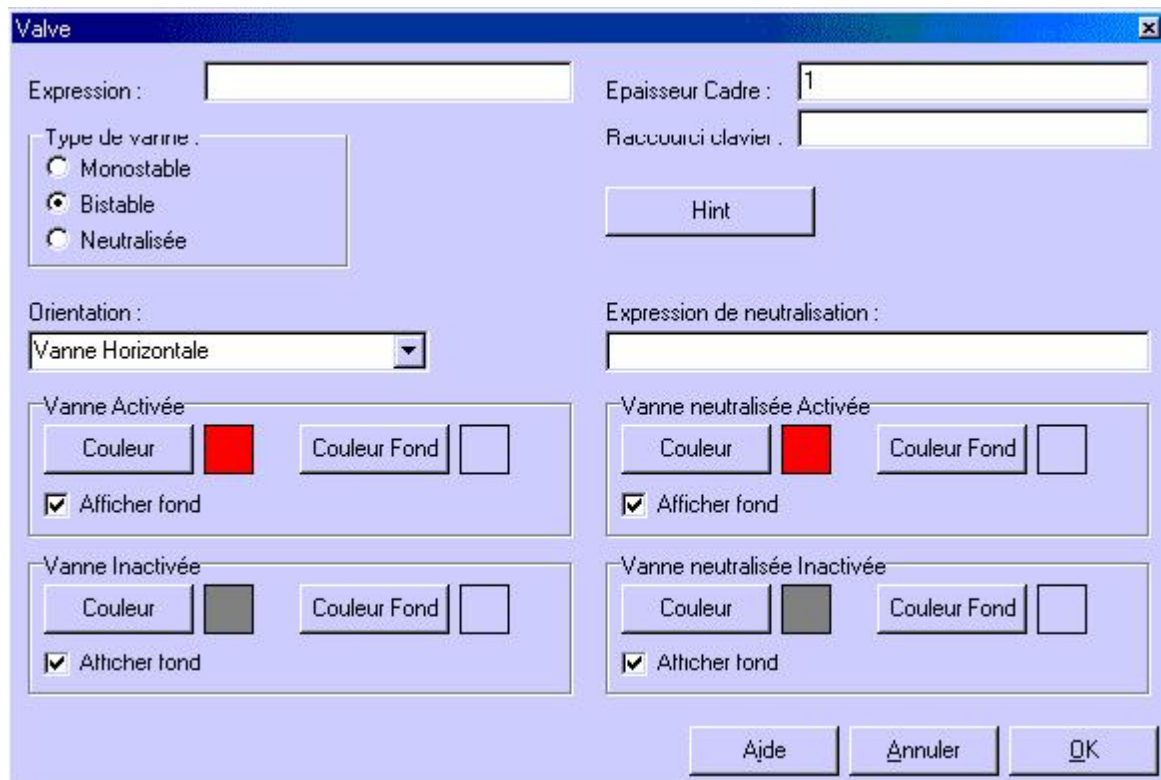
### 3.2.6.14.2 Vanne

L'objet vanne permet d'afficher une vanne sur une page de supervision. Cette vanne peut contrôler une variable automate de type booléen. Elle peut de plus être neutralisée par une variable automate de type booléen.

Exemple de vanne :



**Boîte de dialogue « Vanne » :**



Les champs sont les suivants :

#### Expression



Cette fenêtre peut contenir le nom d'une variable automate de type booléen. Si la variable change d'état, la vanne change d'apparence en conséquence. De même lorsque l'utilisateur presse l'électrovanne, la variable automate est forcée à 1.

Exemple : SA10

Il est possible de spécifier l'adresse IP distante du noyau APIGRAF PLC dans l'expression. Si c'est le cas, c'est cette adresse qui sera prise en compte pour le rafraîchissement.

Exemple : !193.0.1.1 !SA10

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

### Type de vanne (Monostable/Bistable/Neutralisée)

Cette option permet d'indiquer si l'électrovanne est monostable ou bistable. Si elle est monostable, elle revient lorsque l'on cesse de la presser. Si elle est bistable, il faut cliquer deux fois pour la relâcher.

Si l'électrovanne est neutralisée, aucune action n'est possible dessus, elle sert alors uniquement à la visualisation (Cette neutralisation est permanente : voir plus loin pour une neutralisation temporaire).

### Epaisseur cadre

L'épaisseur correspond à la largeur du cadre en pixels. Plus cette valeur est grande, plus le cadre de l'électrovanne semble épais.

### Raccourci clavier

Cette option permet de spécifier une touche dont l'utilisation permet d'activer directement le bouton APIGRAF. Les valeurs acceptées dans ce champ sont :

Lettres de A à Z, les majuscules et les minuscules n'étant pas différenciées.  
Les chiffres de 0 à 9

*Attention à l'utilisation conjointe de fenêtres de saisie et de touche de raccourci de type alphanumérique sur des boutons ! !. En effet si lors d'une saisie, on tape le chiffre 9 et que dans la même page se trouve un bouton avec « 9 » pour raccourci, le bouton sera enclenché.*

Les touches de fonctions de F2 à F12 (La touche F1 étant réservée pour l'aide Windows®)  
INSERT, SUPPR, ESCAPE, BACK

Les combinaisons de touches avec ALT : ALT A, ALT B, ALT C, ..... , ALT Z, ALT 0,... ALT 9

### Bouton « Hint »

Ce bouton permet de définir les paramètres de « Hint » de l'objet. Le texte « Hint » que l'on définit est le texte qui apparaît sur fond jaune lorsque l'on passe le curseur de la souris sur l'objet.

Quand on clique sur le bouton « Hint », le fenêtre suivante s'ouvre :



Cette fenêtre présente deux contrôles :

### Option « Afficher le Hint »

Si cette option n'est pas cochée, le texte du Hint n'apparaîtra pas quand le curseur de la souris passera sur l'objet même si celui-ci est défini. Il faut donc cocher cette option si l'on veut utiliser le « Hint » de l'objet.

### Fenêtre de saisie « Hint »

Cette fenêtre permet d'entrer le texte du Hint. Ce texte peut être un paramètre (exemple {1}).

### **Vanne Horizontale / Vanne Verticale**

Permet de définir l'orientation du dessin de la vanne.

### **Zones Vanne Activée/ Vanne Inactivée**

Ces deux zones permettent de définir les couleurs et le type d'affichage selon que la vanne est inactive (variable automate à 0) ou active (variable automate à 1) dans le cas où la vanne n'est pas neutralisée.

Les couleurs de dessin et de fond peuvent y être définies ainsi que la nécessité d'afficher ou non le fond.

### **Expression de neutralisation**

Cette fenêtre peut contenir le nom d'une variable automate de type booléen. Si la variable est égale à 1, la vanne ne peut plus être actionnée par l'utilisateur. La vanne peut être de nouveau activée par l'utilisateur.

Exemple : SA20

Il est possible de spécifier l'adresse IP distante du noyau APIGRAF PLC dans l'expression. Si c'est le cas, c'est cette adresse qui sera prise en compte pour le rafraîchissement.

Exemple : !193.0.1.1 !SA20

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

### **Zones Vanne Neutralisée Activée/ Vanne Neutralisée Inactivée**

Ces deux zones permettent de définir les couleurs et le type d'affichage selon que la vanne est inactive (variable automate à 0) ou active (variable automate à 1) dans le cas où la vanne est neutralisée.

Une vanne est neutralisée si elle a été définie comme neutralisée au niveau du type de vanne ou si la variable de neutralisation saisie dans l'expression de neutralisation est égale à 1.

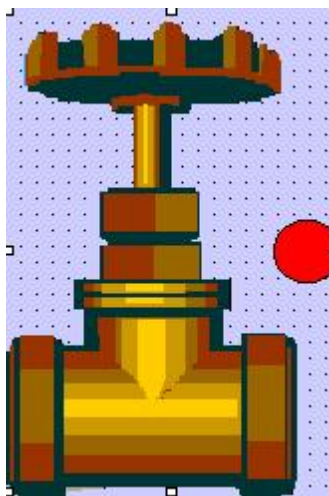
Les couleurs de dessin et de fond peuvent y être définies ainsi que la nécessité d'afficher ou non le fond.

## **3.2.6.14.3 Vanne bitmap**

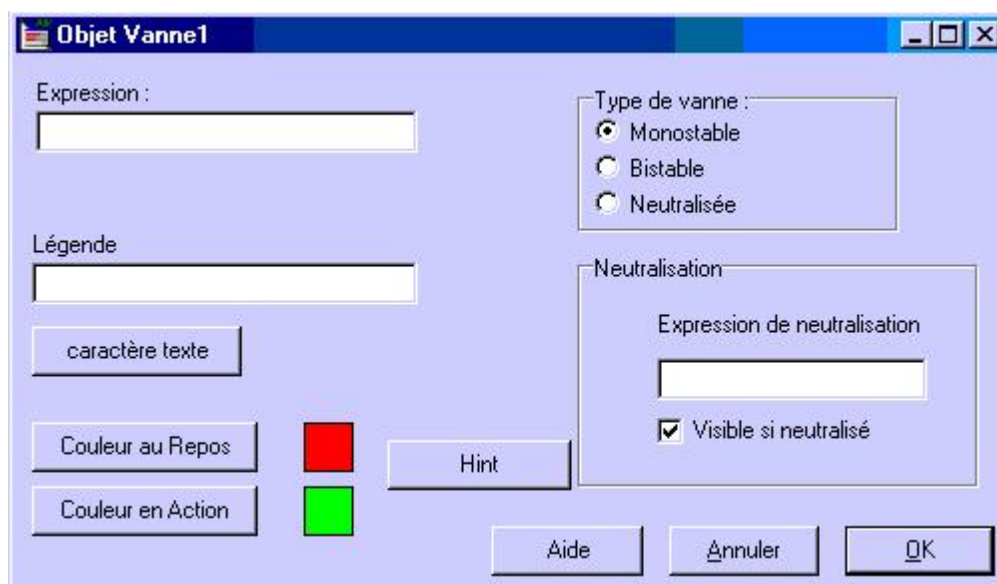
L'objet vanne bitmap permet d'afficher une vanne sur une page de supervision. Le voyant de droite de celle-ci indique si elle est actionnée ou non (vert en action et rouge au repos par défaut).

Cette vanne peut contrôler une variable automate de type booléen. Elle peut de plus être neutralisée par une variable automate de type booléen.

Exemple de vanne :



**Boîte de dialogue « Vanne » :**



Les champs sont les suivants :

### Expression

Cette fenêtre peut contenir le nom d'une variable automate de type booléen. Si la variable change d'état, la vanne change d'apparence en conséquence. De même lorsque l'utilisateur presse l'électrovanne, la variable automate est forcée à 1.

Exemple : SA10

Il est possible de spécifier l'adresse IP distante du noyau APIGRAF PLC dans l'expression. Si c'est le cas, c'est cette adresse qui sera prise en compte pour le rafraîchissement.

Exemple : !193.0.1.1 !SA10

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

### Type de vanne (Monostable/Bistable/Neutralisée)

Cette option permet d'indiquer si l'électrovanne est monostable ou bistable. Si elle est monostable, elle revient lorsque l'on cesse de la presser. Si elle est bistable, il faut cliquer deux fois pour la relâcher.

Si l'électrovanne est neutralisée, aucune action n'est possible dessus, elle sert alors uniquement à la visualisation (Cette neutralisation est permanente : voir plus loin pour une neutralisation temporaire).

### Couleur au Repos

Spécifie la couleur du voyant (à droite de la vanne) au repos (variable = 0 ou False).

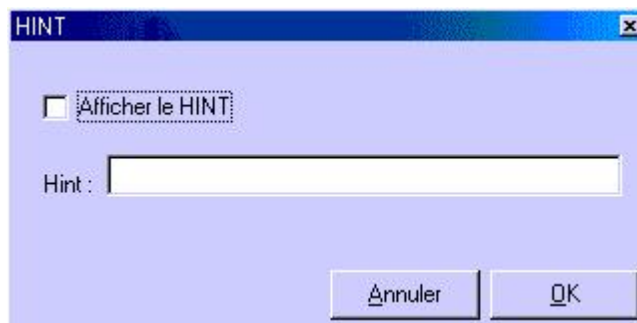
### Couleur en Action

Spécifie la couleur du voyant (à droite de la vanne) en action (variable = 1 ou True).

### Bouton « Hint »

Ce bouton permet de définir les paramètres de « Hint » de l'objet. Le texte « Hint » que l'on définit est le texte qui apparaît sur fond jaune lorsque l'on passe le curseur de la souris sur l'objet.

Quand on clique sur le bouton « Hint », le fenêtre suivante s'ouvre :



Cette fenêtre présente deux contrôles :

#### Option « Afficher le Hint »

Si cette option n'est pas cochée, le texte du Hint n'apparaîtra pas quand le curseur de la souris passera sur l'objet même si celui-ci est défini. Il faut donc cocher cette option si l'on veut utiliser le « Hint » de l'objet.

#### Fenêtre de saisie « Hint »

Cette fenêtre permet d'entrer le texte du Hint. Ce texte peut être un paramètre (exemple {1}).

#### Expression de neutralisation

Cette fenêtre peut contenir le nom d'une variable automate de type booléen. Si la variable est égale à 1, la vanne ne peut plus être actionnée par l'utilisateur. La vanne peut être de nouveau activée par l'utilisateur.

Exemple : SA20

Il est possible de spécifier l'adresse IP distante du noyau APIGRAF PLC dans l'expression. Si c'est le cas, c'est cette adresse qui sera prise en compte pour le rafraîchissement.

Exemple : !193.0.1.1 !SA20

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

#### Option « visible si neutralisé »

Si cette option n'est pas cochée, l'objet deviendra invisible lorsqu'il sera neutralisé.

### 3.2.6.15 Objets de l'onglet Heure/Date

#### 3.2.6.15.1 Affichage Heure

Cet objet permet d'afficher l'heure sur une page de supervision.

#### Boîte de dialogue "Affichage Heure"



Les paramètres sont :

- **Caractères**

Permet de définir la police à utiliser pour afficher l'heure (Police, taille, couleur)

- **Hint**

Ce bouton permet de définir les paramètres de « Hint » de l'objet. Le texte « Hint » que l'on définit est le texte qui apparaît sur fond jaune lorsque l'on passe le curseur de la souris sur l'objet.

Quand on clique sur le bouton « Hint », la fenêtre suivante s'ouvre :



Cette fenêtre présente deux contrôles :

**Option « Afficher le Hint »**

Si cette option n'est pas cochée, le texte du Hint n'apparaîtra pas quand le curseur de la souris passera sur l'objet même si celui-ci est défini. Il faut donc cocher cette option si l'on veut utiliser le « Hint » de l'objet.

**Fenêtre de saisie « Hint »**

Cette fenêtre permet d'entrer le texte du Hint. Ce texte peut être un paramètre (exemple {1}).

- **Fond Transparent**

Quand cette option est cochée, l'affichage de l'heure est effectué sans fond

- **Couleur de fond**

Spécifier ici la couleur de fond du texte

- **Expression de neutralisation**

Permet de spécifier la variable ou l'expression de neutralisation (Celle-ci doit obligatoirement être de type booléen). L'objet "Affichage Heure" sera invisible chaque fois que la variable ou l'expression sera égale à true.

*Exemple : SA10*

Il est possible de spécifier l'adresse IP distante du noyau APIGRAF PLC dans l'expression. Si c'est le cas, c'est cette adresse qui sera prise en compte pour le rafraîchissement.

*Exemple : !193.0.1.1 !SA10*

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

### 3.2.6.15.2 Affichage Date

Cet objet permet d'afficher la date courante sur une page de supervision.

**Boîte de dialogue "Affichage Date"**



Les paramètres sont :

- **Caractères**

Permet de définir la police à utiliser pour afficher la date (Police, taille, couleur)

- **Hint**

Ce bouton permet de définir les paramètres de « Hint » de l'objet. Le texte « Hint » que l'on définit est le texte qui apparaît sur fond jaune lorsque l'on passe le curseur de la souris sur l'objet.

Quand on clique sur le bouton « Hint », le fenêtre suivante s'ouvre :



Cette fenêtre présente deux contrôles :

**Option « Afficher le Hint »**

Si cette option n'est pas cochée, le texte du Hint n'apparaîtra pas quand le curseur de la souris passera sur l'objet même si celui-ci est défini. Il faut donc cocher cette option si l'on veut utiliser le « Hint » de l'objet.

**Fenêtre de saisie « Hint »**

Cette fenêtre permet d'entrer le texte du Hint. Ce texte peut être un paramètre (exemple {1}).

- **Fond Transparent**

Quand cette option est cochée, l'affichage de la date est effectué sans fond

- **Couleur de fond**

Spécifier ici la couleur de fond du texte

- **Expression de neutralisation**

Permet de spécifier la variable ou l'expression de neutralisation (Celle-ci doit obligatoirement être de type booléen). L'objet "Affichage Date" sera invisible chaque fois que la variable ou l'expression sera égale à true.

Exemple : SA10

Il est possible de spécifier l'adresse IP distante du noyau APIGRAF PLC dans l'expression. Si c'est le cas, c'est cette adresse qui sera prise en compte pour le rafraîchissement.

Exemple : !193.0.1.1 !SA10

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

### 3.2.6.15.3 Calendrier

Cet objet affiche un calendrier et permet de sélectionner une date à affecter à une variable de type chaîne.

#### Boîte de dialogue "Calendrier"

Les paramètres sont :

- **Expression**

Indiquer dans ce champ le nom d'une la variable de **type chaîne**.

Cette variable sera affectée par la date sélectionnée dans l'objet calendrier. Si la date contenue dans la variable change et est valide alors le jour courant du calendrier sera également modifié

Exemple : C10

Il est possible de spécifier l'adresse IP distante du noyau APIGRAF PLC dans l'expression. Si c'est le cas, c'est cette adresse qui sera prise en compte pour le rafraîchissement.

Exemple : !193.0.1.1 !C10

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

- **Afficher les numéros de semaine**

Si cette option est cochée, les numéros de semaines seront affichés à côté du calendrier

- **Encercler la date d'aujourd'hui**

Si cette option est cochée, la date courante (celle du système) apparaîtra encerclée dans le calendrier

- **Afficher la date d'aujourd'hui**

Si cette option est cochée, la date courante (celle du système) sera affichée en bas du calendrier

- **Affichage format Time/date OPTIMA PLC**

Spécifie si l'expression est une variable de type time, date, date\_and\_time ou time\_of\_day d'OPTIMA PLC directement au format Date/Heure.

- **Type Time OPTIMA PLC**

Spécifie le type date de la variable OPTIMA PLC à afficher. Les choix suivants sont possibles :  
Date, Date\_and\_time

- **Expression de neutralisation**

Permet de spécifier la variable ou l'expression de neutralisation (Celle-ci doit obligatoirement être de type booléen). Le calendrier sera neutralisé chaque fois que la variable ou l'expression sera égale à true.

*Exemple : SA10*

Il est possible de spécifier l'adresse IP distante du noyau APIGRAF PLC dans l'expression. Si c'est le cas, c'est cette adresse qui sera prise en compte pour le rafraîchissement.

*Exemple : !193.0.1.1 !SA10*

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

- **Visible si neutralisé**

Si cette option est cochée, le calendrier sera visible quand il sera neutralisé. Cependant, l'opérateur ne pourra pas agir dessus.

Si cette option n'est pas cochée, le calendrier sera invisible quand il sera neutralisé.

- **Couleur arrière plan**

Spécifie la couleur d'arrière plan du calendrier

- **Couleur Texte**

Spécifie la couleur des textes du calendrier, essentiellement les dates

- **Couleur fond titre**

Spécifie la couleur de fond du titre en haut du calendrier

- **Couleur titre**

Spécifie la couleur du titre en haut du calendrier



- **Couleur dates hors mois**

Spécifie la couleur des dates qui ne sont pas dans le mois affiché.

### 3.2.6.15.4 Fenêtre de saisie Date/Heure

[Fenêtre de saisie Date/Heure \(Détails\)](#) 



### 3.2.6.16 Objets de l'onglet ALARMES

#### 3.2.6.16.1 Historique des alarmes

Cet objet permet d'afficher l'historique des alarmes. Les alarmes sont gérées par un module OPTIM'ALARM qui peut être local (sur le même poste) ou distant.

L'objet de supervision "Historique des alarmes" se connecte (via TCP/IP) sur le module OPTIM'ALARM désigné par son adresse IP (pour le mode local : 127.0.0.1 ou LOCAL\_HOST).

Cet objet affiche alors la liste des alarmes.

Date/Heure	Type	Libellé	Priorité
 08/02/2007 13:52:19	104	Porte coupe feu principale type 1	1
 08/02/2007 13:52:24	103	GGGGGGGGG	2

Il est alors possible de faire un double-click sur les alarmes afin de les acquitter à condition d'avoir les droits nécessaires si la gestion des utilisateurs est activée. Lors du double-click, la fenêtre suivante s'ouvre :



L'utilisateur peut alors acquitter l'alarme. Si la gestion des utilisateurs est activée et qu'il ne dispose pas des droits nécessaires, le système lui demande de s'authentifier.

#### **Boîte de dialogue "Historique des alarmes" :**



Les paramètres sont :

- **Adresse IP du module OPTIM'ALARM:**

Indiquer dans ce champ l'adresse IP du poste où se trouve le module OPTIM'ALARM qui centralise les alarmes. Si le module OPTIM'ALARM se trouve sur le même poste, entrer la valeur 127.0.0.1 pour l'adresse IP (ou Local\_Host).

### 3.2.6.17 Objets de l'onglet ENREGISTREUR

#### 3.2.6.17.1 Enregistreur de données

L'enregistreur de données est utile lorsque l'utilisateur doit stocker des valeurs de données avec une fréquence d'acquisition peu élevée : ( 1 acquisition toutes les 100ms à plusieurs heures)

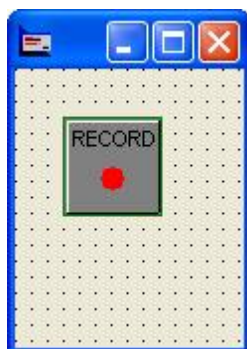
L'enregistrement et le stockage des données se fait entièrement à l'aide du superviseur qui lit simplement de manière ponctuelle les données dans l'automate.

Les données ainsi enregistrées peuvent être visualisées sous forme de courbes dans l'objet suivant " Courbe Enregistreur qui se trouve dans le même onglet d'objets.

L'enregistreur peut stocker jusqu'à 8 tableaux et donc enregistrer 8 valeurs différentes. Il enregistre également automatiquement les valeurs de temps auquel chaque enregistrement a été effectué.

Les données peuvent être enregistrés cycliquement dans un fichier de sorte que les tableaux soit sauvegardés en cas de redémarrage de la supervision.

L'objet enregistreur n'apparaît qu'en phase de conception des pages de supervision et disparaît lors de l'exécution. Il a l'aspect suivant :



**Boîte de dialogue "Enregistreur de données"**

**ENREGISTREUR**

Nom :

Nombre de points :

Enregistrement

☒ cyclique

☐ sur front montant expression

☐ sur front descendant expression

☐ sur front expression

Période :   
Valeur max : 50000

Base de temps période

☐ milliseconde

☒ seconde

☐ minute

☐ heure

Expression de déclenchement :

Num	Nom du tableau	Expression	Actif
1	TAB0		Non
2	TAB1		Non
3	TAB2		Non
4	TAB3		Non
5	TAB4		Non
6	TAB5		Non
7	TAB6		Non
8	TAB7		Non

☒ Buffer cyclique

Nombre de valeurs effacées en début de tableau quand il est plein :

☐ Enregistrement des données dans un fichier

Nom du fichier :

Temps entre deux enregistrements :  Minutes

Les paramètres sont :

- **Nom**

Indique le nom de l'enregistreur. Ce nom permettra d'identifier dans la courbe associée l'enregistreur à visualiser.

- **Nombre de points**

Nombre de points maximal de chaque tableau de données de l'enregistreur

- **Enregistrement**

Permet de sélectionner le mode d'enregistrement : Cyclique, sur front montant expression, sur front descendant expression, sur front expression. Le front indique un changement d'état de l'expression booléenne.

- **Période**

Si le mode d'enregistrement sélectionné est cyclique, entrer ici la valeur de l'intervalle entre chaque mesure. Préciser en dessous les unités de l'intervalle : milliseconde, seconde, minute, heure.

- **Expression de déclenchement**

Si le mode d'enregistrement sélectionné est : sur front montant expression, sur front descendant expression ou sur

front expression : Entrer ici l'expression qui va déclencher les enregistrements. Cette expression doit être booléenne.

- **Liste des tableaux**

Cette liste montre les 8 tableaux d'enregistrement. Seuls certains tableaux peuvent être actifs. Il est également possible de renommer les tableaux. Les noms par défaut étant : TAB0, TAB1, ..., TAB7

- **Bouton "Modifier"**

Pour paramétrer un tableau d'enregistrement (ou voie d'enregistrement), sélectionner le tableau à paramétrer dans la liste, celui-ci doit apparaître en surbrillance, puis cliquer sur le bouton "modifier". La fenêtre de paramétrage suivante s'ouvre alors :

Renseigner (facultatif) le nom du tableau

Renseigner l'expression à enregistrer ( valeur numérique )

Cocher la case "Enregistrement tableau actif" pour rendre effectif l'enregistrement dans le tableau.

- **Option "Buffer cyclique"**

Si cette option est cochée, une fois que le tableau sera rempli, un certain nombre de valeurs sera effacé en début de tableau pour permettre d'ajouter de nouvelles valeurs en fin de tableau.

- **Nombre de valeurs effacées en début de tableau lorsqu'il est plein**

Si l'option "Buffer cyclique" est cochée, ce paramètre spécifie le nombre de valeurs à effacer en début de tableau quand il est plein pour permettre d'ajouter de nouvelles valeurs en fin de tableau.

- **Enregistrement des données dans un fichier**

Si cette option est cochée, toutes les données seront enregistrées dans un fichier.

- **Nom du fichier**

Spécifier le nom du fichier qui contiendra toutes les données.

- **Temps entre deux enregistrements**

Permet de définir l'intervalle de temps entre deux enregistrement de fichiers.

### 3.2.6.17.2 Courbe Enregistreur

Cet objet permet de visualiser les données provenant de l'objet "Enregistreur de données". Cet objet se trouve sur le même onglet d'objet.

#### Boîte de dialogue "Enregistreur de données" - Onglet "Général"

**GRAPHE ENREGISTREUR**

Général Axes et Echelles Titres Légendes Fond Neutralisation Seuils Impression Zoom 3D Autres

Nom de l'enregistreur :

Tableaux de mesures sur axe Y

Nom	Unité

Ajouter Supprimer

Type d'affichage

☒ Traits  
☐ Histogramme

Type de barre

☐ Rectangle  
☐ Rectangle dégradé  
☐ Pyramide  
☐ Pyramide inversée  
☐ Flèche

Echelle en Y

☒ Axe Y1  
☐ Axe Y2

Affichage Temps

☒ Afficher temps sur l'axe des X

Format des dates/temps :

dd/mm/yy "hh:nn" ▼

Options d'affichage

Couleur de la courbe

Hint

Aide Annuler OK

Les paramètres sont :

- **Nom de l'enregistreur**

Indique le nom de l'enregistreur à visualiser.

- **Tableaux de mesure sur Axe Y**

Indique le nom de l'enregistreur à visualiser.

Cliquer sur le bouton "Ajouter" pour ajouter une nouvelle courbe de l'enregistreur à visualiser. Ce paramètre correspond aux voies de l'enregistreur : TAB0,TAB1,..TAB7 (noms par défaut de l'enregistreur).

**TABLEAU DE MESURES**

Veuillez entrer le nom du tableau

OK Annuler

Entrer le nom du tableau de l'enregistreur ( TAB0,...TAB7 ou autre si le nom par défaut a été modifié).

- **Affichage temps**

Cocher cette option pour que la courbe indique en abscisses les différent horodatages des acquisitions.

- **Format de dates/temps**

sélectionner ici un format d'affichage de l'horodatage

### Type d'affichage

Permet de sélectionner le type d'affichage pour la courbe sélectionnée dans la liste des tableaux de mesure. Il y a deux possibilités

- Traits
- Histogramme

### Type de barre

Cette option n'est accessible que si le type de graphe est un histogramme. Elle permet de sélectionner le type de barres du graphe.

### Echelle en Y

Il y a 2 échelles verticales disponibles dans le graphe, une à gauche (Y1) et une à droite (Y2). Ce champ permet de spécifier quelle échelle verticale sera utilisée pour l'affichage de la courbe.

### Couleur de la courbe

Ce bouton permet de spécifier la couleur de la courbe sélectionnée dans la liste des tableaux de mesure.

### Bouton « Hint »

Ce bouton permet de définir les paramètres de « Hint » de l'objet. Le texte « Hint » que l'on définit est le texte qui apparaît sur fond jaune lorsque l'on passe le curseur de la souris sur l'objet.

Quand on clique sur le bouton « Hint », le fenêtre suivante s'ouvre :



Cette fenêtre présente deux contrôles :

#### Option « Afficher le Hint »

Si cette option n'est pas cochée, le texte du Hint n'apparaîtra pas quand le curseur de la souris passera sur l'objet même si celui-ci est défini. Il faut donc cocher cette option si l'on veut utiliser le « Hint » de l'objet.

#### Fenêtre de saisie « Hint »

Cette fenêtre permet d'entrer le texte du Hint. Ce texte peut être un paramètre (exemple {1}).

### Boîte de dialogue graphe - Onglet "Axes et Echelles"

**Objet graphe**

Général Axes et Echelles Titre Legendes Fond Neutralisation Seuils Impression Zoom 3D Autres

**Echelles**

☒ Echelle X Automatique  
X Min 0  
X Max 10000  
Coefficient d'affichage en X 1

☒ Echelle Y1 Automatique  
Y Min 0  
Y Max 10000  
Coefficient d'affichage en Y1 1

☒ Echelle Y2 Automatique  
Y Min 0  
Y Max 10000  
Coefficient d'affichage en Y2 1

Aide Annuler OK

### Echelles

Permet de spécifier les échelles des trois axes suivants X, Y1 et Y2. L'axe Y2 peut être utilisé dans le cas où plusieurs courbes n'ont pas les mêmes unités.

Il y a deux modes d'échelles, le mode automatique et le mode manuel.

Dans le mode automatique, il faut cocher la case « Echelle X, Y1 ou Y2 Automatique » suivant l'axe concerné.

Dans ce cas, les valeurs Min et Max de l'échelle seront ajustées aux valeurs des tableaux et les champs Min et Max correspondants à l'échelle n'ont pas besoin d'être renseignés.

Si la case « Echelle automatique » n'est pas cochée. Les valeurs min et max de l'échelle seront les valeurs figurant dans les champs Min et Max de l'échelle.

### Coefficients d'affichage

Il est possible de spécifier pour chacune des trois échelles disponibles un coefficient d'affichage. Ce coefficient définit le rapport entre la valeur affichée et la valeur réelle dans le tableau.

#### Option "Echelle X Automatique"

Cocher cette option pour que l'étendue de l'échelle des X s'adapte automatiquement aux valeurs des courbes. Si cette option n'est pas cochée, il est possible de donner les valeurs XMin et XMax de l'échelle des X.

#### Coefficient d'affichage en X

Sa valeur par défaut est 1. Cette valeur est multipliée aux valeurs du tableau de mesure en X.

#### Option "Echelle Y1 Automatique"

Cocher cette option pour que l'étendue de l'échelle des Y1 (Echelle de gauche) s'adapte automatiquement aux valeurs des courbes. Si cette option n'est pas cochée, il est possible de donner les valeurs Y1Min et Y1Max de l'échelle Y1.

#### Coefficient d'affichage en Y

Sa valeur par défaut est 1. Cette valeur est multipliée aux valeurs des tableaux de mesure positionnés par rapport à l'axe Y1 (axe de gauche).

**Option "Echelle Y2 Automatique"**

Cocher cette option pour que l'étendue de l'échelle des Y2 ( Echelle de droite ) s'adapte automatiquement aux valeurs des courbes. Si cette option n'est pas cochée, il est possible de donner les valeurs Y2Min et Y2Max de l'échelle Y2.

**Coefficient d'affichage en Y**

Sa valeur par défaut est 1. Cette valeur est multipliée aux valeurs des tableaux de mesure positionnés par rapport à l'axe Y2 (axe de droite).

**Boîte de dialogue graphe - Onglet "Titre"**

The image shows a software dialog box titled "Objet graphe" with a blue border and a standard Windows-style title bar. It features a series of tabs at the top: "Général", "Axes et Echelles", "Titre" (which is currently selected and highlighted in yellow), "Legendes", "Fond", "Neutralisation", "Seuils", "Impression", "Zoom", "3D", and "Autres". The main area of the dialog is light beige. On the left side, there are five text input fields, each preceded by a label: "Titre du graphe", "Expression à ajouter au titre", "Titre Axe X", "Titre Axe Y1", and "Titre Axe Y2". All these fields are currently empty. At the bottom right of the dialog, there are three buttons: "Aide", "Annuler", and "OK".

**Titres**

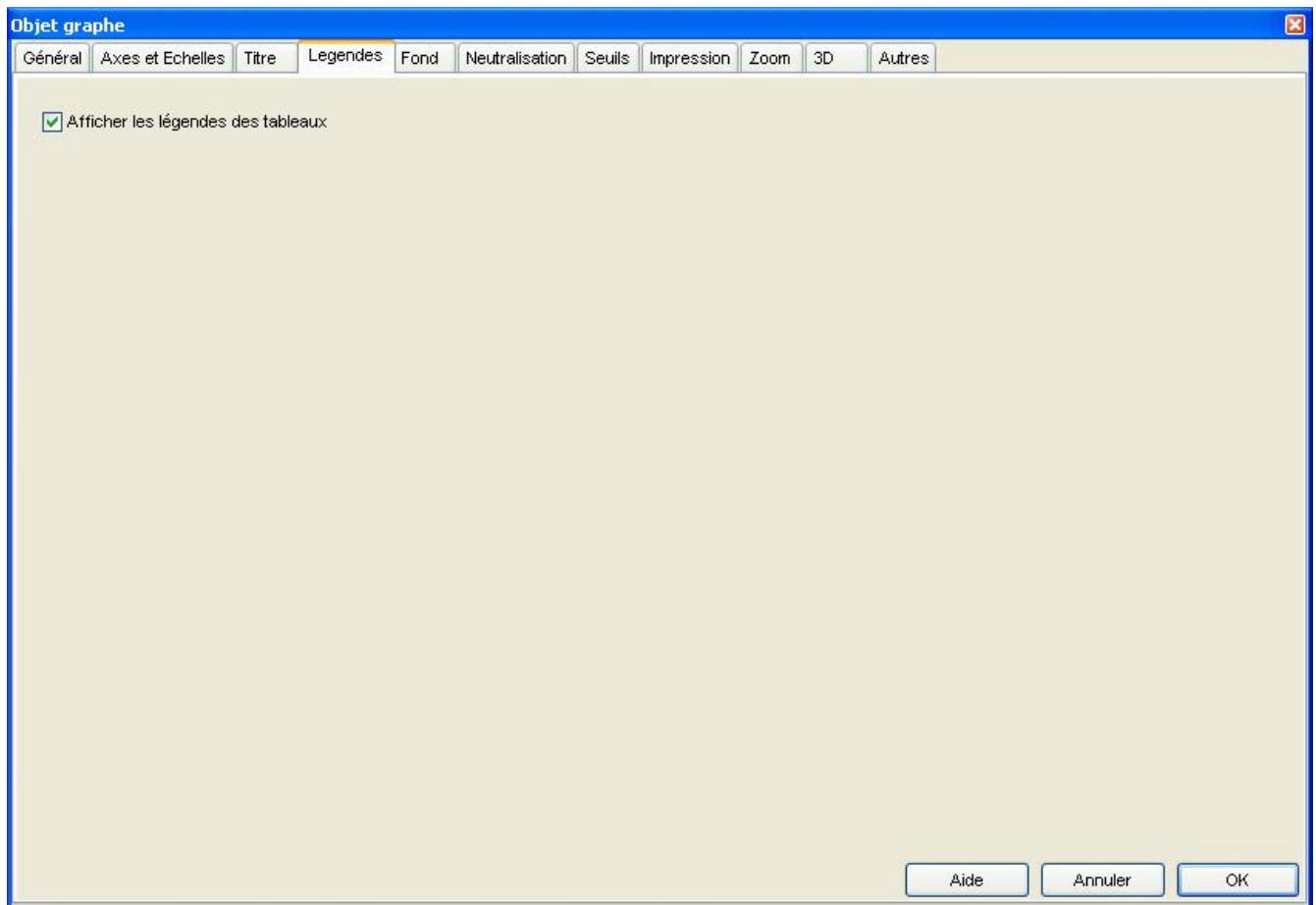
Un titre général peut être donné à la courbe ainsi que des légendes pour chacun des trois axes disponibles.

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

On peut également spécifier une expression à ajouter au titre. Par exemple un numéro de série contenu dans un registre : R100.

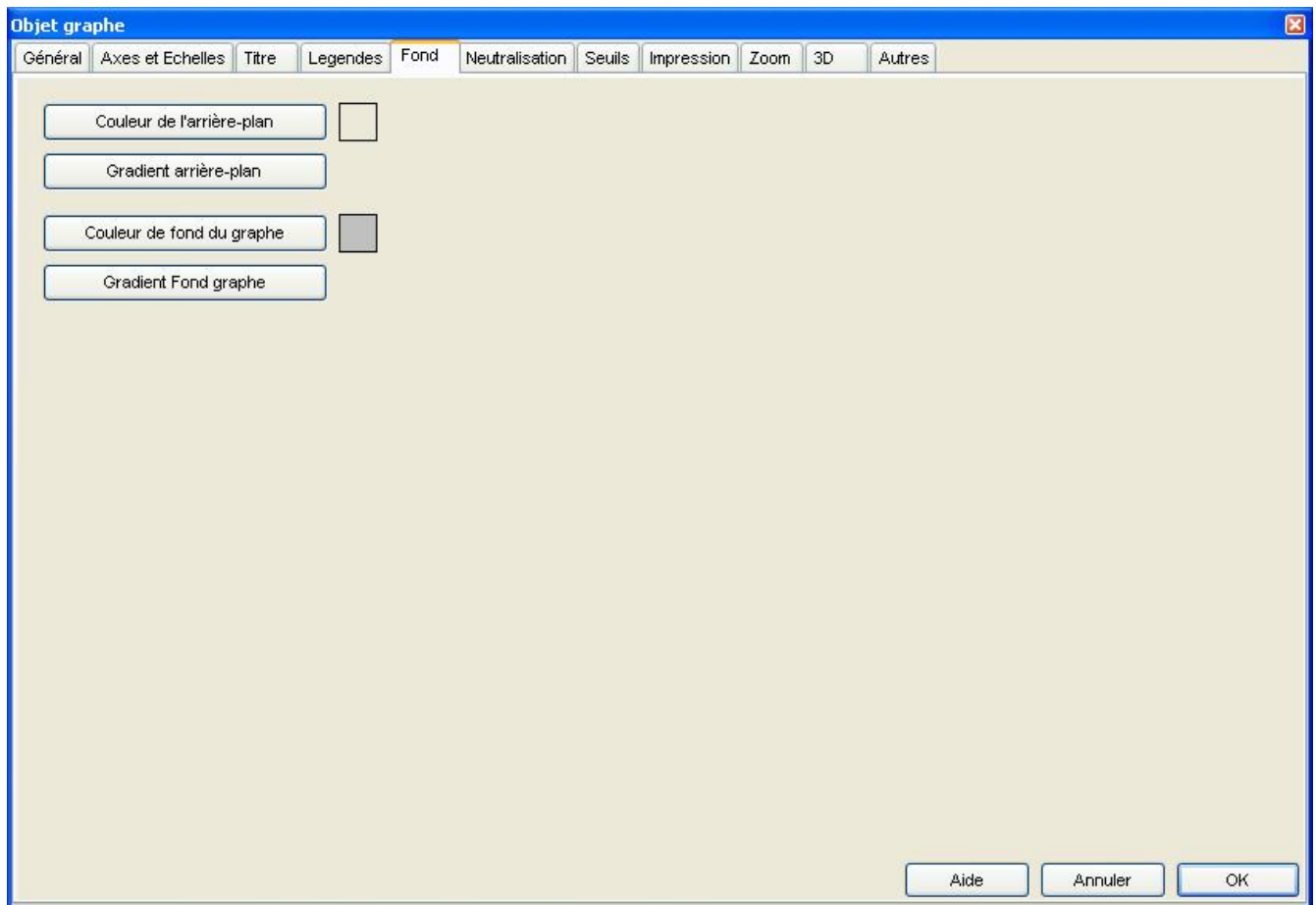
**Boîte de dialogue graphe - Onglet "Titre"**





Cocher l'option "Afficher les légendes" pour afficher les noms des tableaux de mesures associés aux couleurs des courbes correspondantes.

#### **Boîte de dialogue graphe - Onglet "Fond"**

**Couleur du rectangle du graphe**

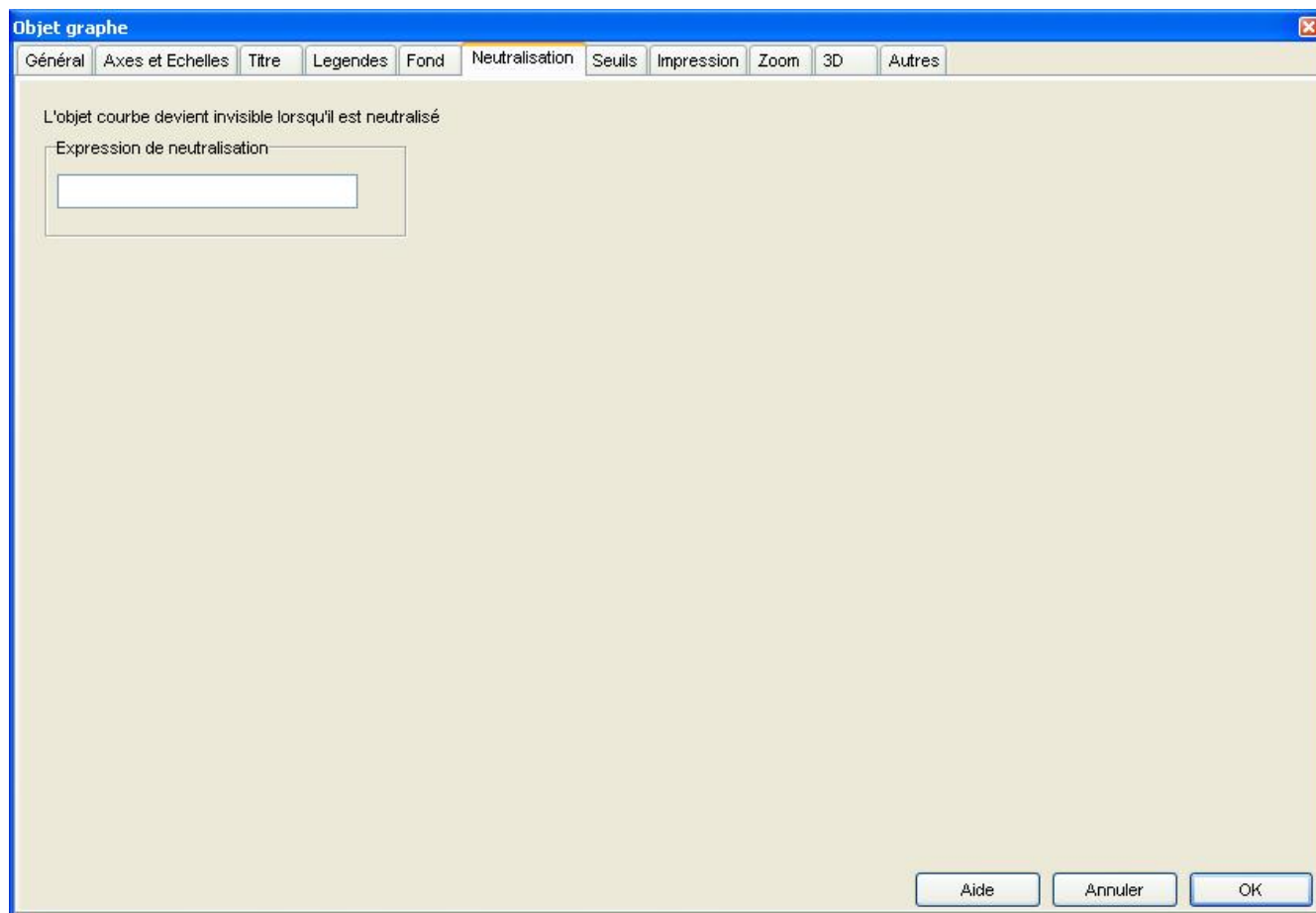
Ce bouton permet de choisir la couleur d'arrière plan du graphe.

**Couleur de l'arrière plan**

Ce bouton permet de choisir la couleur du contour de l'objet graphe.

Des boutons gradients permettent également de définir des dégradés de couleurs en fond de l'objet et en arrière plan de la courbe.

**Boîte de dialogue graphe - Onglet "Neutralisation"**



### Expression de neutralisation

Permet de spécifier la variable ou l'expression de neutralisation (Celle-ci doit obligatoirement être de type booléen). L'objet courbe sera invisible chaque fois que la variable ou l'expression sera égale à true.

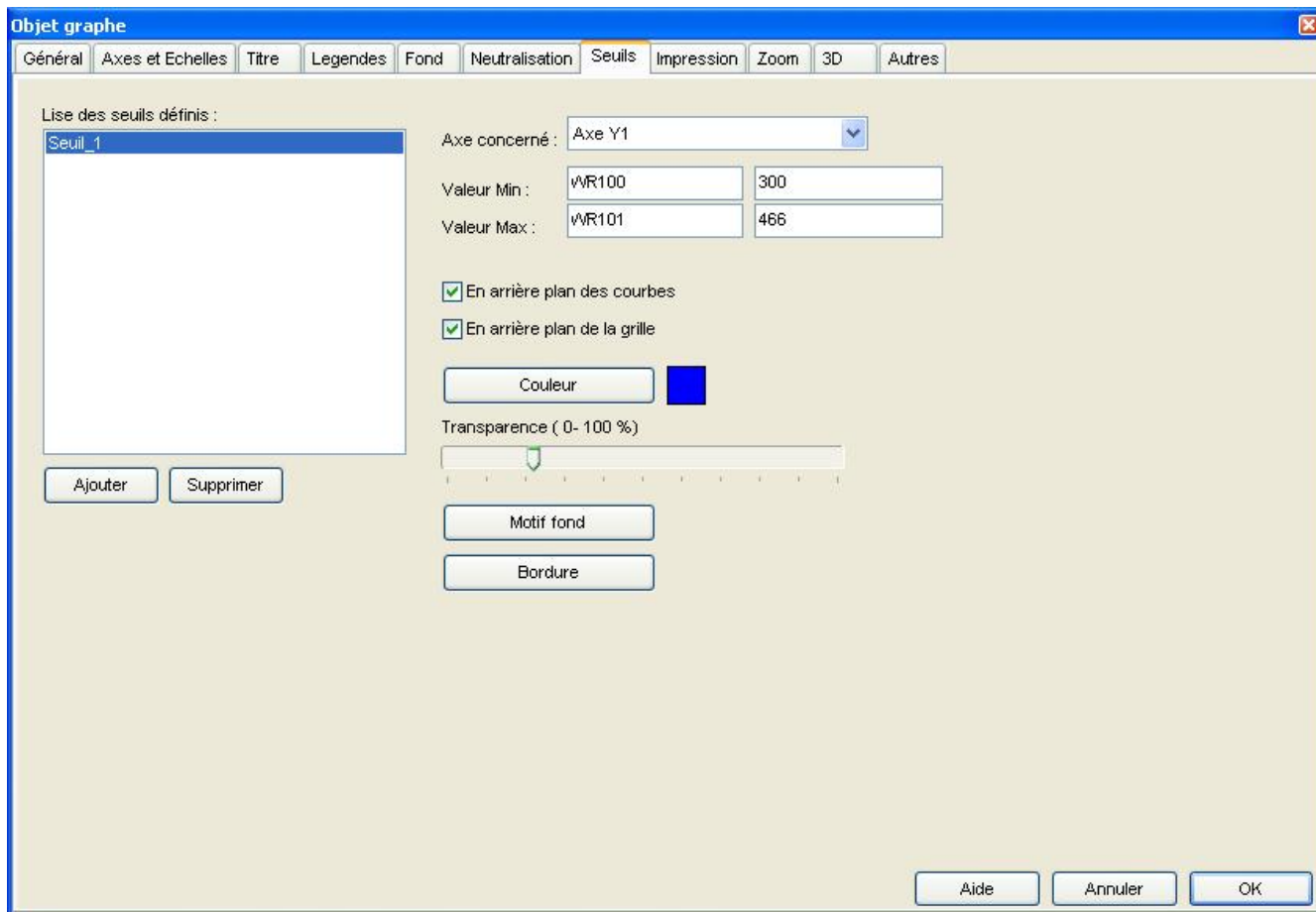
*Exemple : SA10*

Il est possible de spécifier l'adresse IP distante du noyau APIGRAF PLC dans l'expression. Si c'est le cas, c'est cette adresse qui sera prise en compte pour le rafraîchissement.

*Exemple : !193.0.1.1 !SA10*

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

### Boîte de dialogue graphe - Onglet "Seuils"



Les seuils sont des zones colorées qui s'affichent sur la courbes et qui permettent en général de visualiser des zones limites. Des seuils peuvent être ajoutés ou supprimés dans la liste déroulante à l'aide des boutons "Ajouter" et "Supprimer". Une fois un seuil ajouté, il est possible de paramétrer son aspect :

#### **Axe concerné :**

Définit l'axe auquel se rapporte le seuil, un seuil peut donc être horizontal si il se réfère à Y1 ou Y2 ou vertical si il se réfère à l'axe X.

#### **Valeur Min et Valeur Max :**

Ces valeurs définissent l'étendue de la zone seuil, on peut entrer des valeurs numériques constantes ou bien des expressions numériques.

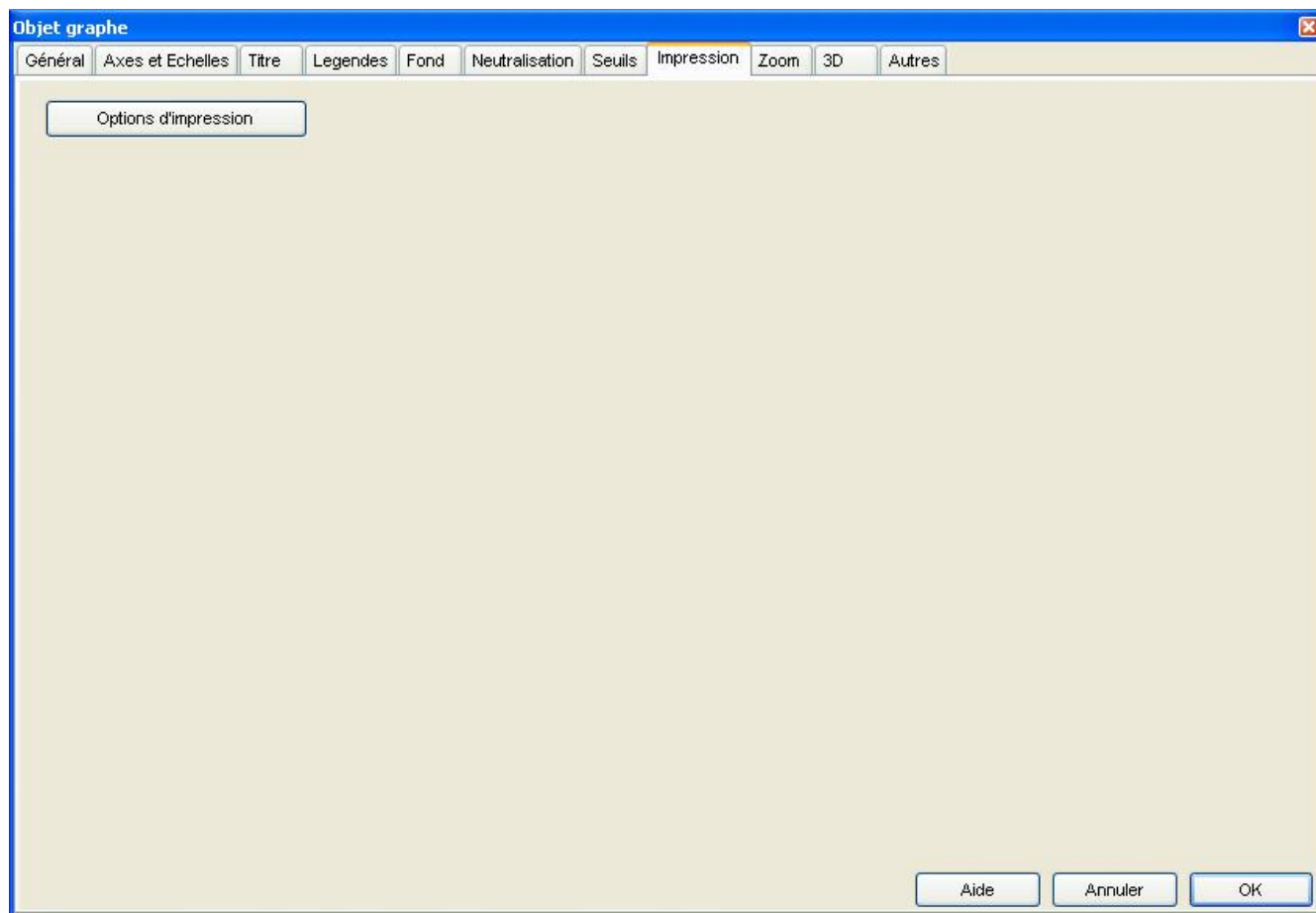
*Exemple : R10*

Il est possible de spécifier l'adresse IP distante du noyau OPTIMA PLC ou APIGRAF PLC dans l'expression. Si c'est le cas, c'est cette adresse qui sera prise en compte pour le rafraîchissement.

*Exemple : !193.0.1.1 !R10*

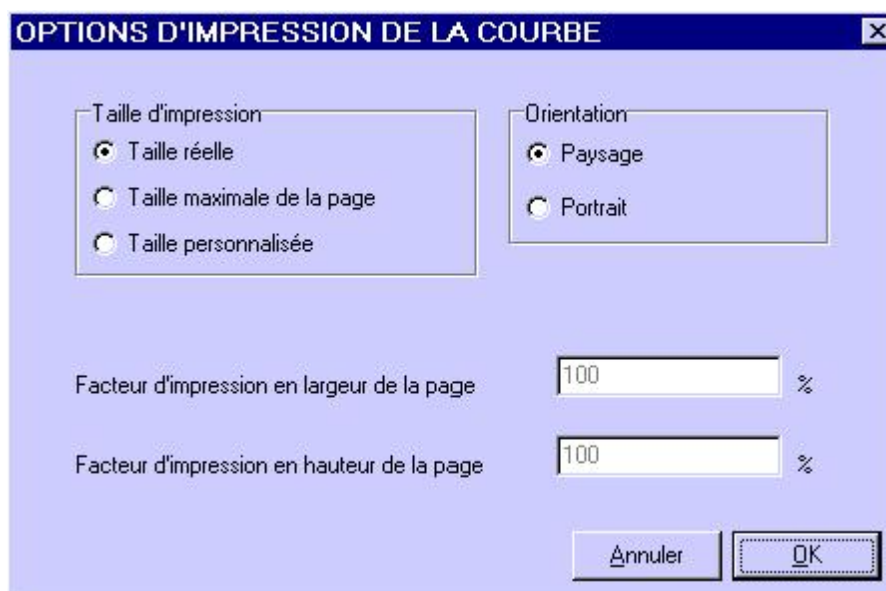
L'apparence de la zone seuil peut être également redéfinie: la transparence, le motif du fond, la couleur, ...

#### **Boîte de dialogue graphe - Onglet "Impression"**



### Options d'impression

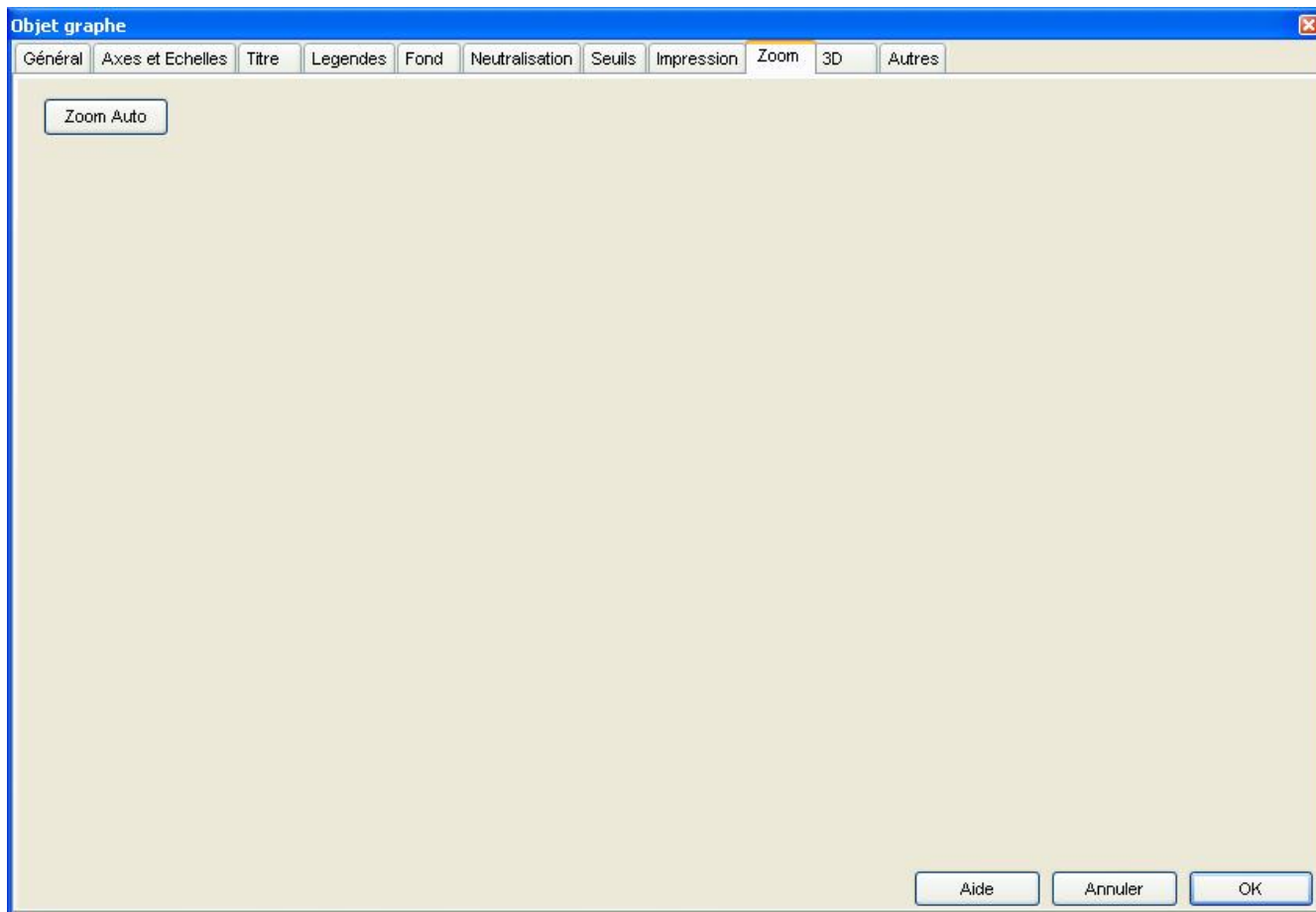
Les courbes peuvent être imprimées en mode exécution. Il faut pour cela double-cliquer sur la courbe à imprimer en mode exécution. Les options d'impression permettent de paramétrer l'impression de la courbe :



Si l'option « Taille personnalisée » est choisie, les facteurs d'impression en largeur et en hauteur de la page seront pris en compte pour l'impression. Ces facteurs seront appliqués à la taille d'affichage à l'écran de la courbe.

L'option « Taille maximale de la page » va étendre la courbe sur toute la taille du papier de sortie. Dans ce cas la définition de l'impression est moins bonne.

### Boîte de dialogue graphe - Onglet "Zoom"



Le bouton "Zoom" ouvre la fenêtre suivante :

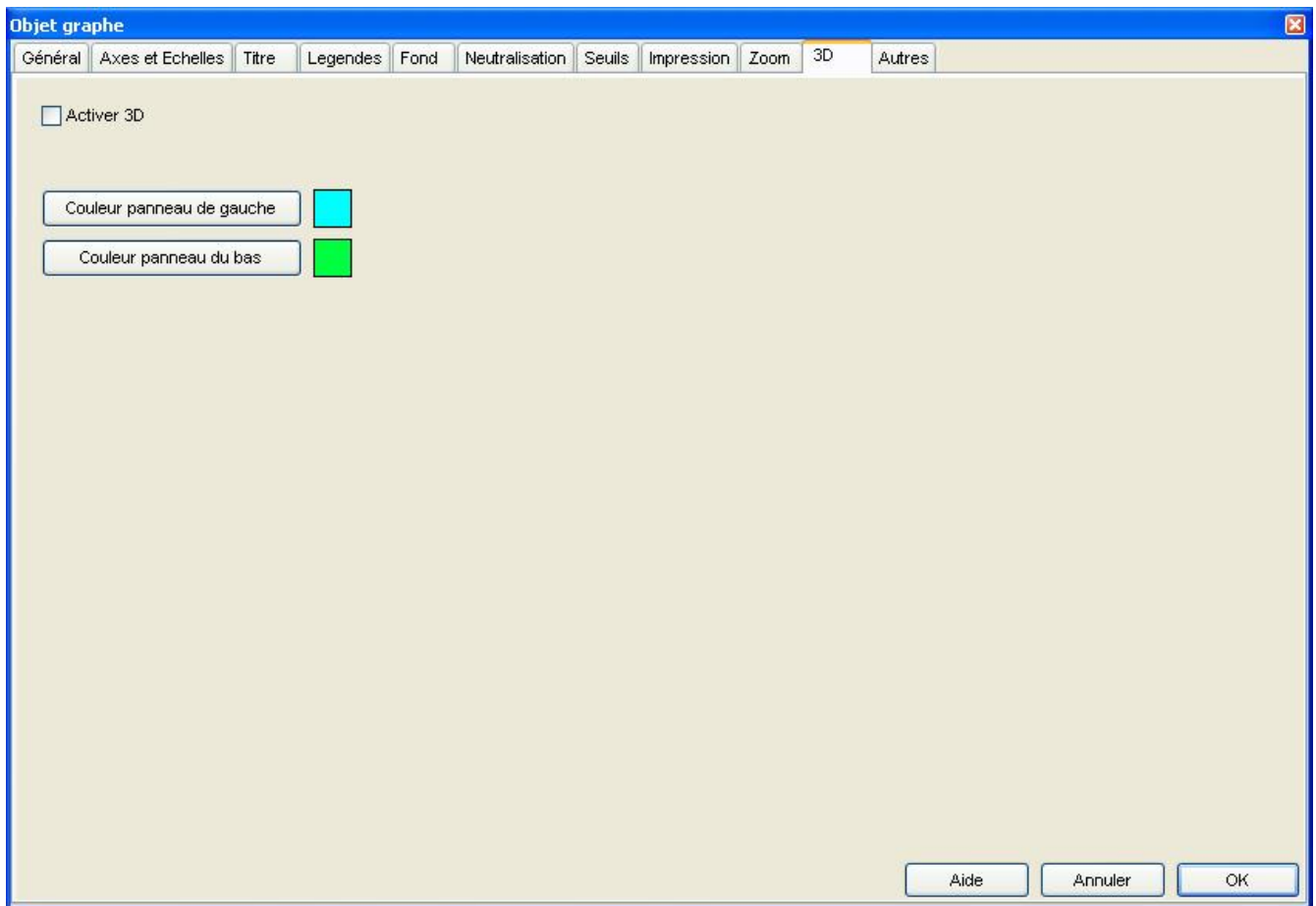
Cette fenêtre permet d'associer des variables de contrôle des zooms de la courbe.

**Exemple** : Entrer SA10 dans la boîte de saisie "Expression zoom plus abscisses". Définir ensuite un bouton API monostable avec la valeur SA10. A chaque pression du bouton, la courbe va être zoomée en X avec le pas spécifié (2 par défaut).

On peut faire la même chose pour contrôler le déplacement ("Shift") de la courbe.

Une expression reset zoom permet d'annuler tous les zooms effectués sur la courbe.

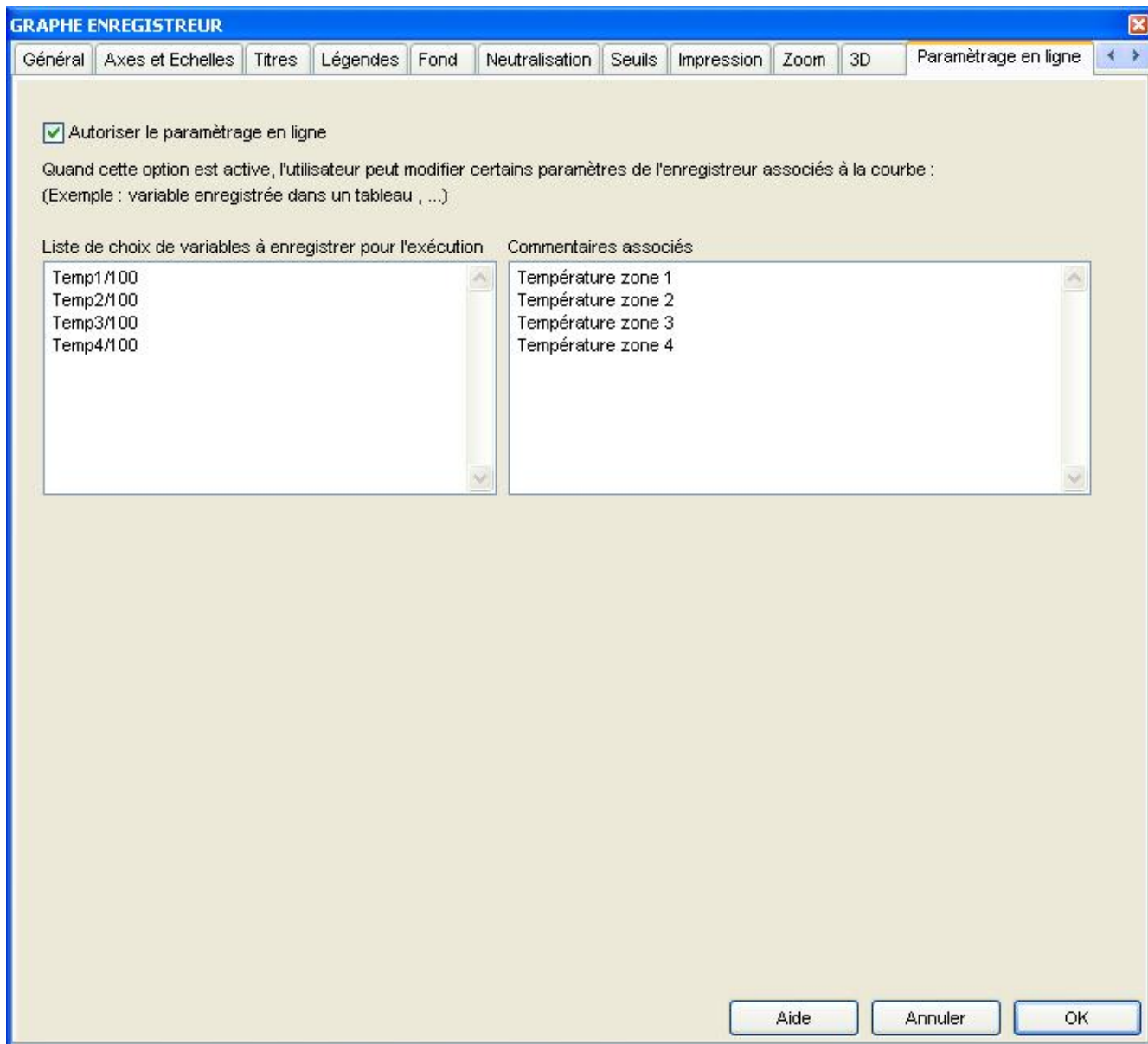
### **Boîte de dialogue graphe - Onglet "3D"**



#### **Option "Activer 3D"**

Permet d'afficher la fenêtre du graphe en 3 dimensions. On peut alors définir les couleurs des panneaux de gauche et du bas créés par la visualisation 3D.

### **Boîte de dialogue "Enregistreur de données" - Onglet "Paramétrage en ligne"**



### Option "Autoriser le paramétrage en ligne"

Cette option quand elle est activée permet à l'utilisateur de modifier les paramètres des données de l'enregistreur associé à la courbe en phase d'exécution de la supervision.

### Liste de choix de variables à enregistrer pour l'exécution

Cette liste va définir toutes les variables qui pourront être choisies dans une liste en phase d'exécution. Entrer dans le champ une variable par ligne

### Commentaires associés (facultatif)

Cette liste permet d'associer des commentaires pour chaque variable de la liste. Ces commentaires apparaîtront lorsque l'utilisateur choisira une nouvelle variable.

Entrer dans le champ un commentaire par ligne.

Le commentaire de la première ligne correspondra à la variable de la ligne 1 etc...

Si l'option "**Autoriser le paramétrage en ligne**" est active,

dans ce cas un double clic sur la cadre de la courbe pendant l'exécution de la supervision aura pour effet d'ouvrir la fenêtre de menu suivante :

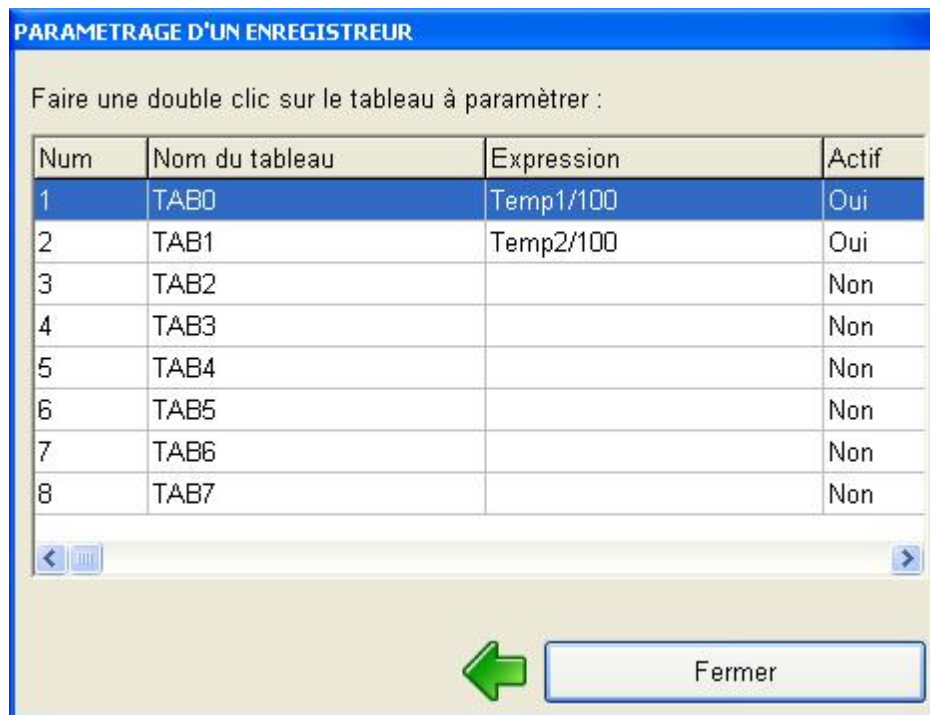




Le bouton "imprimer" permet d'imprimer la courbe sur l'imprimante définie par défaut

Le bouton "Paramètrer" ouvre la fenêtre de paramétrage en ligne :

**Exemple :**



L'utilisateur peut modifier un des tableaux de la liste en faisant un double clic sur celui-ci. La fenêtre suivante s'ouvre :



ENREGISTREUR - PARAMETRAGE D'UN TABLEAU DE MESURE

Nom du Tableau : TAB1

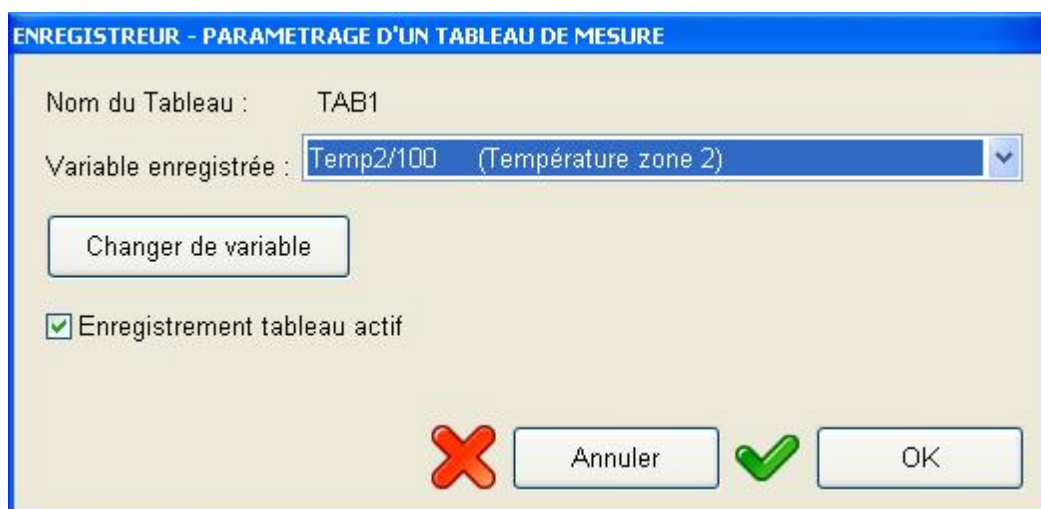
Variable enregistrée : Temp2/100

Changer de variable

☒ Enregistrement tableau actif

Annuler OK

Pour choisir une nouvelle variable dans la liste des variables possibles, appuyer sur le bouton "Variable". Un Selecteur à liste déroulante de variables apparaît ensuite sur la fenêtre :



ENREGISTREUR - PARAMETRAGE D'UN TABLEAU DE MESURE

Nom du Tableau : TAB1

Variable enregistrée : Temp2/100 (Température zone 2)

Changer de variable

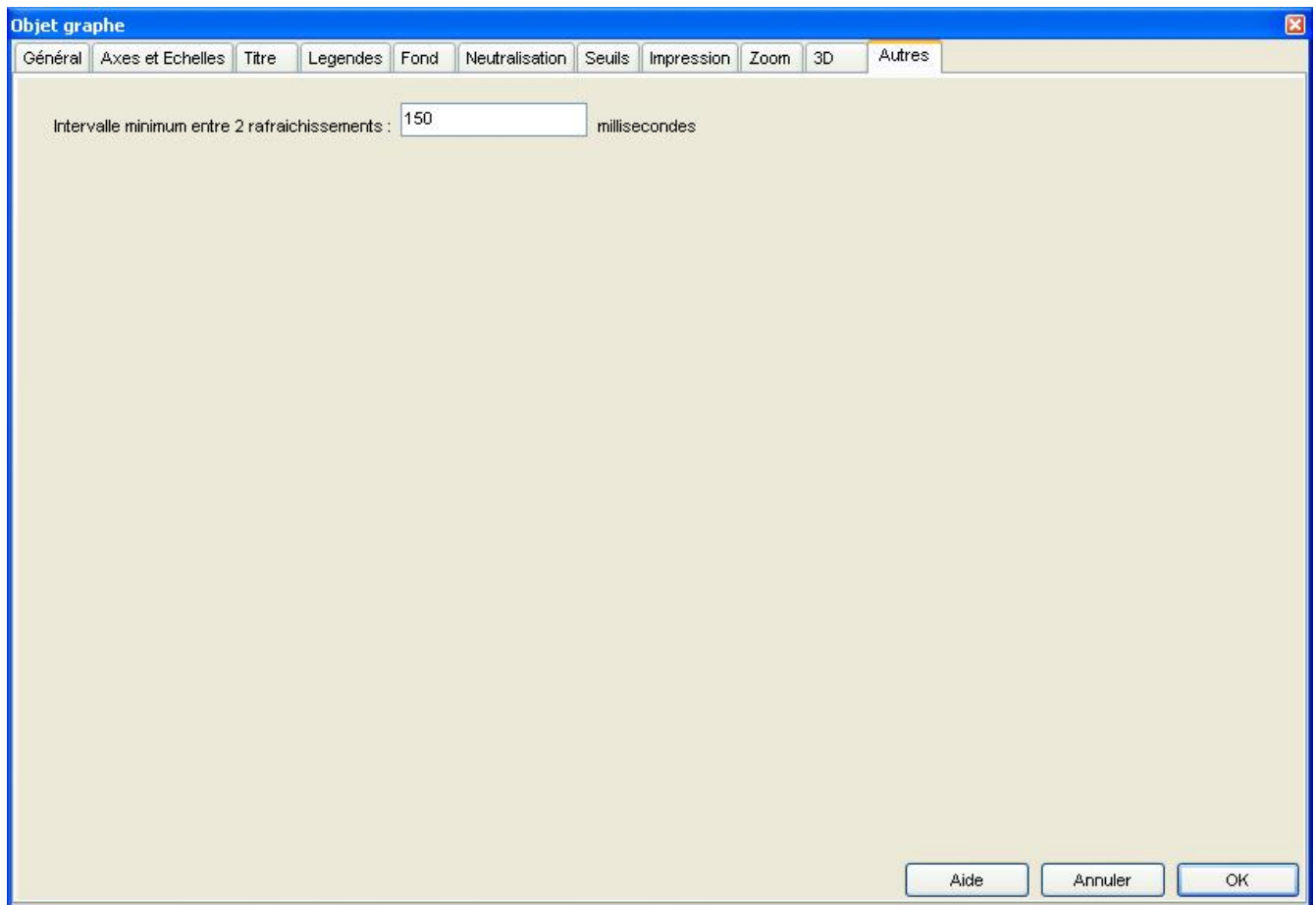
☒ Enregistrement tableau actif

Annuler OK

Il est également possible dans cette fenêtre d'inhiber ou d'activer l'enregistrement d'un tableau.

**Remarque :** si une modification a été effectuée, tous les tableaux de mesure de l'enregistreur seront remis à 0.

#### **Boîte de dialogue graphe - Onglet "Autres"**



Les affichages complexes de courbes volumineuses demandent beaucoup de ressources. On peut donc limiter la fréquence de rafraichissement des affichages.

### 3.2.6.18 Objets de l'onglet MULTIMEDIA

#### 3.2.6.18.1 Média player

L'objet Média player permet de lire des vidéos sur une page de supervision. Cet objet utilise le moteur de WINDOWS MEDIA PLAYER pour effectuer la lecture. Pour lire certains types de fichiers, il est donc nécessaire de mettre à jour Windows média player ou les codecs windows média player.



L'objet se présente sous forme d'un écran noir dans lequel seront jouées les vidéos.

**Boîte de dialogue "Média player" - Onglet Général :**

**MEDIA Player**

Général Neutralisation

Expression du fichier  OPC

Si le fichier est constant, entrer le nom du fichier entre côtes , Exemple : 'C:\FILE1.WMV'

☒ Démarrage automatique de la lecture

☐ Rebouclage automatique ( actif si les commandes ne sont pas utilisées )

Expression de commande  OPC

Expression index début lecture (facultatif)  OPC secondes

Expression index fin lecture (facultatif)  OPC secondes

Expression index courant lecture (facultatif)  OPC secondes

Expression du statut du player (facultatif)  OPC

Barre de contrôles  ▼

Expression durée totale du fichier (facultatif)  secondes (type réel)

Aide Annuler OK

**Expression du fichier**

Il est possible de spécifier un fichier vidéo ou une variable de type chaîne dont la valeur contiendra le nom du fichier vidéo :

Pour un nom de fichier vidéo, mettre des cotes simples :

Exemple : 'C:\TEST.AVI'

Pour une variable, entrer le nom de la variable :

Exemple : C13

Il est possible de spécifier l'adresse IP distante du noyau OPTIMA PLC ou APIGRAF dans l'expression. Si c'est le cas, c'est cette adresse qui sera prise en compte pour le rafraîchissement.

Exemple : !193.0.1.1 !C13

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

Dans le cas d'une variable, le nom du fichier vidéo à lire est dynamique et correspond à la valeur de la variable.

### **Option "Démarrage automatique de la lecture"**

Si cette option est cochée, la lecture du fichier vidéo débute automatiquement. Dans le cas d'un fichier vidéo dynamique, la lecture du fichier vidéo recommence automatiquement si le nom de la vidéo à lire change.

### **Option "Rebouclage automatique ( actif si les commandes ne sont pas utilisées )"**

Si les commandes (voir plus bas) ne sont pas utilisées et si cette option est cochée, le fichier vidéo recommencera au début de manière automatique dès que la lecture de celui-ci sera achevée.

### **Expression de commande (Optionnel) :**

Permet de spécifier une variable automate ( de type numérique entière, exemple R10,...) qui commandera le player. Suivant la valeur associée à la variable automate, le player effectuera les actions suivantes :

0	Aucune action
1	Lecture
2	Stop
3	Mettre en pause
4	Lecture continue

### **Expression index début lecture (Optionnel) :**

Permet de spécifier une valeur ou un variable automate contenant le valeur de d'index de début de la lecture. Cette valeur est de type numérique entière et est exprimée en secondes :

Exemples : 1 , R10,...

Si la valeur vaut 1, la lecture débutera le film à la position 1 seconde.

### **Expression index fin lecture (Optionnel) :**

Permet de spécifier une valeur ou un variable automate contenant le valeur de d'index de fin de la lecture. Cette valeur est de type numérique entière et est exprimée en secondes :

Exemples : 1 , R10,...

Si la valeur vaut 10, la lecture sera stoppé automatiquement à la position 10 secondes du film.

### **Expression index courant lecture (Optionnel) :**

Spécifier une variable automate de type numérique. La valeur de la variable automate sera automatiquement affectée par la position en cours de la lecture exprimée en secondes.

**Expression du statut du player (Optionnel) :**

Entrer dans ce champ une variable automate de type numérique entier. La valeur de la variable sera affectée par le statut du player.

Si la valeur du statut est 3, le fichier est en cours de lecture.

Si la valeur du statut est  $\leq 1$ , 9 ou 10, le fichier est stoppé ou fini.

**Barre de contrôle :**

Permet de sélectionner le mode d'affichage de la barre de contrôle du player : Aucune ( Pas de barre de contrôle visible) ou complète.

**Expression durée totale du fichier (Optionnel) :**

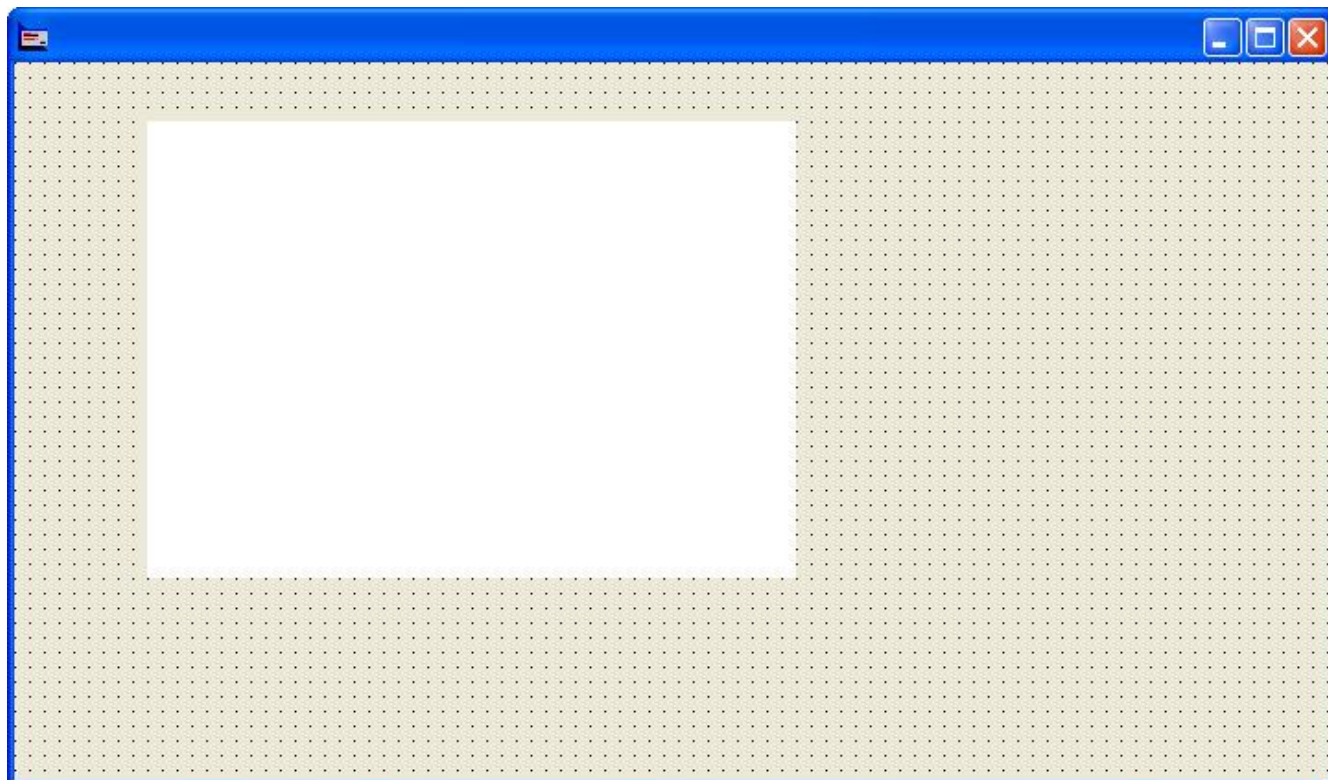
Entrer dans ce champ une variable de type numérique réel ( exemple : RR10). La valeur de la variable sera affectée par la durée totale de la vidéo en cours exprimée en secondes.

**Boîte de dialogue "Média player" - Onglet Neutralisation :**

L'onglet neutralisation permet de spécifier une variable booléenne de neutralisation qui rendra invisible le player quand la valeur de cette variable sera égale à 1.

### 3.2.6.18.2 Flash ShockWave player

L'objet Flash player permet de lire des fichiers flash sur une page de supervision. Cet objet utilise l'ActiveX ShockWaveFlash d'Adobe. Pour que cet objet fonctionne correctement, il peut être nécessaire de télécharger et d'installer ShockWave Flash Player.



L'objet se présente sous forme d'une fenêtre dans laquelle seront jouées les fichiers flash.

**Boîte de dialogue "Flash player" - Onglet Général :**

**Objet FLASH**

Général Neutralisation

Expression du fichier  OPC

Si le fichier est constant, entrer le nom du fichier entre côtes , Exemple : 'C:\FILE1.WMV'

☒ Démarrage automatique de la lecture

☒ Rebouclage automatique ( actif si les commandes ne sont pas utilisées )

Expression de commande player (facultatif)  OPC

Expression du statut du player (facultatif)  OPC

Aide Annuler OK

**Expression du fichier**

Il est possible de spécifier un fichier flash ou une variable de type chaîne dont la valeur contiendra le nom du fichier flash :

Pour un nom de fichier flash, mettre des cotes simples :

Exemple : 'C:\TEST.SWF'

Pour une variable, entrer le nom de la variable :

Exemple : C13

Il est possible de spécifier l'adresse IP distante du noyau OPTIMA PLC ou APIGRAF dans l'expression. Si c'est le cas, c'est cette adresse qui sera prise en compte pour le rafraîchissement.

Exemple : !193.0.1.1 !C13

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

Dans le cas d'une variable, le nom du fichier vidéo à lire est dynamique et correspond à la valeur de la variable.

**Option "Démarrage automatique de la lecture"**

Si cette option est cochée, la lecture du fichier flash débute automatiquement. Dans le cas d'un fichier vidéo dynamique, la lecture du fichier flash recommence automatiquement si le nom du fichier à lire change.

**Option "Rebouclage automatique ( actif si les commandes ne sont pas utilisées )"**

Si les commandes (voir plus bas) ne sont pas utilisées et si cette option est cochée, le fichier flash recommencera au début de manière automatique dès que la lecture de celui-ci sera achevée.

#### **Expression de commande (Optionnel) :**

Permet de spécifier une variable automate ( de type numérique entière, exemple R10,...) qui commandera le player. Suivant la valeur associée à la variable automate, le player effectuera les actions suivantes :

0	Aucune action
1	Lecture
2	Stop
3	Mettre en pause
4	Lecture continue

#### **Expression du statut du player (Optionnel) :**

Entrer dans ce champ une variable automate de type numérique entier. La valeur de la variable sera affectée par le statut du player.

Si la valeur du statut est 3, le fichier est en cours de lecture.

Si la valeur du statut est 0, le fichier est stoppé ou fini.

#### **Boîte de dialogue "Flash player" - Onglet Neutralisation :**

L'onglet neutralisation permet de spécifier une variable booléenne de neutralisation qui rendra invisible le player quand la valeur de cette variable sera égale à 1.

### **3.2.6.19 Objets de l'onglet AUTRES**

#### **3.2.6.19.1 Recette**

Ce type d'objet permet la création de recettes dans l'écran de supervision.

Une recette est un ensemble de valeurs données à un ensemble de variables PLC.

Les recettes sont sauvegardées dans des fichiers RECETTE qui ont l'extension :  
\*.RECW

Un objet « RECETTE » se traduit à l'exécution par un bouton.

Lors de l'exécution de la page, un clic de la souris sur le bouton associé à la recette ouvre une boîte de dialogue dans laquelle les valeurs de la recette sont affichées.

Vous pouvez alors y saisir de nouvelles valeurs, rappeler des sauvegardes de recettes, sauvegarder des recettes, transférer les données de la recette vers les données PLC, ..

Ceci dépend des possibilités qui sont paramétrées dans cet objet.

#### **Boîte de dialogue « recette » :**



Objet Recette

Liste des variables :

Raccourci clavier :

Possibilités offertes :

Type de bouton :

Répertoire des fichiers :

Libellé :

Epaisseur cadre :

Texte du bouton:

☐ Choix du repertoire autorisé

Couleur fond  Caractères

Image de fond

Supprimer l'image

☐ Ajuster la taille de l'image à celle de l'objet

☐ Transparence automatique de l'image

Neutralisation

Expression de neutralisation :

Couleur de fond  Caractères

Texte du bouton

☒ Visible si neutralisé

Aide Annuler OK

### Champ « Liste des variables » :

Toutes les variables que vous souhaitez voir dans la recette doivent être énumérées par leur nom et séparées par une virgule.

Des suites de variables peuvent être définies en indiquant le type de variable et l'indice du premier élément, suivi de deux points (..) et de l'indice du dernier élément (Xi..j). La visualisation d'un complément logique d'une variable booléenne est obtenue en la faisant précéder du caractère \ (Xi).

#### Exemple :

La déclaration suivante :

R0,R3..5,G3,L40..45,SA2,\SA3..5,A5,DT10,C15

Permettra de visualiser les variables suivantes dans la recette :

R0 R3 R4 R5 G3 L40 L41 L42 L43 L44 L45 SA2 \SA3 \SA4 \SA5 A5 DT10 C15

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

### Raccourci clavier

Cette option permet de spécifier une touche dont l'utilisation permet d'activer directement le bouton RECETTE. Les valeurs acceptées dans ce champ sont :

Lettres de A à Z, les majuscules et les minuscules n'étant pas différenciées.

Les chiffres de 0 à 9

*Attention à l'utilisation conjointe de fenêtres de saisie et de touche de raccourci de type alphanumérique sur des boutons ! ! En effet si lors d'une saisie, on tape le chiffre 9 et que dans la même page se trouve un bouton avec « 9 » pour raccourci, le bouton sera enclenché.*

Les touches de fonctions de F2 à F12 (La touche F1 étant réservée pour l'aide Windows®)

INSERT, SUPPR, ESCAPE, BACK

Les combinaisons de touches avec ALT : ALT A, ALT B, ALT C, ..... , ALT Z, ALT 0,... ALT 9

### **Possibilités**

Les choix possibles sont :

#### -Toutes les possibilités :

Toutes les fonctions possibles de la recette sont accessibles (consultations, modification, sauvegarde, chargement, création )

#### -Ouvrir seulement

Lorsque l'objet recette est activé ( c'est à dire lorsque l'on clique sur le bouton associé à la recette ), une boîte contenant les noms des fichiers existants apparaît. L'opérateur ne peut que choisir l'un d'eux et le valider. La recette est alors immédiatement transférée dans les variables PLC.

#### -Sauver seulement

Lors de l'activation de l'objet recette, l'opérateur peut entrer le nom d'un fichier, dans lequel les variables PLC de la recette sont sauvegardées. Si le fichier existe déjà, celui-ci sera écrasé.

#### -Créer seulement

Lors de l'activation de l'objet recette, l'opérateur peut entrer le nom d'un fichier, dans lequel les variables PLC de la recette sont sauvegardées. Le fichier ne doit pas exister sinon l'opération ne sera pas possible.

#### -Mettre à jour seulement

Lorsque l'objet recette est activé, les variables PLC de la recette sont sauvegardées dans le fichier spécifié dans le champ « Nom du fichier par défaut » (Ce champ n'apparaît que si l'option « mettre à jour seulement est sélectionnée »). Il n'y a donc pas de choix de fichier pour la sauvegarde.

### **Type de bouton**

Ce champ permet de choisir entre un bouton rectangulaire ou un bouton rond (ou elliptique).

### **Répertoire fichiers**

Permet de spécifier le sous répertoire dans lequel se trouvent ou se trouveront les fichiers recettes.

Si le champ est vide, les fichiers recettes se trouveront dans le répertoire du projet.

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

### **Libellé**

Le libellé est le nom que vous donnez à votre recette.

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

### **Epaisseur du cadre**

Spécifie l'épaisseur du cadre du bouton RECETTE.

### **Texte du bouton**

Spécifie le texte qui apparaît sur le bouton RECETTE.

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

### **Couleur du Fond**

Permet de modifier la couleur de fond du bouton RECETTE.

### **Bouton « Caractères »**

Permet de spécifier la police de caractères utilisée par le texte du bouton RECETTE.

### **Bouton « Image de fond »**

Permet de spécifier une Image Bitmap qui apparaîtra en fond sur le bouton de la RECETTE.

### **Bouton « Supprimer Image »**

Permet de supprimer l'image de fond associée à l'objet recette.

**Option « Ajuster la taille de l'image à celle de l'objet »**

Si cette option est cochée, l'image apparaîtra étirée sur toute la surface du bouton de la RECETTE.

**Option « Transparence Automatique »**

Lorsque cette option est activée, la couleur de fond de l'image déterminée automatiquement sera transparente.

**Groupe de champs « Neutralisation »**

Ces champs concernent la neutralisation du bouton associé à la recette. Le bouton, s'il est neutralisé, ne sera plus actionnable. Toutes les actions sur la recette ne seront donc pas réalisables.

**Expression de neutralisation**

Permet de spécifier une variable PLC (de type booléen) qui lorsque celle-ci sera à l'état « VRAI » neutralisera le bouton associé à la RECETTE.

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

**Option « visible si neutralisé »**

Si cette option n'est pas cochée, le bouton deviendra invisible lorsqu'il sera neutralisé.

**Bouton « Couleur de Fond » du groupe de champs « neutralisation »**

Ce bouton permet de spécifier la couleur de fond du bouton de la RECETTE lorsque celui-ci sera neutralisé.

**Bouton « Caractères » du groupe de champs « neutralisation »**

Ce bouton permet de choisir la police de caractère du texte du bouton associé à la RECETTE lorsque celui-ci sera neutralisé.

**Champ « Texte du bouton » du groupe de champs « neutralisation »**

Entrer dans ce champ le texte à afficher sur le bouton de la recette lorsque celui-ci sera neutralisé.

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

**Bouton « Hint »**

Ce bouton permet de définir les paramètres de « Hint » de l'objet. Le texte « Hint » que l'on définit est le texte qui apparaît sur fond jaune lorsque l'on passe le curseur de la souris sur l'objet.

Quand on clique sur le bouton « Hint », le fenêtre suivante s'ouvre :



Cette fenêtre présente deux contrôles :

**Option « Afficher le Hint »**

Si cette option n'est pas cochée, le texte du Hint n'apparaîtra pas quand le curseur de la souris passera sur l'objet même si celui-ci est défini. Il faut donc cocher cette option si l'on veut utiliser le « Hint » de l'objet.

**Fenêtre de saisie « Hint »**

Cette fenêtre permet d'entrer le texte du Hint. Ce texte peut être un paramètre (exemple {1}).

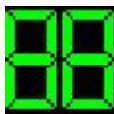
### 3.2.6.19.2 Afficheurs 7 segments

Cet objet indique la valeur d'une variable sous la forme d'un afficheur 7 segments.

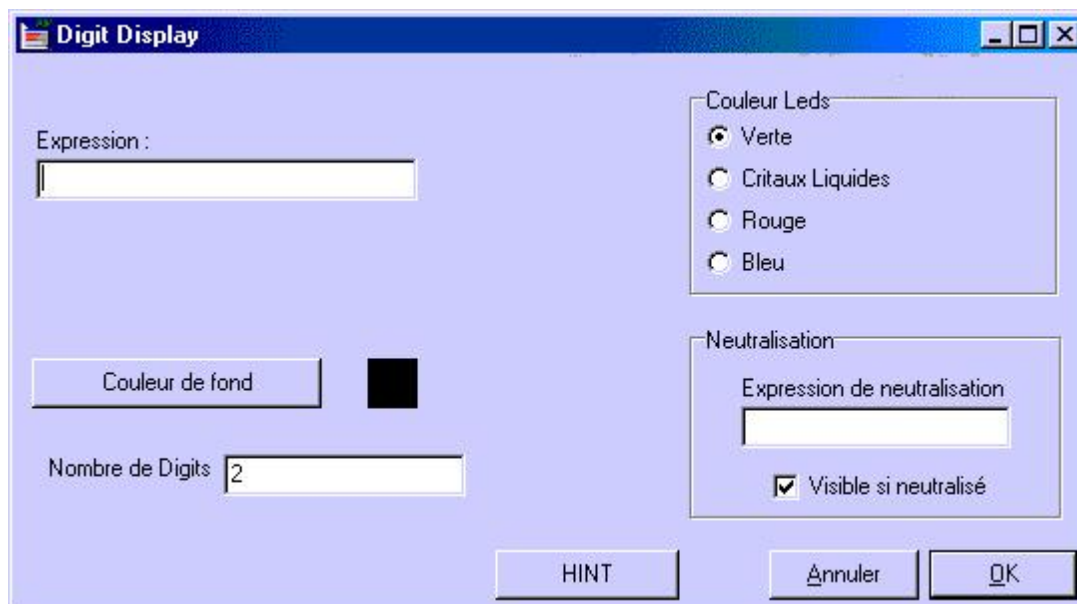
Cette variable doit être du type numérique entière ou décimale.

Il est possible de définir les couleurs de ces segments ainsi que la couleur de fond.

Exemple d'afficheur 7 segments :



**Boîte de dialogue « Afficheur 7 segments » :**



Les paramètres sont :

### Expression

Indiquer dans ce champ le nom de la variable à visualiser avec l'afficheur. Cette variable doit être de type numérique entier (Registre).

*Exemple : R12*

Il est possible de spécifier l'adresse IP distante du noyau APIGRAF PLC dans l'expression. Si c'est le cas, c'est cette adresse qui sera prise en compte pour le rafraîchissement.

*Exemple : !193.0.1.1 !R12*

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

### Couleur de Fond

Spécifie la couleur de fond de l'afficheur. Si la couleur des leds est « cristaux liquides » alors ce paramètre est neutralisé et prend la couleur gris par défaut.

### Nombre de Digits

Indique le nombre de chiffre que l'afficheur aura. Ce paramètre doit être un nombre entier.

Exemple : Variable allant de 0 à 99, mettre 2 dans le nombre de digits.

### Groupe radio Couleur Led

Ce paramètre permet de sélectionner la couleur des leds en cochant la case en face de la couleur désirée. Si la case « cristaux liquides » est sélectionné alors les leds sont noire sur un fond gris.

### Expression de neutralisation

Permet de spécifier la variable ou l'expression de neutralisation (Celle-ci doit obligatoirement être de type booléen).

*Exemple : SA10*

Il est possible de spécifier l'adresse IP distante du noyau APIGRAF PLC dans l'expression. Si c'est le cas, c'est cette adresse qui sera prise en compte pour le rafraîchissement.

*Exemple : !193.0.1.1 !SA10*

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1}) Voir Section « Appel d'une page de supervision avec passage de paramètres »)

#### Option « visible si neutralisé »

Si cette option n'est pas cochée, l'afficheur deviendra invisible lorsqu'il sera neutralisé.

#### Bouton « Hint »

Ce bouton permet de définir les paramètres de « Hint » de l'objet. Le texte « Hint » que l'on définit est le texte qui apparaît sur fond jaune lorsque l'on passe le curseur de la souris sur l'objet.

Quand on clique sur le bouton « Hint », la fenêtre suivante s'ouvre :



Cette fenêtre présente deux contrôles :

#### Option « Afficher le Hint »

Si cette option n'est pas cochée, le texte du Hint n'apparaîtra pas quand le curseur de la souris passera sur l'objet même si celui-ci est défini. Il faut donc cocher cette option si l'on veut utiliser le « Hint » de l'objet.

#### Fenêtre de saisie « Hint »

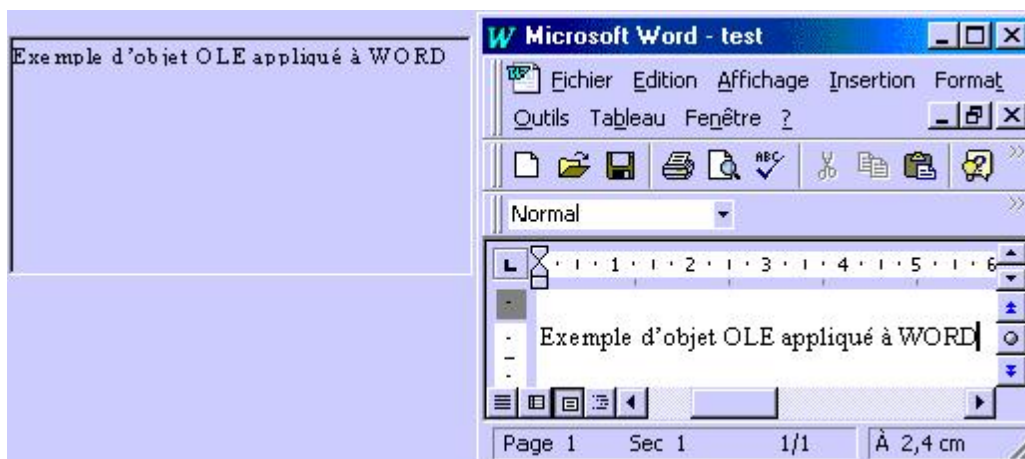
Cette fenêtre permet d'entrer le texte du Hint. Ce texte peut être un paramètre (exemple {1}).

### 3.2.6.19.3 OLE Container

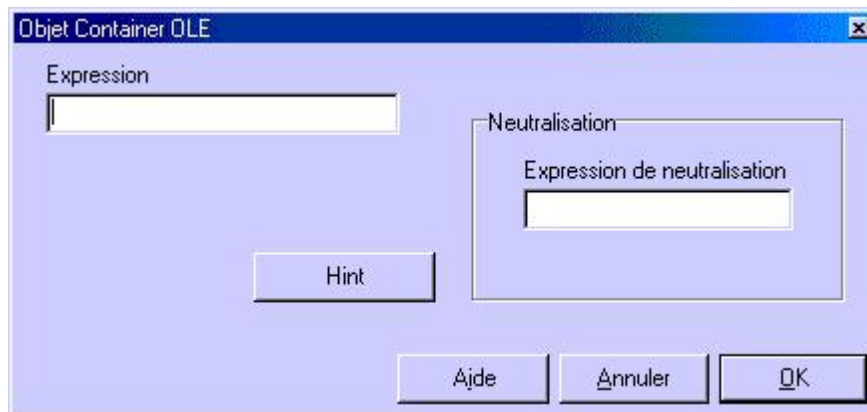
Le composant OLE Container (conteneur OLE) dote votre application de capacités lui permettant d'effectuer des liaisons et incorporations d'objets se trouvant sur un serveur OLE.

Lorsque vous activez un objet se trouvant dans un conteneur OLE, le contrôle passe à l'application serveur OLE. L'utilisateur peut alors accéder à toutes les fonctionnalités de l'application serveur à partir de votre application conteneur.

Exemple de conteneur OLE :



#### Boîte de dialogue «Objet OLE Container » :



Les paramètres sont :

### Expression

Permet de définir le nom du fichier qu'il faut insérer via un conteneur OLE

Indiquer dans ce champ la variable automate à utiliser (de type chaîne)

*Exemple : C20*

Il est possible de spécifier l'adresse IP distante du noyau APIGRAF PLC dans l'expression. Si c'est le cas, c'est cette adresse qui sera prise en compte pour le rafraîchissement.

*Exemple : !193.0.1.1 !C20*

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)/

### Expression de neutralisation

Permet de spécifier la variable ou l'expression de neutralisation (Celle-ci doit obligatoirement être de type booléen).

*Exemple : SA10*

Il est possible de spécifier l'adresse IP distante du noyau APIGRAF PLC dans l'expression. Si c'est le cas, c'est cette adresse qui sera prise en compte pour le rafraîchissement.

*Exemple : !193.0.1.1 !SA10*

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

### Option « visible si neutralisé »

Si cette option n'est pas cochée, l'objet deviendra invisible lorsqu'il sera neutralisé.

### Bouton « Hint »

Ce bouton permet de définir les paramètres de « Hint » de l'objet. Le texte « Hint » que l'on définit est le texte qui apparaît sur fond jaune lorsque l'on passe le curseur de la souris sur l'objet.

Quand on clique sur le bouton « Hint », le fenêtre suivante s'ouvre :



Cette fenêtre présente deux contrôles :

**Option « Afficher le Hint »**

Si cette option n'est pas cochée, le texte du Hint n'apparaîtra pas quand le curseur de la souris passera sur l'objet même si celui-ci est défini. Il faut donc cocher cette option si l'on veut utiliser le « Hint » de l'objet.

**Fenêtre de saisie « Hint »**

Cette fenêtre permet d'entrer le texte du Hint. Ce texte peut être un paramètre (exemple {1}).

### 3.2.6.19.4 Gestionnaire d'évènement

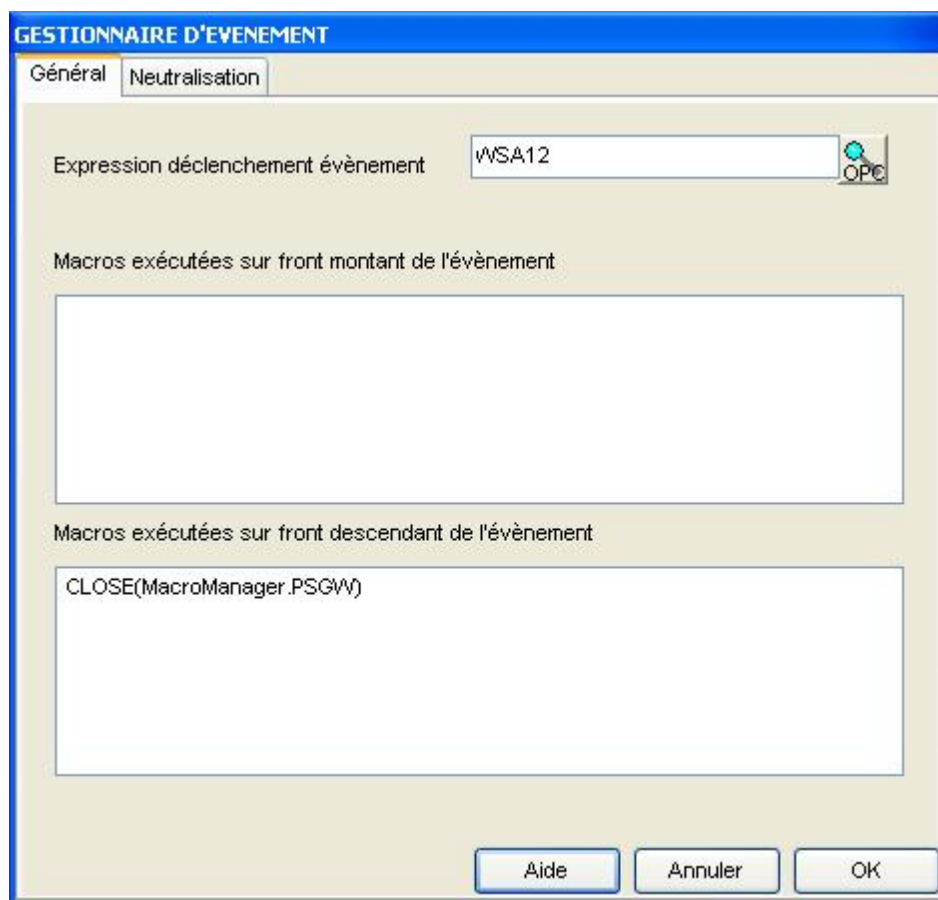
L'objet gestionnaire d'évènement permet d'exécuter des macros prédéfinies sur un changement d'état d'une expression ou d'une variable automate.

Il est possible de définir des macros à exécuter sur le front montant du changement d'état ou sur le front descendant du changement d'état.

Cet objet est uniquement visible durant la phase de conception.



**Boîte de dialogue «Gestionnaire d'évènement» - Onglet général :**



Les paramètres sont :

### Expression

Permet de définir l'expression ou la variable automate qui va générer l'évènement. Le type de l'expression ou de la variable doit donc être booléen.

*Exemple : SA10*

Il est possible de spécifier l'adresse IP distante du noyau OPTIMA PLC ou APIGRAF PLC dans l'expression. Si c'est le cas, c'est cette adresse qui sera prise en compte pour le rafraîchissement.

*Exemple : !193.0.1.1 !SA10*

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)/

Si la variable provient d'un serveur OPC, le bouton "OPC" permet d'explorer le serveur OPC et de trouver la variable à superviser dans un explorateur.

### Macros exécutées sur front montant de l'évènement

Entrer ici les différentes macros à exécuter quand la valeur de l'expression passe à 1 ( ou True)  
Attention : aller à la ligne pour chaque nouvelle macro.

### Macros exécutées sur front descendant de l'évènement

Entrer ici les différentes macros à exécuter quand la valeur de l'expression passe à 0 ( ou False)

### Boîte de dialogue «Gestionnaire d'évènement» - Onglet Neutralisation :

Les paramètres sont :

### Expression de neutralisation

Permet de spécifier la variable ou l'expression de neutralisation (Celle-ci doit obligatoirement être de type booléen).



Exemple : SA10

Il est possible de spécifier l'adresse IP distante du noyau APIGRAF PLC dans l'expression. Si c'est le cas, c'est cette adresse qui sera prise en compte pour le rafraîchissement.

Exemple : !193.0.1.1 !SA10

Il est possible de rentrer une référence à un paramètre dans ce champ (exemple : {1} Voir Section « Appel d'une page de supervision avec passage de paramètres »)

Si la variable provient d'un serveur OPC, le bouton "OPC" permet d'explorer le serveur OPC et de trouver la variable à superviser dans un explorateur.

Les évènements ne sont plus déclenchés si la valeur de l'expression de neutralisation est égale à 1 ( ou True )

### Exemple d'utilisation typique de cet objet

On crée une page TEST.PSGW, on veut que cette page se ferme automatiquement à partir de l'automate. On insère un objet Gestionnaire d'évènement dans la page. On entre la valeur de l'expression SA20 et la macro exécutée sur front montant de l'évènement : CLOSE(TEST.PSGW)

Dès que l'automate OPTIMA PLC va passer SA20 à TRUE, la fenêtre de supervision va se fermer.

## 3.2.7 Explorateur d'objets

### 3.2.7.1 Utilisation de l'explorateur d'objets

L'explorateur d'objets est accessible depuis le menu "Affichage" -> "Explorateur d'objets"

L'explorateur permet de visualiser sous forme de liste hiérarchisée tous les objets présents sur une page de supervision.

Après avoir ouvert une page de supervision, on peut ouvrir l'explorateur d'objets :



Il suffit de cliquer avec le bouton droit de la souris pour sélectionner un objet dans la liste. Une fois sélectionné dans la liste, l'objet correspondant de la page de supervision apparaît également sélectionné.

L'Explorateur d'objet permet de réaliser plusieurs opérations :

[Masquer un objet en conception](#) <sup>295</sup>

[Masquer tous les objets en conception](#) <sup>296</sup>

[Annuler le masquage d'un objet en conception](#) <sup>296</sup>

[Annuler le masquage de tous les objets en conception](#) <sup>296</sup>

[Configurer un objet](#) <sup>296</sup>

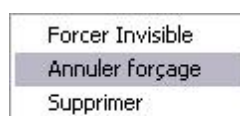
[Supprimer un objet](#) <sup>297</sup>

### 3.2.7.2 Masquer un objet en conception

Ouvrir l'explorateur d'objet :



Faire un click droit avec la souris sur l'objet dans la liste, le menu contextuel suivant apparaît :



Choisir le menu "Forcer invisible". L'objet correspondant n'est plus visible sur la page de supervision.

### 3.2.7.3 Masquer tous les objets en conception

Ouvrir l'explorateur d'objet :



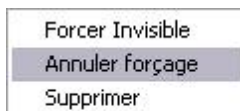
Appuyer sur le bouton "Tous invisibles". Tous les objets de la page de supervision ne sont plus visibles.

### 3.2.7.4 Annuler le masquage d'un objet en conception

Ouvrir l'explorateur d'objet :



Faire un click droit avec la souris sur l'objet dans la liste, le menu contextuel suivant apparaît :



Choisir le menu "Annuler Forçage". L'objet correspondant redevient visible sur la page de supervision.

### 3.2.7.5 Annuler le masquage de tous les objets en conception

Ouvrir l'explorateur d'objet :



Appuyer sur le bouton "Annuler Forçages". Tous les objets de la page de supervision qui n'étaient plus visibles redeviennent visibles.

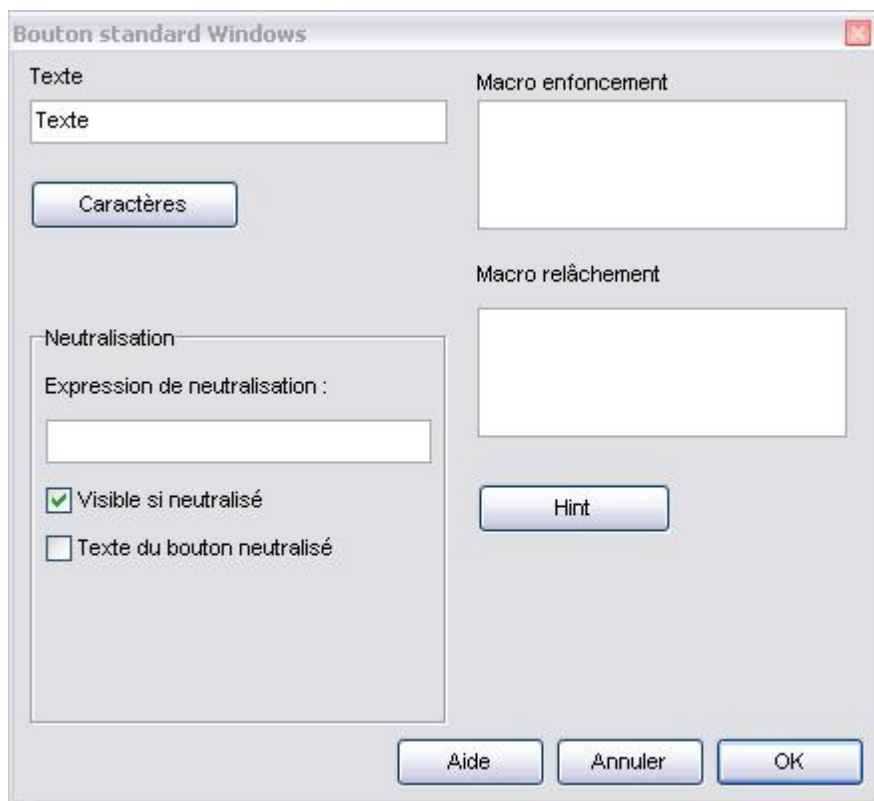
### 3.2.7.6 Configurer un objet

Ouvrir l'explorateur d'objet :



Sélectionner un objet dans la liste.

Appuyer sur le bouton "Configurer". La fenêtre de paramétrage de l'objet s'ouvre :



### 3.2.7.7 Supprimer un objet

Ouvrir l'explorateur d'objet :



Sélectionner l'objet à supprimer dans la liste.

Appuyer sur le bouton "Supprimer". L'objet est alors supprimé de la page de supervision.

### 3.2.8 Appel d'une page de supervision depuis une autre page

L'appel d'une page de supervision depuis une autre page se fait en utilisant la macro « SUPERG ». (Voir le chapitre dédié à l'utilisation des macros).

*Exemple:* Appel de la page « Page2.psgw »

Si le fichier page2.pswg correspond à une page de supervision du projet alors l'appel de cette page depuis une autre peut se faire de la manière suivante :

1. Créer un bouton simple sur la page principale ( page depuis laquelle on fait l'appel )
2. Double-cliquer sur le bouton pour accéder à sa page de paramétrage.
3. Dans la fenêtre « Macro relâchement », taper le texte suivant : SUPERG(PAGE2)
4. En mode exécution, l'enfoncement du bouton simple aura donc pour effet de charger la page « Page2 » à l'écran.

*Remarque:* Il est possible d'appeler une page de supervision en lui passant des paramètres. (Voir section suivante « [Appel d'une page de supervision avec passage de paramètres](#) »).

### 3.2.9 Appel d'une page de supervision avec passage de paramètres

Une page de supervision peut être appelée avec des paramètres. Procéder comme suit :

- Dans la page, insérer des références à des paramètres avec {1}, {2}, etc.
- Appeler la page en précisant les paramètres à la suite du nom de fichier (SUPERG(PAGE,SA10,R20))

*Exemple :*

Une page TEST.PSG contient deux boutons API qui doivent être liés à deux variables booléennes dont les noms sont connus seulement au moment de l'appel de la page, par exemple SA10 et SA11.

Dans le premier bouton, à la place de l'expression classique, entrer {1}.

Dans le second bouton, à la place de l'expression classique, entrer {2}.

Dans la macro qui appelle la page, entrer SUPERG(TEST,SA10,SA20) pour remplacer dans la page test {1} par SA10 et {2} par SA20.

*Autre exemple d'application :*

Le passage de paramètres peut être utilisé dans le cas où l'utilisateur veut pouvoir ouvrir la même page de supervision en supervisant les données de différents postes PLC.

Dans ce cas, il faut dans les options de la page, dans le champ «Adresse IP par défaut », entrer {1}.

Il suffit ensuite d'appeler la page de supervision en passant l'adresse IP à superviser en paramètre :

SUPERG(PAGE1,193.0.1.1)

Ou SUPERG(PAGE1,193.0.1.2)

### 3.2.10 Fermeture d'une page depuis un bouton

Pour fermer la page de supervision en cours, on peut utiliser la macro « CLOSE ». (Voir chapitre dédié aux macros).

1. Créer un bouton simple sur la page.
2. Double-cliquer sur le bouton pour accéder à sa page de paramétrage.
3. Dans la fenêtre « Macro relâchement », taper le texte suivant : CLOSE
4. En mode exécution, l'enfoncement de ce bouton aura pour effet de fermer la page.

### 3.2.11 Gestion des mots de passe

L'utilisateur a la possibilité d'utiliser des mots de passe pour permettre l'ouverture de certaines pages de supervision.

Le mot de passe est défini par un numéro.

Pour créer un modifier un mot de passe, on fait appel à la macro ChnMDP.

Exemple :

1. Dans une page de supervision, créer un bouton simple.
2. Dans la macro de relâchement du bouton simple, taper la macro : ChnMDP(1). Enregistrer la page sous le nom « page1 »
3. En exécution le fait d'appuyer sur ce bouton va ouvrir une boîte de dialogue permettant de définir ou de modifier si celui-ci existe, le mot de passe numéro 1
4. Créer une nouvelle page de supervision et l'enregistrer sous le nom « page2 »
5. Dans les propriétés de la page nommée « page2 », choisir l'onglet OPTIONS. Saisir 1 dans la boîte « Mot de passe à l'ouverture de la page ».
6. Dans la page1, créer un autre bouton simple. Entrer la macro de relâchement : SUPERG(page2)
7. En exécution, appuyer sur le bouton simple précédent. Si le mot de passe 1 a été défini, celui-ci sera demandé avant d'ouvrir la page 2

## 3.3 Objets personnalisés

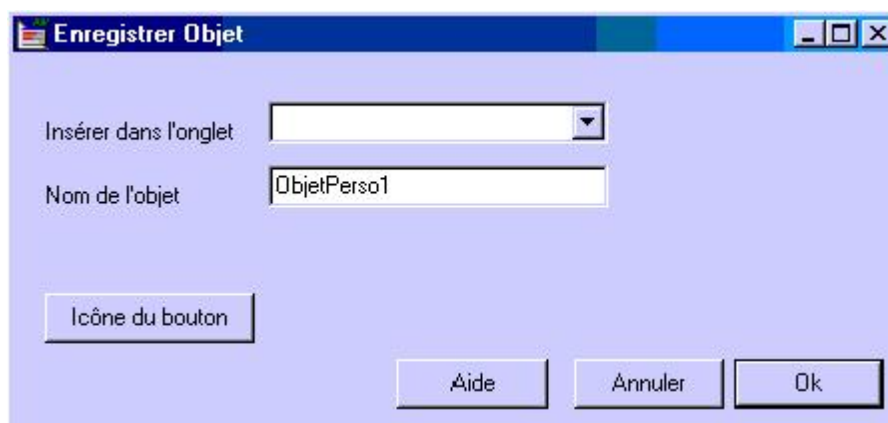
### 3.3.1 Enregistrement d'objets Personnalisés

Il est possible d'enregistrer des objets personnalisés, qui sont en faite des objets classiques avec des caractéristiques définies (couleurs, expression, taille...).

Un bouton sera créé dans un onglet existant ou non qui permettra de rappeler cet objet.

Pour enregistrer un objet il faut le sélectionner et cliquer dans le menu «objet » sur «ajouter », Il est possible aussi de faire un clic droit sur l'objet pour faire apparaître le menu contextuel et de cliquer sur «enregistrer objet ».

Le menu ouvre la fenêtre suivante :



#### Insérer dans l'onglet:

Cette case relate tous les onglets existants en cliquant sur la flèche à droite de la case pour inserer le bouton de l'objet à enregistrer. Il est possible de sélectionner un onglet dans la liste ou de taper le nom de l'onglet que vous voulez créé.

#### Nom de l'objet:

Cette case permet de rentrer le nom du bouton qui s'affichera dans le Hint quand le curseur passera au-dessus. Il défini aussi le nom du fichier ou sera enregistré l'objet:

(ex: User0ObjetPerso0.bib)

#### Icône du bouton :

Le bouton aura une icône par défaut:

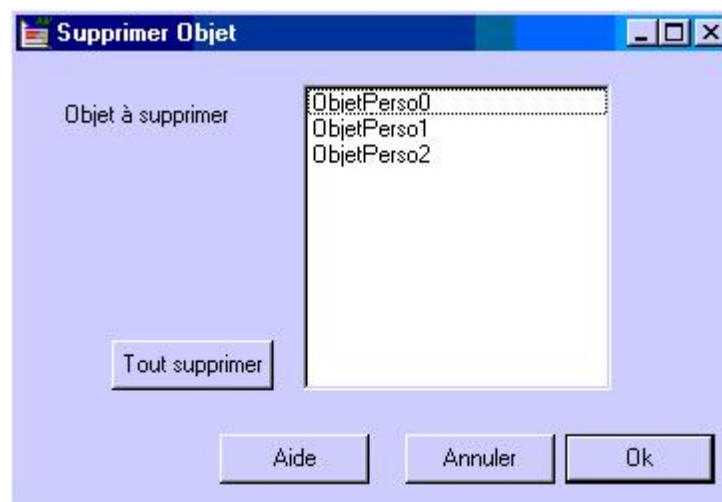


Il est possible de personnaliser l'image de l'icône en cliquant sur le bouton et en choisissant un fichier image valide. Un aperçu de l'image est disponible dans cette boîte de dialogue.

### 3.3.2 Suppression d'objets personnalisés

Il est possible d'effacer des objets personnalisés. Cela effacera le bouton, l'onglet créé s'il n'a plus de bouton dedans ainsi que le fichier de l'objet enregistrer.

Pour cela, sélectionnez un objet et faite «supprimer » dans le menu «objet ».  
Ce menu ouvre la fenêtre suivante :



Il suffit de sélectionner dans le liste les objets que vous voulez supprimer et de faire 'Ok'.  
Les paramètres sont les suivant:

#### **Objet à supprimer:**

Cette liste relate tous les objets qui ont été enregistrés (objets personnalisés).

Il suffit de sélectionner le ou les objets à supprimer. Pour sélectionner plusieurs objets, cliquer sur un objet et faite glisser la souris jusqu'au dernier objet que vous voulez supprimer ou cliquer sur ceux que vous voulez en restant appuyer sur la touche "Control".

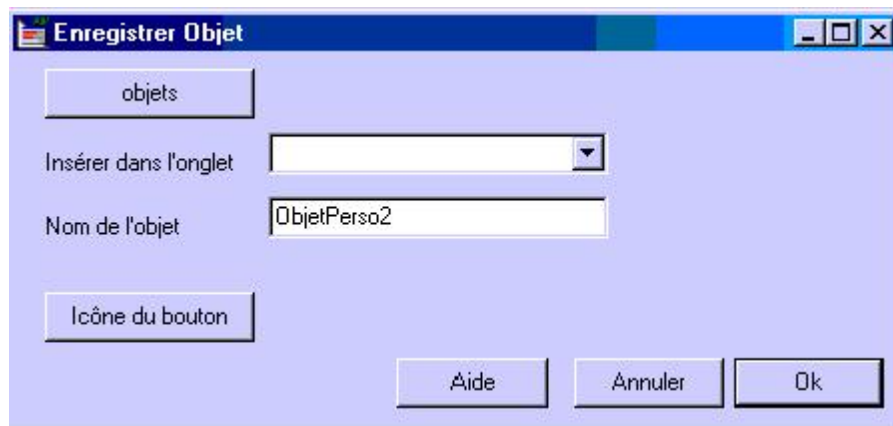
#### **Tout sélectionner:**

Permet de sélectionner tous les objets de la liste.

### 3.3.3 Importation d'objets personnalisés

Il est possible d'importer des objets personnalisés grâce aux fichiers d'extension « bib ». Ces fichiers se trouvent dans « APIGRAFIP\BIN\PERSO\ ».

Pour cela, faites « importer » dans le menu « Objet ».  
Ce menu ouvre la fenêtre suivante :



Les différents paramètres de cette fenêtre sont:

**Objets:**

Ce bouton permet de choisir le fichier correspondant à l'objet à importer. Ce fichier doit avoir l'extension "bib".

**Insérer dans l'onglet:**

Cette case relate tous les onglets existants en cliquant sur la flèche à droite de la case pour insérer le bouton de l'objet à enregistrer. Il est possible de sélectionner un onglet dans la liste ou de taper le nom de l'onglet que vous voulez créer.

**Nom de l'objet:**

Cette case permet de rentrer le nom du bouton qui s'affichera dans le Hint quand le curseur passera au-dessus. Il définit aussi le nom du fichier ou sera enregistré l'objet:  
(ex: User0ObjetPerso0.bib)

**Icône du bouton :**

Le bouton aura une icône par défaut:

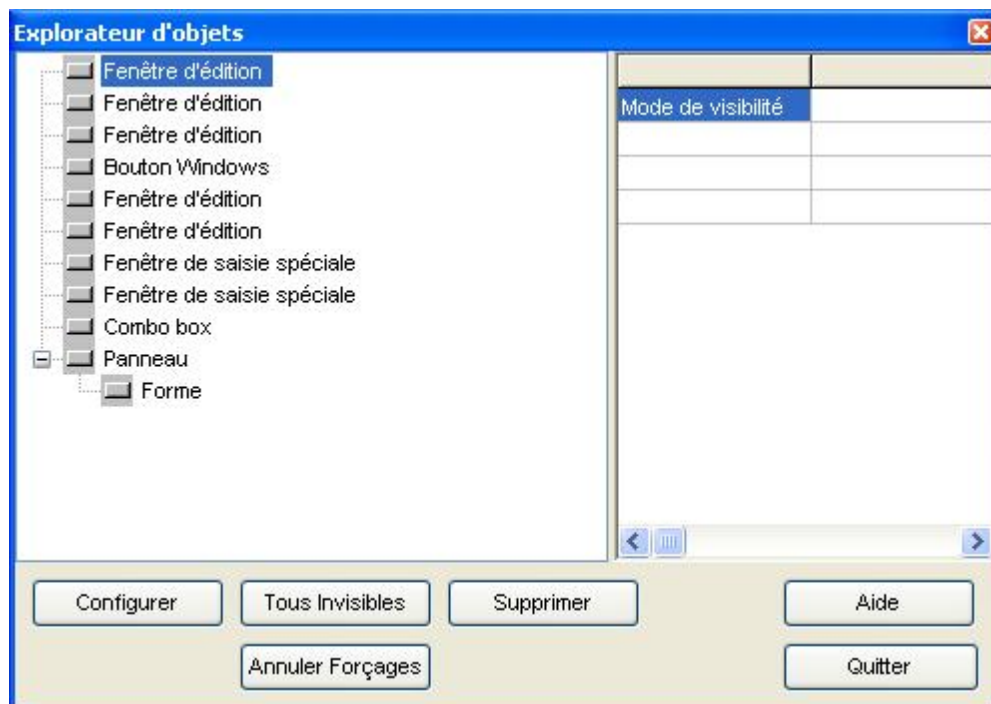


Il est possible de personnaliser l'image de l'icône en cliquant sur le bouton et en choisissant un fichier image valide. Un aperçu de l'image est disponible dans cette boîte de dialogue.

## 3.4 Explorateur d'objets

### 3.4.1 Explorateur d'objets

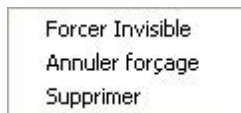
L'explorateur d'objets liste dans une fenêtre arborescente tous les objets présents dans la page de supervision courante. Il est possible depuis cette page de forcer un objet à être invisible pendant la phase de conception de la page de supervision.



Sélectionner un objet dans la liste à l'aide de la souris, l'objet correspondant apparaît alors sélectionné dans la page de supervision.

### **Forcer un objet à être invisible**

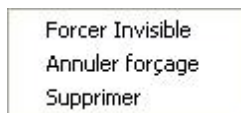
Sélectionner un objet dans la liste arborescente et faire un clic droit dessus puis choisir "Forcer invisible" dans le menu contextuel :



L'objet n'est plus visible.

### **Annuler Forcer un objet à être invisible**

Sélectionner un objet dans la liste arborescente et faire un clic droit dessus puis choisir "Annuler forçage" dans le menu contextuel :



L'objet est visible.

#### **• Bouton "Annuler Forçages"**

Annule tous les forçages d'invisibilité. Tous les objets sont alors visibles dans la page de supervision courante.

#### **• Bouton "Configurer"**

Sélectionner un objet dans la liste arborescente et appuyer sur le bouton "Configurer" pour faire apparaître la fenêtre de configuration de l'objet.

#### **• Bouton "Supprimer"**

Sélectionner un objet dans la liste arborescente et appuyer sur le bouton "Supprimer" pour supprimer l'objet sélectionné.

- **Bouton "Tous invisibles"**

Force l'invisibilité de tous les objets de la page courante.

## 3.5 Utilisation des macros

### 3.5.1 Macros locales

Les macros locales sont exécutées sur le poste de supervision (poste Windows®)

#### 3.5.1.1 BringToFrontSuperGWindow (Macro locale)

Met en avant plan la page de supervision passée en paramètre à condition que la page soit déjà ouverte (avec la macro SUPERG ou éventuellement cachée). Si aucune page n'est passée en paramètre, la page courante est mise en avant plan.

La macro est généralement appelée depuis les macros de relâchement d'un bouton (type bouton de fonction ou autre).

BRINGTOFRONTSUPERGWINDOW

Met en avant plan la fenêtre courante

BRINGTOFRONTSUPERGWINDOW(TEST)

BRINGTOFRONTSUPERGWINDOW(TEST.PSGW)

BRINGTOFRONTSUPERGWINDOW(C:\REP1\TEST)

Affiche la page Test

Remarque : Si la page ne se trouve pas dans le repertoire courant du projet, il faut spécifier le chemin complet de la page. Si le chemin comporte des espaces (comme dans "Program Files"), il faut mettre le nom du fichier entre doubles quotes :

BRINGTOFRONTSUPERGWINDOW("C:\Program Files\REP1\TEST")

La macro BRINGTOFRONTSUPERGWINDOW est utilisée généralement conjointement avec les macros HIDESUPERGWINDOW et SHOWSUPERGWINDOW qui permettent d'afficher et de cacher une fenêtre pour permettre à l'utilisateur de gérer les affichages des fenêtres déjà ouvertes.

#### 3.5.1.2 ChnMdp (Macro locale)

Permet de modifier ou de créer un mot de passe dans le projet.

Le mot de passe ainsi créé peut alors être spécifié comme mot de passe d'ouverture d'une page de supervision (Voir Onglet Options des propriétés de la page de supervision)

Exemple :

ChnMDP(1)

Si le mot de passe 1 n'existe pas, la macro demandera la saisie du mot de passe 1.

Si celui-ci existe, l'ancien mot de passe devra être saisi pour pouvoir être modifié.

#### 3.5.1.3 ChnUserLanguage (Macro locale)

Cette macro permet de basculer le langage des pages de supervision dans le cas où des fichiers de langage ont été défini dans le projet.

Cette macro est généralement placée dans la macro relâchement d'un bouton simple.

Exemple :

CHNUSERLANGUAGE(2)

#### 3.5.1.4 Close (Macro locale)

Fermeture de la fenêtre de supervision courante.



## CLOSE

Dans les objets de supervision de type bouton, cette macro ne doit être utilisée qu'au moment du relâchement.

Il est possible de spécifier un nom de page en paramètres. Dans ce cas c'est la page nommée en paramètre (si celle-ci existe) qui sera fermée et non la page courante.

Exemples :

```
CLOSE (TEST.PSGW)
CLOSE (TEST)
CLOSE(C:\REP1\TEST)
CLOSE("C:\Program Files\REPA\TEST")
```

Remarque : Si la page ne se trouve pas dans le repertoire courant du projet, il faut spécifier le chemin complet de la page. Si le chemin comporte des espaces (comme dans "Program Files"), il faut mettre le nom du fichier entre doubles quotes :

```
CLOSE("C:\Program Files\REPA\TEST")
```

### 3.5.1.5 CopyPageBitmap (Macro locale)

Fait une copie d'écran de la page courante dans un fichier bitmap.

#### COPYPAGEBITMAP

La macro fait une copie d'écran de la page de supervision courante dans le répertoire des fichiers PSGW. Cette macro est généralement depuis la macro de relachement d'un objet de supervision de type bouton.

Exemple :

Pour une page TEST.PSGW, un fichier PAGECOPY-TEST-1.BMP sera créé puis un fichier PAGECOPY-TEST-2.BMP, etc.

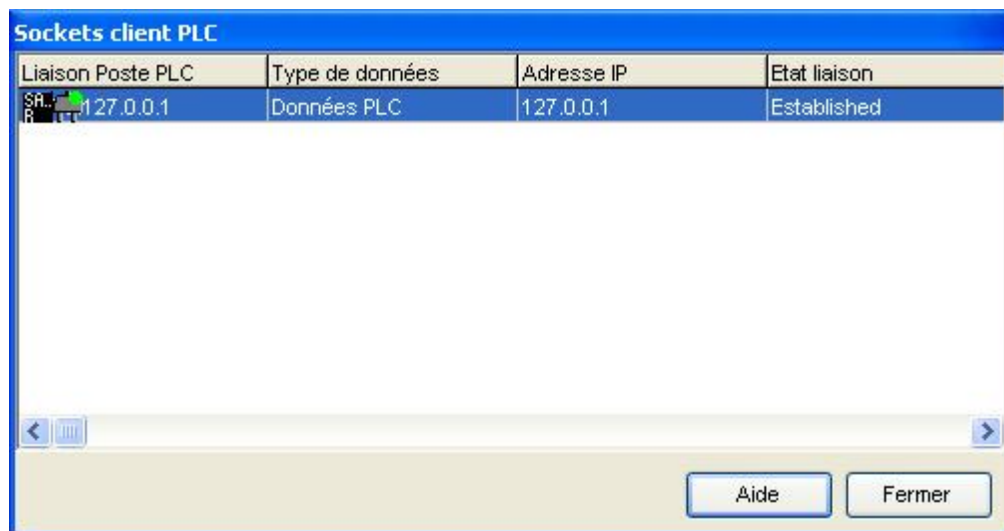
Il est possible de spécifier un nom de répertoire en paramètre :

COPYPAGEBITMAP(C:\COURBES) copie les bitmaps dans le rép C:\COURBES


COPYPAGEBITMAP(\COURBES) copie les bitmaps dans le rép C:\...\PSGW\COURBES (sous répertoire COURBES du projet de supervision)

### 3.5.1.6 DiagSocketPLC

Appel de la fenêtre de diagnostique des sockets PLC.



Pour plus de détails, consulter l'aide du menu :

[Clients TCP/IP Données PLC](#) 

### 3.5.1.7 ExecProg (Macro locale)

Cette macro permet d'exécuter une application.

Exemple :

EXECProg(C:\WINDOWS\CALC.EXE)

(Voir le projet d'exemple : \EXEMPLES\MACROS)

Il est possible de passer un paramètre en ligne de commande :

Exemple :

EXECProg(C:\WINDOWS\notepad.exe,C:\ESSAI.TXT)

Cette macro permet d'ouvrir le fichier NOTEPAD.EXE avec NOTEPAD.

### 3.5.1.8 HideCursor (Macro locale)

Cette macro permet de rendre invisible le pointeur de la souris dans la fenêtre de supervision.

Cette macro est utilisée pour les écrans tactiles quand on ne veut pas avoir à l'écran le pointeur de la souris.

Pour réafficher à nouveau le pointeur de la souris, il faut appeler la macro ShowCursor.

### 3.5.1.9 HideSuperGWindow (Macro locale)

Cache la page de supervision passée en paramètre. Si aucune page n'est passée en paramètre, la page courante est cachée.

La macro est généralement appelée depuis les macros de relâchement d'un bouton (type bouton de fonction ou autre).

HIDESUPERGWINDOW

Cache la fenêtre courante

HIDESUPERGWINDOW(TEST)

HIDESUPERGWINDOW(TEST.PSGW)

HIDESUPERGWINDOW(C:\REP1\TEST)

Remarque : Si la page ne se trouve pas dans le repertoire courant du projet, il faut spécifier le chemin complet de la page. Si le chemin comporte des espaces (comme dans "Program Files"), il faut mettre le nom du fichier entre doubles quotes :

HIDESUPERGWINDOW("C:\Program Files\REP1\TEST")

### 3.5.1.10 PrintPage (Macro locale)

Cette macro permet d'imprimer la page de supervision en cours. Il n'y a pas de paramètres.

Exemple :

PRINTPAGE

### 3.5.1.11 RunHelp (Macro locale)

Cette macro permet d'exécuter un fichier d'aide utilisateur. Le fichier d'aide est ouvert au niveau du menu général. Ce fichier contient l'extension \*.HLP. Elle lance le menu général du fichier d'aide passé en paramètre. Cette macro est généralement placée dans la macro relâchement d'un bouton simple.

Exemple :

RunHelp(USERHELP.HLP)

Cet exemple ouvre le fichier d'aide USERHELP.HLP à condition que celui-ci soit présent dans le répertoire du projet. Si le fichier d'aide à ouvrir se trouve dans un autre répertoire que le répertoire du projet, il faut spécifier le chemin complet du fichier d'aide.

Exemple :

RunHelp(C:\HELP\USERHELP.HLP)

### 3.5.1.12 RunHelpIndex (Macro locale)

Cette macro permet d'ouvrir un fichier d'aide directement au niveau de l'outil de recherche d'aide ou index. Cette macro est généralement placée dans la macro relâchement d'un bouton simple.

Exemple :

RunHelpIndex(USERHELP.HLP)

Si le fichier d'aide à ouvrir se trouve dans un autre répertoire que le répertoire du projet, il faut spécifier le chemin complet du fichier d'aide.

Exemple :

RunHelpIndex(C:\HELP\USERHELP.HLP,23)

### 3.5.1.13 RunHelpTopic (Macro locale)

Cette macro permet de gérer une aide contextuelle dans un projet. Elle ouvre en effet un fichier d'aide passé en paramètre sur un numéro de rubrique (TOPIC Num ou MAP Number) passé en paramètre. Cette macro est généralement placée dans la macro relâchement d'un bouton simple.

Exemple :

RunHelpTopic(USERHELP.HLP,23)

Le premier paramètre désigne le fichier d'aide.

Le deuxième paramètre désigne le numéro de rubrique de l'aide à ouvrir.

Si le fichier d'aide à ouvrir se trouve dans un autre répertoire que le répertoire du projet, il faut spécifier le chemin complet du fichier d'aide.

Exemple :

RunHelpTopic(C:\HELP\USERHELP.HLP,23)

Si toutefois la rubrique n'est pas trouvée dans le fichier d'aide, un message windows® avertira l'utilisateur.

### 3.5.1.14 ShowCursor (Macro locale)

Cette macro permet de réafficher le pointeur de la souris dans la fenêtre de supervision. Elle est utilisée suite à un appel à la macro HideCursor qui rend invisible le pointeur de la souris sur la page ou la macro a été appelée.

Cette macro est utilisée pour les écrans tactiles quand on ne veut pas avoir à l'écran le pointeur de la souris.

### 3.5.1.15 ShowSuperGWindow (Macro locale)

Affiche la page de supervision passée en paramètre à condition que la page soit déjà ouverte (avec la macro SUPERG ou éventuellement cachée). Si aucune page n'est passée en paramètre, la page courante est affichée. La macro est généralement appelée depuis les macros de relâchement d'un bouton (type bouton de fonction ou autre).

SHOWSUPERGWINDOW

Affiche la fenetre courante

```
SHOWSUPERGWINDOW(TEST)
```

```
SHOWSUPERGWINDOW(TEST.PSGW)
```

```
SHOWSUPERGWINDOW(C:\REP1\TEST)
```

Affiche la page Test

*Remarque* : Si la page ne se trouve pas dans le repertoire courant du projet, il faut spécifier le chemin complet de la page. Si le chemin comporte des espaces (comme dans "Program Files"), il faut mettre le nom du fichier entre doubles quotes :

```
SHOWSUPERGWINDOW("C:\Program Files\REP1\TEST")
```

La macro SHOWSUPERGWINDOW est utilisée généralement conjointement avec la macro HIDESUPERGWINDOW qui permet de cacher une fenêtre pour permettre à l'utilisateur de gérer les affichages des fenêtres déjà ouvertes.

### 3.5.1.16 SuperG (Macro locale)

Appel d'une page de supervision. Pour appeler par exemple la page de supervision stockée dans le fichier ESSAI.PSWG, utiliser la macro : SUPERG(ESSAI)

### 3.5.1.17 VisuText (Macro locale)

Permet de visualiser dans une fenêtre un fichier de type texte.

*Exemple :*

```
VisuText(C:\File1.Txt)
```

### 3.5.1.18 WFCopy (Macro locale)

Cette macro permet de copier un ou plusieurs fichiers locaux.

Le premier paramètre désigne le ou les fichiers source.

Le deuxième paramètre désigne le fichier de destination.

*Exemples :*

```
WFCOPY (C:\ESSAI.TXT,C:\ESSAI.BAK)
```

```
WFCOPY(C:\REP1\*.TXT,C:\REP2\*.NEW)
```

### 3.5.1.19 WFCreatedirectory (Macro locale)

Cette macro permet de créer un répertoire local.

Le premier paramètre indique le nom du répertoire à créer.

*Exemple :*

```
WFCREATEDIRECTORY(C:\REP1)
```

### 3.5.1.20 WFDelete (Macro locale)

Cette macro permet de supprimer un ou plusieurs fichiers locaux.

Le premier paramètre indique le nom du ou des fichiers à supprimer.

*Exemples :*

```
WFDELETE(C:\REP1\FILE1.TXT)
```

```
WFDELETE(C:\REP1\*.TXT)
```

### 3.5.1.21 WFDeleteDirectory (Macro locale)

Cette macro permet de supprimer un répertoire local ainsi que son contenu.

Le premier paramètre indique le nom du répertoire à supprimer.

*Exemple :*

```
WFDELETEDIRECTORY(C:\REP1)
```

### 3.5.1.22 WFRename (Macro locale)

Cette macro permet de renommer un fichier ou un répertoire local.

Le premier paramètre désigne le nom du fichier ou du répertoire à renommer.

Le deuxième paramètre indique le nouveau nom du fichier ou du répertoire.

Exemples :

WFRENAME(C:\REP1\FILE1.TXT,C:\REP1\NEWFILE1.TXT)

WFRENAME(C:\OLDREP,C:\NEWREP)

**3.5.1.23 WFSetsFileAttr (Macro locale)**

Cette macro permet de modifier les attributs d'un fichier local.

Le premier paramètre désigne le nom du fichier auquel appliquer les nouveaux attributs.

Le deuxième paramètre indique l'attribut à appliquer au fichier. Ce paramètre est de type entier.

Exemples :

WFSETFILEATTR(C:\REP1\FILE1.TXT,36)

Cette macro transforme le fichier File1.txt en fichier système (fichier caché).

WFSETFILEATTR(C:\REP1\FILE1.TXT,32)

Cette macro rend les attributs normaux au fichier File1.Txt (Attributs : normaux)

**3.5.1.24 WNetSChangeFileName (Macro locale)**

Cette macro permet de renommer un fichier ou un répertoire sur un poste PLC.

Le premier paramètre est optionnel, il désigne l'adresse IP du poste PLC.

Le deuxième paramètre indique l'ancien nom de fichier.

Le troisième paramètre indique le nouveau nom de fichier.

Exemples :

WNETSCCHANGEFILENAME (193.0.1.1,C:\OLD.TXT,C:\NEW.TXT)

WNETSCCHANGEFILENAME (C:\OLD.TXT,C:\NEW.TXT)

Le deuxième exemple utilise l'adresse IP distante par défaut du projet.

**3.5.1.25 WNetSCopyFileRemote (Macro locale)**

Cette macro permet de copier un fichier d'un poste PLC vers un autre poste PLC.

Le premier paramètre désigne l'adresse IP du poste PLC contenant le fichier source.

Le deuxième paramètre indique le nom du fichier source.

Le troisième paramètre indique l'adresse IP du poste PLC de destination.

Le quatrième paramètre désigne le nom du fichier de destination.

Exemple :

WNETSCOPYFILEREOTE(193.0.1.1,C:\OLD.TXT,193.0.1.2,C:\NEW.TXT)

**3.5.1.26 WNetSCreateDirectory (Macro locale)**

Cette macro permet de créer un répertoire sur un poste PLC.

Le premier paramètre est optionnel, il désigne l'adresse IP du poste PLC.

Le deuxième paramètre indique le nom du répertoire à créer.

Exemples :

WNETSCREATEDIRECTORY(193.0.1.1,C:\REP1)

WNETSCREATEDIRECTORY(C:\REP1)

Le deuxième exemple utilise l'adresse IP distante par défaut du projet.

**3.5.1.27 WNetSDeleteDirectory (Macro locale)**

Cette macro permet de supprimer un répertoire sur un poste PLC.

Le premier paramètre est optionnel, il désigne l'adresse IP du poste PLC.

Le deuxième paramètre indique le nom du répertoire à supprimer.

Exemples :

WNETSDLETEDIRECTORY(193.0.1.1,C:\REP1)

WNETSDLETEDIRECTORY(C:\REP1)

Le deuxième exemple utilise l'adresse IP distante par défaut du projet.

**3.5.1.28 WNetSReadAndMoveFile (Macro locale)**

Cette macro permet de copier un fichier d'un poste PLC vers le poste WIN qui appelle cette macro. A la suite de la copie, le fichier source est supprimé sur le poste PLC.

Le premier paramètre est optionnel, il désigne l'adresse IP du poste PLC contenant le fichier source.

Le deuxième paramètre indique le nom du fichier source sur le poste PLC.

Le troisième paramètre indique le nom du fichier de destination sur le poste WIN.

Exemples :

```
WNETSREADANDMOVEFILE(193.0.1.1,C:\FILE1.TXT,C:\REP1\FILEDOS1.TXT)
```

```
WNETSREADANDMOVEFILE(C:\FILE1.TXT,C:\REP1\FILEDOS1.TXT)
```

Le deuxième exemple utilise l'adresse IP distante par défaut du projet.

**3.5.1.29 WNetSWriteAndMoveFile (Macro locale)**

Cette macro permet de copier un fichier du poste WIN qui appelle cette macro vers un poste PLC. A la suite de la copie, le fichier source est supprimé sur le poste WIN.

Le premier paramètre est optionnel, il désigne l'adresse IP du poste PLC sur lequel on doit effectuer la copie.

Le deuxième paramètre indique le nom du fichier source sur le poste WIN.

Le troisième paramètre indique le nom du fichier de destination sur le poste PLC.

Exemples :

```
WNETSWRITEANDMOVEFILE(193.0.1.1,C:\FILE1.TXT,C:\REP1\FILEWIN1.TXT)
```

```
WNETSWRITEANDMOVEFILE(C:\FILE1.TXT,C:\REP1\FILEWIN1.TXT)
```

Le deuxième exemple utilise l'adresse IP distante par défaut du projet.

**3.5.1.30 WNetSReadFile (Macro locale)**

Cette macro permet de copier un fichier d'un poste PLC vers le poste WIN qui appelle cette macro.

Le premier paramètre est optionnel, il désigne l'adresse IP du poste PLC contenant le fichier source.

Le deuxième paramètre indique le nom du fichier source sur le poste PLC.

Le troisième paramètre indique le nom du fichier de destination sur le poste WIN.

Exemples :

```
WNETSREADFILE(193.0.1.1,C:\FILE1.TXT,C:\REP1\FILEDOS1.TXT)
```

```
WNETSREADFILE(C:\FILE1.TXT,C:\REP1\FILEDOS1.TXT)
```

Le deuxième exemple utilise l'adresse IP distante par défaut du projet.

**3.5.1.31 WNetSWriteFile (Macro locale)**

Cette macro permet de copier un fichier du poste WIN qui appelle cette macro vers un poste PLC.

Le premier paramètre est optionnel, il désigne l'adresse IP du poste PLC sur lequel on doit effectuer la copie.

Le deuxième paramètre indique le nom du fichier source sur le poste WIN.

Le troisième paramètre indique le nom du fichier de destination sur le poste DOS

Exemples :

```
WNETSWRITEFILE(193.0.1.1,C:\FILE1.TXT,C:\REP1\FILEWIN1.TXT)
```

```
WNETSWRITEFILE(C:\FILE1.TXT,C:\REP1\FILEWIN1.TXT)
```

Le deuxième exemple utilise l'adresse IP distante par défaut du projet.

**3.5.1.32 WNetSSetFileAttr (Macro locale)**

Cette macro permet de modifier les attributs d'un fichier sur un poste PLC.

Le premier paramètre est optionnel, il désigne l'adresse IP du poste PLC sur lequel on doit effectuer la copie.

Le deuxième paramètre indique le nom du fichier sur le poste PLC.

Le troisième paramètre l'attribut à appliquer au fichier du poste PLC. Ce paramètre est de type entier.

Exemples :

```
WNETSSETFILEATTR(193.0.1.1,C:\FILE1.TXT,32)
```

```
WNETSSETFILEATTR(C:\FILE1.TXT,32)
```

Le deuxième exemple utilise l'adresse IP distante par défaut du projet.

**3.5.2 Macros externes (exécutées sur le PLC)**

Les macros externes sont exécutées sur le poste noyau temps réel, ces macros sont les macros standard d'APIGRAF DOS.

Les macros DOS « exploitables » depuis la supervision Windows® sont décrites dans les chapitres suivants.

Les macros externes sont exécutées sur le noyau APIGRAF PLC dont l'adresse IP correspond soit à l'adresse IP de la page de supervision d'où est effectuée la macro (Si une adresse a été spécifiée pour la page) soit sur le noyau APIGRAF dont l'adresse IP correspond à l'adresse IP par défaut du projet.

Il est cependant possible de spécifier l'adresse IP du noyau APIGRAF PLC directement dans le texte de la macro locale.

Exemple :

SETSA1=1 (Set de la variable SA1 sur le noyau APIGRAF PLC de l'adresse IP par défaut de la page ou du projet).

!193.0.1.1!SETSA1=1 (Set de la variable SA1 sur le noyau APIGRAF PLC dont l'adresse IP est 193.0.1.1).

### 3.5.2.1 BIP (Macro PLC)

Provoque un bip sur le haut-parleur du PC.

### 3.5.2.2 DECR### (Macro PLC)

Décrémentation du registre R No « ### »

Exemple :

DECR10

### 3.5.2.3 INCR### (Macro PLC)

Incrémentation du registre R No « ### »

Exemple :

INCR10

### 3.5.2.4 INVSA### (Macro PLC)

Inversion de l'état du SA No « ### »

Exemple :

INVSA5

### 3.5.2.5 SET...= (Macro PLC)

Affectation d'une valeur à une variable automate.

Exemples :

SET R15=100

SET R15=R10+R11

SET L10=1

SET SA23=0

## 3.5.3 Macros utilisateur

### 3.5.3.1 Généralités

APIGRAF WIN offre la possibilité aux utilisateurs de développer leurs propres macros. Ces macros pourront alors être exécutés depuis la supervision.

Les macros utilisateurs sont développées dans des DLL « utilisateur » qui sont créées à l'aide de DELPHI®.

### 3.5.3.2 Développement d'une DLL Utilisateur

Afin de développer une DLL utilisateur, on doit disposer d'un compilateur DELPHI 32 bits (ou autre)

A l'aide de delphi, créer un projet de type DLL.

**La première unité mentionnée dans le USES de la DLL doit absolument être SHAREMEM**

Dans le menu « Outils -> Options du débogueur », choisir l'onglet « Exceptions du langage » et décocher la case « Arrêter sur exceptions Delphi » si la case correspondante est cochée.

L'architecture de la DLL doit respecter le format suivant :

**Architecture d'une DLL Utilisateur :**

**LIBRARY** LIBUSER ;

**USES** SHARMEM ;

{ \$R \*.RES }

**type**

```
TParamsList = class(TStringList)
    private
        vMacro : string;
    public
        EndMacro      : string;
        property Macro : string read vMacro;
    end;

Function GetMacroNumber : Integer ; Stdcall ;
begin
    ...
end ;

Function GetMacroName (AmacroNum : Integer ;
                        Var AMacroName : string ;
                        Var AmacroAddress : pointer)
                        : Boolean ; stdcall ;
begin
    ...
end ;

function GetItemInfo ( Anum : Integer;
                        var AType : Integer;
                        var AStrID : shortstring;
                        var AIPAddress : shortstring;
                        var AOPCServerName : shortstring;
                        var AAddress : pointer )
                        : boolean; stdcall; export;
begin
    ...
end ;

exports
    GetMacroNumber,
    GetMacroName,
    GetItemInfo;

End.
```

La Fonction **GetMacroNumber** doit renvoyer le nombre de macros.

La Fonction **GetMacroName** doit renvoyer en paramètres pour un numéro de macro donnée (AmacroNum allant de 1 au nombre de macros présentes dans la DLL ; le nombre de macros présentes dans la DLL est retourné par la fonction GetMacroNumber ) le nom de la macro dans AmacroName et l'adresse de procédure correspondant à la macro dans AmacroAddress sous forme de pointeur.

La fonction **GetItemInfo** doit renvoyer en paramètres les infos concernant les variables de la DLL à mettre à jour avec les variables automates

Les paramètres sont les suivants :

**AType** : spécifie le type de variable à relier à la variable automate

- 1 : type boolean
- 2 : type Longint
- 3 : type real
- 4 : type Extended
- 5 : type shortstring

**AStrID** : désignation de la variable automate à superviser

Exemple : 'R10'

**AIPAddress** : Adresse IP du poste PLC sue lequel on doit superviser la variable.

Exemple : '193.0.1.1'

**AAddress** : pointeur sur la variable à relier à la variable automate

Exemple : @long1



**Exemple de fonction GetItemInfo :**

Dans cet exemple on veut utiliser deux variables DELPHI (Long1 de type longint et Bool1 de type Booléen). On veut que la valeur de Long1 soit mise à jour avec la valeur de la variable automate R10 à l'adresse IP 193.0.1.1

De même, on désire que la valeur de Bool1 soit mise à jour avec la valeur de la variable automate SA10 à l'adresse IP 193.0.1.1

Une modification de la valeur de Long1 ou de la valeur de Bool1 entraînera une mise à jour de la valeur de R10 ou de SA10.

La fonction GetItemInfo doit retourner TRUE pour Anum=1 et Anum=2 et FALSE sinon, puisqu'il y a 2 variables à superviser (R10 et SA10).

**Attention : il faut bien respecter les types des variables**

```
function GetItemInfo ( Anum           : Integer;
                      var AType       : Integer;
                      var AStrID      : shortstring;
                      var AIPAddress  : shortstring;
                      var AOPCServerName : shortstring;
                      var AAddress    : pointer      )
: boolean; stdcall; export;

begin
Result := True ;
AOPCServerName := '';
Case Anum of
  1      : begin
            AIPAddress := '193.0.1.1' ;
            Atype      := 2 ;
            AstrID      := 'R10' ;
            Aaddress    := @Long1 ;
          end ;
  2      : begin
            AIPAddress := '193.0.1.1' ;
            Atype      := 1 ;
            AstrID      := 'SA10' ;
            Aaddress    := @Bool1 ;
          end ;
  else    begin
            Result := False;
          end ;
End ;
end ;
```

**Format d'une macro Utilisateur**

```
Procedure MacroUser1 ( Params : TparamsList ) ;
Begin
...
end ;
```

L'unique paramètre des macro doit être de type TparamsList ( C'est une liste de chaînes)

Le nombre de paramètres est ici donné par Params.Count

L'accès aux paramètres ( qui sont de type string ) est réalisé avec :

Params[0],...,Params[Params.Count – 1]

**Exemple d'appel d'une macro utilisateur depuis une page de supervision.**

MacroUser1(Chaine1,12)

Dans cet exemple on passe 2 paramètres sous forme de string, la macro utilisateur doit se charger de la conversion des types.

### 3.5.3.3 Exemple de DLL Utilisateur

Le source de la DLL ainsi que la DLL sont disponibles dans le répertoire DLLUSER du répertoire Exemples.

L'exemple suivant illustre une DLL utilisateur ayant 3 macros utilisateur : puissance, messageinfo et somme.

```
library LibUser;
uses sharemem,
    classes,
    SysUtils,
    dialogs,
    windows;
{$R *.RES}
type
    TparamsList = class(TStringList)
        private
            vMacro      : string;
        public
            EndMacro     : string;
            property Macro : string read vMacro;
        end;
var
    X1  : Longint;
    X2  : Longint;
    X3  : Longint;
    X4  : Longint;
    X5  : Longint;
    X6  : Longint;
    X7  : Longint;
const
    NbMacro = 3;
function GetItemInfo ( Anum          : Integer;
                       var AType      : Integer;
                       var AStrID     : shortstring;
                       var AIPAddress : shortstring;
                       var AAddress   : pointer      )
                       : boolean; stdcall; export;

begin
    Result := True;
    AIPAddress := '193.0.1.1';
    Case Anum of
        1 : begin
            AAddress := @X1;
            AType     := 2;
            AStrID    := 'R12';
        end;
        2 : begin
            AAddress := @X2;
            AType     := 2;
            AStrID    := 'R13';
        end;
        3 : begin
            AAddress := @X3;
            AType     := 2;
            AStrID    := 'R14';
        end;
        4 : begin
            AAddress := @X4;
```

```

        AType      := 2;
        AStrID     := 'R15';
    end;
5 : begin
    AAddress      := @X5;
    AType         := 2;
    AStrID        := 'R16';
    end;
6 : begin
    AAddress      := @X6;
    AType         := 2;
    AStrID        := 'R20';
    end;
7 : begin
    AAddress      := @X7;
    AType         := 2;
    AStrID        := 'R21';
    end;
else begin
    Result := False;
    Exit;
end;
end; {Case}
end;
Procedure Somme(Params : TParamsList);
begin
    X1 := X2 + X3 + X4 + X5;
end;
Procedure MessageInfo(Params : TParamsList);
begin
    if Params.Count <= 0
    then Exit;
    MessageDlg(Params[0], mtinformation, [MbOK], 0);
end;
Procedure Puissance(Params : TParamsList);
var
    Pui      : Integer;
    Res      : Integer;
    i        : Integer;
begin
    if Params.Count <= 0
    then Exit;
    try
        pui := StrToInt(Params[0]);
    except
        Exit;
    end;
    if Pui <= 1
    then Exit;
    Res := 1;
    for i:= 1 to pui
    do begin
        Res := Res * X7;
    end;
    X6 := Res;
end;
{Fonction qui retourne le nombre de macros présentes}
Function GetMacroNumber : Integer; stdcall; export;
begin
    Result := NbMacro;
end;
Function GetMacroName (AMacroNum : Integer;
                        var AMacroName : String;
                        var AMacroAddress : Pointer) :

```

```

Boolean; stdcall;

begin
Result := False;
if (AMacroNum < 1)
or (AMacroNum > NbMacro)
then Exit;
Case AMacroNum of
1 : begin
AMacroName      := 'Somme';
AMacroAddress   := @Somme;
end;
2 : begin
AMacroName      := 'MessageInfo';
AMacroAddress   := @MessageInfo;
end;
3 : begin
AMacroName      := 'Puissance';
AMacroAddress   := @Puissance;
end;
end;{Case}
Result := True;
end;
{ La DLL doit obligatoirement exporter ces deux fonctions }
{ afin d'être utilisé dans APIGRAF WIN }
exports
    GetMacroNumber,
    GetMacroName,
    GetItemInfo;
end.

```

### 3.5.3.4 Utilisation d'une DLL Utilisateur dans un projet

Une fois que la dll a été compilé, il faut spécifier au projet qu'il doit utiliser la DLL.

Dans le menu Macros->Macros Utilisateur-> Charger DLL Utilisateur, il faut spécifier la DLL à charger.

Si la DLL a été correctement chargée et que celle-ci est correctement formatée, les macros disponibles dans cette DLL apparaissent désormais dans la liste des macros locales (VOIR Menu Macros-> Macros locales ) et sont accessibles dans le projet. Le commentaire associé à ces macros dans la liste fait apparaître le nom de fichier DLL et le numéro de macro du fichier.

Une fois chargée, la dll sera rechargée automatiquement à chaque ouverture du projet ou à chaque exécution d'une page de supervision du projet. Le nom des DLL chargées est en effet consigné dans le fichier \*.INI du projet.

### 3.5.3.5 Suppression d'une DLL utilisateur d'un projet

Pour supprimer une DLL utilisateur d'un projet, il faut aller dans le menu macros -> Macros Utilisateur -> Décharger une DLL Utilisateur.

La liste des DLL Utilisateur apparaît alors.

Choisir dans la liste la dll à supprimer et cliquer sur le bouton « Décharger ».

La dll ne sera alors plus chargée dans la projet.

### 3.5.3.6 Consultation des DLL utilisateur

Le menu Macros -> Macros Utilisateur -> Liste des DLL Utilisateur permet de visualiser la liste de toutes les DLL utilisateur chargée dans le projet. Pour chaque DLL, le nombre de macros est indiqué.

Pour visualiser les macros disponibles dans chaque DLL, il faut se reporter au menu Macros-> Macros locales qui permet de visualiser toutes les macros locales disponibles. Le commentaire associé aux macros spécifie si celles-ci sont chargées dans des DLL Utilisateur.

## 3.6 Gestion des traductions

### 3.6.1 Ecriture des fichiers de langage

Il est possible d'avoir des pages de supervision en plusieurs langues. Pour cela il faut avant tout créer des fichiers de langage. Ces fichiers de langage contiendront les différentes traductions.

Les noms des fichiers de langage sont du type LANGAGE.1, LANGAGE.2, etc...Chaque fichier langage sera affecté à un langage. Les fichiers devront être présents dans le répertoire du projet.

#### 3.6.1.1 Création d'un nouveau fichier de langage

Choisir dans le menu « Outils », « Traductions utilisateur » puis « Nouveau fichier de langage ».

Si aucun fichier langage n'était présent, le nom du fichier créé est « LANGAGE.1 »

Les traductions devront être entrées de la façon suivante dans le fichier :

@LG1:

Exemple de traduction

@LG2 :

Texte N°2

@LG3 :

Texte N°3

.... Etc

Dans cet exemple, si on désire la correspondance en anglais, il faudra créer un nouveau fichier langage

« LANGAGE.2 » :

@LG1:

Translation example

@LG2:

Text #2

@LG3:

Text #3

#### 3.6.1.2 Utilisation des fichiers de langage dans les pages de supervision

Si on veut par exemple que le texte d'un bouton soit traduit, il faut entrer dans la paramètre texte du bouton, un code du Type @LG2.

Vous pouvez aussi associer un commentaire au code (exemple : @LG2:Bouton Page 2). Dans ce cas le texte associé au bouton sera répertorié dans le fichier langage au niveau suivant :

@LG2 :

Texte N°2

#### 3.6.1.3 Changement de langue

Une fois les fichiers langages complets et les pages de supervision codées avec le symbole @LG, la macro CHNUSERLANGUAGE en lui passant en paramètre le numéro du fichier de langage à utiliser.

Exemple : CHNUSERLANGUAGE(2)

Cette macro peut être placée par exemple dans un bouton.

#### 3.6.1.4 Exemple de traduction

Un projet d'exemple est disponible dans ..\APIGRAFIP\EXEMPLES\TRADUCTION.

Ce projet contient deux fichiers de langage LANGAGE.1 et LANGAGE.2.

## 3.7 Gestion des utilisateurs

### 3.7.1 Généralités

Il est possible de restreindre les possibilités d'action dans le runtime en fonction de l'utilisateur. Pour cela, le mode gestion des utilisateurs doit être activé dans le menu Options/Gestion des utilisateurs.

Dans ce mode, un nom d'utilisateur et un mot de passe peuvent être demandés avant de donner accès à une page de supervision, en fonction du groupe d'appartenance de l'utilisateur.

Dans ce cas, il faut définir dans les propriétés de la page de supervision un niveau minimum pour l'accès à celle-ci.

Les groupes d'utilisateurs sont les suivants :

Groupe	Droits
Admin	Droits complets.
Maintenance	Droits complets, sauf pour la gestion des utilisateurs et la modification des options de fonctionnement.
Supervisors	Accès limités à la création d'opérateurs, à la modification des variables standards de l'automate (R, SA, etc.) et à la visualisation de l'ensemble des informations.
Operators	Accès limités à la visualisation de certaines informations (variables standards et entrées/sorties).

Par défaut, la gestion des utilisateurs n'est pas activée, et l'accès est équivalent au groupe Admin. En cas d'activation de la gestion des utilisateurs, si cette activation est faite pour la première fois, seul l'utilisateur "Admin", du groupe Admin, est prédéfini. Le mot de passe initial est "Admin".

En cas de perte des mots de passe empêchant l'accès au runtime, la suppression du fichier Users.dat situé dans \Bin (répertoires d'installation d'Optima View) permet de reprendre la configuration initiale décrite ci-dessus.

## 3.8 Exécution des pages de supervision

### 3.8.1 Fichiers créés dans un projet de supervision

Lors de la création d'un nouveau projet, un fichier « NomProjet.INI » est créé dans le projet. Ce fichier contient certaines informations concernant le projet.

Une copie de ce fichier nommée « NomProjet.INB » peut également être présente dans le répertoire du projet. Si aucun de ces fichiers n'est présent dans le répertoire, les informations contenues dans ceux-ci seront perdues et un message d'erreur avertit l'utilisateur. Un nouveau fichier \*.INI sera alors créé à la fermeture de la page.

### 3.8.2 Exécution d'une page depuis l'atelier

Pour exécuter une page de supervision depuis l'éditeur, il faut tout d'abord ouvrir la page que l'on souhaite exécuter dans le projet.

Ensuite, il faut cliquer sur l'icône « Exécution » de la barre d'outils flottante « FONCTIONS ». Cette fonction est symbolisée par un engrenage.

### 3.8.3 Exécution d'une page avec son icône

Une page de supervision est sauvegardée dans un fichier \*.PSWG. Ce type d'extension de fichier est lié à l'exécutable des pages de supervision SUPERGW.EXE.

Pour exécuter directement une page, il suffit donc de double-cliquer sur son icône.

## 3.9 Clavier virtuel

### 3.9.1 Utilisation du clavier virtuel

Le clavier virtuel est utilisé sur les écrans tactiles qui ne sont pas munis de clavier. C'est un clavier qui s'ouvre à l'écran à côté de la fenêtre de saisie et qui permet à l'utilisateur de saisir des caractères alphanumériques en cliquant sur les boutons de ce clavier.

Cette option permet d'ouvrir le clavier virtuel de manière automatique dès que l'utilisateur clique sur une fenêtre de saisie dans une page de supervision.

*Exemple :* Clavier virtuel alphanumérique



Pour mettre en place le clavier virtuel voir le menu :

[Menu Options générales](#) 

## 3.10 Utilisation de liens DDE

### 3.10.1 Principes généraux

Le lancement d'une page de supervision hors de l'atelier de supervision ou le lancement de l'application « Atelier de supervision » provoque le lancement d'un serveur de données (APIDATAS.EXE). Ce serveur de données peut être utilisé par le serveur DDE APIGRAF IP : APIDDE.EXE. Toutes les données automatées peuvent donc être exploitées dans des applications Windows® telles que Excel, Word...

Le serveur DDE est caractérisé par :

- L'application contenant le serveur : APIDDE
- Le nom du serveur ou (Topic): PLCDATA
- Les rubriques (ou Item) du serveur : Celles-ci sont créées au moment de la demande de lien sur une variable automate particulière sur un noyau APIGRAF PLC d'adresse IP donnée

Exemple : !193.0.1.1 !R12

Cet item permet de récupérer la valeur du registre R12 sur le noyau PLC d'adresse IP 193.0.1.1.

### 3.10.2 Exemple : Création d'un lien DDE avec MICROSOFT® Excel

Dans un nouveau document Excel, tapez dans une case la formule suivante :

=APIDDE|PLCDATA !'193.0.1.1 !R12'

Puis validez avec « ENTREE »

Si le serveur APIDDE n'est pas lancé, EXCEL® proposera de le lancer.

La case affiche alors la valeur de R12 . (Si le serveur APIDDE.EXE est bien lancé)

Un exemple illustrant les liens DDE avec Excel® se trouve dans le sous-répertoire \EXEMPLES\Lien DDE Excel. Dans le répertoire de cet exemple se trouve un fichier Excel contenant un graphique pour illustrer les liens DDE.

### 3.10.3 Autorisation de modification de variables automatées via un lien DDE

Pour autoriser la modification d'une variable Automate via un lien DDE, il faut le spécifier dans l'écriture de la rubrique :

Exemple :

Rubrique n'autorisant pas la modification : !193.0.1.1 !R12

Rubrique autorisant le modification : M, !193.0.1.1 !R12

Dans l'exemple de la section précédente, cela se traduirait par la formule :

=APIDDE|PLCDATA!'M, !193.0.1.1 !R12'

## 3.11 Accès aux variables PLC depuis un programme écrit en DELPHI

### 3.11.1 Principes généraux

Lors de l'installation d'APIGRAF IP, deux fichiers concernant un paquet d'exécution DELPHI 5 sont installés.

Le fichier USERAPIPLC.BPL est copié dans le répertoire C:\WINDOWS\SYSTEM

Le fichier USERAPIPLC.DCP est copié dans le répertoire C:\PROGRAM FILES\OPTIMALOG\APIGRAFIP\BIN\Paquet-Delphi5

Un paquet d'exécution DELPHI permet d'accéder à des fonctions en déclarant simplement l'unité principale du paquet dans la clause USES du programme. Dans le cas présent, il suffit de faire : USES UClientAPI ;

Le paquet d'exécution étant compilé, il est utilisable uniquement avec DELPHI5.

### 3.11.2 Création d'un programme DELPHI5 utilisant le paquet USERAPIPLC

Pour créer un programme DELPHI5 utilisant le paquet USERAPIPLC.DCP, il faut suivre les étapes suivantes :

1. Créer une nouvelle application dans DELPHI5 et copier le fichier USERAPIPLC.DCP dans le répertoire de votre projet DELPHI.

2. Dans le menu de DELPHI : Outils->Options du débogueur, choisir l'onglet « Exceptions du langage ». Dans cet onglet, décocher l'option « Arrêter sur exceptions DELPHI » si celle-ci est cochée.

3. Dans le menu de DELPHI : Projet->Options, choisir l'onglet « Paquets ». Dans cet onglet, cocher l'option « construire avec les paquets d'exécution ». Dans le groupe « Paquets d'exécution », cliquer sur le bouton Ajouter puis sur parcourir pour choisir le paquet USERAPIPLC.DCP qui a été copié dans le répertoire de votre projet DELPHI.

4. Dans la clause USES de votre page principale, insérer le nom de l'unité UClientAPI.

5. A ce stade, vous pouvez utiliser les fonctions du paquet décrites section suivante.

### 3.11.3 Fonctions accessibles depuis le paquet USERAPIPLC

#### 3.11.3.1 Fonction LinkToAPI

**Function LinkToAPI** (AAddress : pointer ; AIDStr : shortstring ; AIPAddress : shortstring ; AdataType : Integer) : Boolean ;

Cette fonction permet de relier une variable DELPHI à une variable automate.

Chaque fois que la variable automate change d'état, la variable DELPHI est mise à jour avec la valeur de la variable automate. Inversement, une modification de la valeur de la variable DELPHI entraînera une mise à jour de variable automate avec la valeur de la variable DELPHI.

#### **Paramètres :**

Aaddress : pointeur sur la variable DELPHI à rafraîchir avec une variable PLC

AIDStr : Chaîne précisant la variable PLC à relier

AIPAddress : Adresse IP du poste PLC

AdataType : type de la variable DELPHI. Les choix possibles sont :

- |   |               |
|---|---------------|
| 1 | : Boolean     |
| 2 | : Longint     |
| 3 | : Real        |
| 4 | : Extended    |
| 5 | : ShortString |

#### **Résultat de la fonction :**

La fonction renvoie TRUE si la demande a bien été prise en compte et FALSE sinon.

Attention : cela ne veut pas dire que si la fonction renvoie TRUE, alors la variable DELPHI est bien rafraîchie avec la valeur de la variable PLC. Il faut utiliser la fonction IsLinkedToAPI ou AllLinksOK pour le savoir.

#### **Exemple :**

LinkToAPI (@X1,'R466','193.0.1.1',2) ;



(Avec X1 : Longint)

Remarque : Si la variable est de type local, le paramètre adresse IP peut être vide :

LinkToAPI(@X1,'WR1','');

Voir [Variables locales](#) <sup>321</sup>

### 3.11.3.2 Fonction IsLinkedToAPI

**Function IsLinkedToAPI** (Aaddress : pointer) : Boolean ;

Cette fonction permet de savoir si une variable DELPHI qui a été précédemment reliée à une variable automate à l'aide de la fonction LinkToAPI est bien rafraîchie.

**Paramètres** :

Aaddress : pointeur sur la variable DELPHI

**Résultat de la fonction** :

La fonction renvoie TRUE si la variable DELPHI est bien rafraîchie et FALSE sinon.

**Exemple** :

Bool1 := IsLinkedToAPI (@X1) ;

### 3.11.3.3 Fonction AllLinksOK

**Function AllLinksOK** : Boolean ;

Cette fonction permet de savoir si toutes variable DELPHI qui ont été précédemment reliées à des variables automates à l'aide de la fonction LinkToAPI sont bien rafraîchies.

**Paramètres** :

Pas de paramètres.

**Résultat de la fonction** :

La fonction renvoie TRUE si toutes les variables DELPHI sont bien rafraîchies et FALSE sinon.

**Exemple** :

Bool1 := AllLinksOK;

### 3.11.3.4 Fonction UnLinkToAPI

**Function UnLinkToAPI** (Aaddress : pointer) : Boolean ;

Cette fonction permet de défaire le lien entre une variable DELPHI et une variable automate.

**Paramètres** :

Aaddress : pointeur sur la variable DELPHI

**Résultat de la fonction** :

La fonction renvoie TRUE si le lien a été défait et FALSE sinon.

Si aucun lien n'existe pour le pointeur sur la variable, la fonction renvoie FALSE.

**Exemple** :

Bool1 := UnLinkToAPI(@X1) ;

### 3.11.3.5 Procédure ExecMacro

**Procedure ExecMacro** (AmacroText : string; AlPaddress : shortstring);

Cette fonction permet d'exécuter une macro distante sur un poste PLC ou une macro locale Apigraf WIN.

La liste des macros locales utilisables depuis cette procédure est :

WNETCHANGEFILENAME  
WNETSCOPYFILEREMOTE  
WNETSCREATEDIRECTORY  
WNETSDELETEDIRECTORY  
WNETSREADANDMOVEFILE  
WNETSWRITEANDMOVEFILE  
WNETSREADFILE  
WNETSWRITEFILE

WNETSSETFILEATTR  
WFCOPY  
WFDELETE  
WFCREATEDIRECTORY  
WFDELETEDIRECTORY  
WFRENAME  
WFSETFILEATTR

*La macro SUPERG n'est pas disponible depuis cette procédure* puisque l'application utilisateur est exécutée en dehors d'APIGRAF WIN. Cependant l'utilisateur peut écrire en DELPHI :

```
WINEXEC( "C:\Program Files\OptimaLog\ApigrafIP\Bin\Exe\SUPERGW.EXE" "C:\Program Files\OptimaLog\ApigrafIP\Projets\Test1\Page1.PSGW",0)
```

Cette procédure exécutera la page de supervision Apigraf WIN page1.psgw.

Syntaxe : WINEXEC ("C:\...\SUPERGW.EXE" "C:\...\Page1.PSGW",0)

Les doubles guillemet doivent être présent quand la chaîne contient des espaces.

Il est aussi possible dans ce cas de passer des paramètres à la page de supervision :

```
WINEXEC( "C:\Program Files\OptimaLog\ApigrafIP\Bin\Exe\SUPERGW.EXE" "C:\Program Files\OptimaLog\ApigrafIP\Projets\Test1\Page1.PSGW" 193.0.1.1,0)
```

Dans cet exemple tous les paramètres {1} mentionnés dans la page de supervision seront remplacés par 193.0.1.1. (Voir [Appel de page avec passage de paramètres](#)<sup>297</sup>)

#### **Paramètres :**

AmacroText : Texte de la macro à exécuter (Attention : dans le cas des macros distantes, il ne faut pas préciser l'adresse IP dans ce paramètre)

AIPAddress : Adresse IP du poste PLC sur lequel effectuer la macro dans le cas d'une macro distante. Si la macro est locale, il faut mettre une chaîne vide dans ce paramètre.

#### **Exemples :**

##### Exécution d'un macro distante:

```
ExecMacro ('SETR100=1200', '193.0.1.1') ;
```

##### Exécution d'un macro locale:

```
ExecMacro("WNETSWRITEFILE(193.0.1.1,C:\ESSAI.TXT,C:\ESSAI2.TXT),")
```

### **3.11.3.6 Fonction PLCState**

**Function PLCState** ( AIPAddress : Shortstring ; ATimeOut : Integer ;  
var ProjectName : Shortstring ; var SocketState : Integer) : Integer ;

Cette fonction permet de connaître l'état d'un poste PLC, de savoir si un projet est démarré, si oui, de quel projet il s'agit, si Apigraf est démarré ou bien si le poste est en ligne de commande.

#### **Paramètres :**

AIPAddress : adresse IP du poste dont on veut connaître l'état (shortstring)

AtimeOut : Spécifie la valeur de time out en millisecondes dans le cas ou le poste ne répond pas.  
(Valeur conseillée : 100)

var ProjectName : si un projet est démarré sur le poste PLC, cette chaîne de type shortstring contiendra le nom du projet en cours (exemple : C:\APIGRAF\EXEMPLES\CHENILL2\CHENILL2.EXE)

var SocketState : cette variable contient l'état du socket et peut être utile si la fonction en time out ne renvoie pas l'état du poste PLC. Les valeurs possibles sont :

0 : Closed  
1 : Listening  
2 : SynSent  
3 : SynReceived  
4 : Established  
5 : FinWait1

6 : FinWait2  
 7 : CloseWait  
 8 : Closing  
 9 : LastAck  
 10 : TimeWait

Dans le cas où le poste ne peut être joint sur le réseau ethernet, la valeur de l'état du socket sera 2 (SynSent).

#### **Résultat de la fonction :**

La fonction renvoie un entier de type Integer.

Les valeurs possibles sont :

0 : Etat du poste PLC Inconnu  
 3 : Un Projet est démarré  
 6 : L'atelier Apigraf PLC est démarré sur le poste mais aucun projet n'est lancé  
 7 : Le poste PLC est en ligne de commande avec INITAPI

#### **Exemple :**

```
Var Pname : shortstring ;
    SockState : Integer ;
EtatPLC := PLCState('193.0.1.1',100,Pname,SockState) ;
```

## **3.12 Variables locales OPTIMA VIEW**

### **3.12.1 Principes généraux**

OPTIMA VIEW possède des variables locales globales. Ces variables sont enregistrées au niveau du serveur de données SERVAPIOPC.EXE

Ces variables sont donc globales à tous les projets utilisant le serveur de données.

L'état de ces variables est enregistré dans un fichier texte : SERVAPIOPC.VAR et ces variables sont rechargées depuis ce fichier à chaque démarrage du serveur de données.

Les variables locales sont accessibles depuis les pages de supervision Optima View ainsi que depuis les programmes utilisateur.

### **3.12.2 Description des variables locales**

Le nombre maximal de chaque type de variable peut être défini au niveau du serveur de données SERVAPIOPC.EXE. Il suffit d'ouvrir le serveur en double cliquant sur son icône dans le coin gauche de l'écran et de choisir le menu "Options->Constantes systèmes"

#### **Registres WR**

Les registres WR sont de type entier sur 4 Octets. Ils sont accessibles depuis la supervision avec les expressions WR0,WR1,...WR100,

...

#### **Registres réels WRR**

Les registres WRR sont de type réel extended sur 10 Octets. Ils sont accessibles depuis la supervision avec les expressions WRR0,WRR1,...WRR100,...

#### **Flags logiques WL**

Les flags WL sont de type booléen. Ils sont accessibles depuis la supervision avec les expressions WL0,WL1,...WL100,...

#### **Interrupteurs logiques WSA**

Les interrupteurs logiques WSA sont de type booléen. Ils sont accessibles depuis la supervision avec les expressions WSA0,WSA1,...WSA100,...

#### **Chaînes de caractères WC**

Les chaînes WC sont de type string et sont accessibles depuis la supervision avec les expressions WC1,WC2,...WC100,...

### **3.12.3 Définition du nombre de variables locales**

Le nombre maximal de chaque type de variable peut être défini au niveau du serveur de données SERVAPIOPC.EXE.

Il suffit d'ouvrir le serveur de données SERVAPIOPC.EXE en double cliquant sur son icône dans la coin gauche de l'écran et de choisir le menu "Options->Constantes systèmes" et de redéfinir NbMaxWR, NbMaxWRR, NbMaxWSA, NbMaxWL, NbMaxWC.

### 3.12.4 Visualisation des variables locales

Pour visualiser les variables locales il faut ouvrir le serveur de données SERVAPIOPC.EXE en double cliquant sur son icône dans le coin gauche de l'écran et d'ouvrir le menu "Visualisation->Variables locales" puis de choisir le type de variable à visualiser dans le sous-menu : WR, WRR, WSA, WL ou WC.

## 3.13 Serveur OPC : ServApiOPC

### 3.13.1 Généralités

Le serveur OPC d'OPTIMA VIEW est contenu dans l'application ServAPIOPC.EXE. Cette application se lance automatiquement dès qu'on tente d'accéder aux données du serveur. De la même manière, le serveur se ferme automatiquement lorsqu'il n'est plus utilisé. La fenêtre principale de l'application servapiOPC peut être ouverte lorsque celle-ci est lancée dans le gestionnaire de services APIGRAF IP. Il faut faire un double click sur l'icône IP dans le coin inférieur droit de l'écran.

Le serveur OPC répond aux exigences OPC : DATA ACCESS V2.0.

Un client OPC externe peut accéder aux données du serveur OPC.

Toutes les variables automates d'apigraf PLC, toutes les variables utilisateurs enregistrées sur la partie PLC à l'aide de « ENREGISTREMOT » ainsi que toutes les variables locales sont accessibles à l'aide du serveur OPC.

Ces variables sont accessibles par défaut en lecture seule (Read only) pour des raisons de sécurité. Cependant l'utilisateur peut définir la liste des variables qu'il souhaite consulter en lecture/écriture (Read/Write).

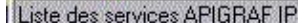
### 3.13.2 Application ServApiOPC

La fenêtre principale de l'application servapiOPC peut être ouverte lorsque celle-ci est lancée dans le gestionnaire de services APIGRAF IP. L'ouverture du serveur OPC se fait depuis le menu d'Optima PLC. Si l'atelier de supervision fonctionne indépendamment d'Optima PLC, suivre les instructions suivantes pour ouvrir le serveur OPC :

Faire un double clic sur l'icône IP dans la barre des tâches à coté de l'horloge :



La fenêtre du gestionnaire de service s'ouvre :



La fenêtre principale du serveur OPC s'ouvre alors à l'écran :





### 3.13.3 Accès aux données du serveur OPC

#### 3.13.3.1 Nom du serveur OPC

Le nom du serveur OPC pour l'accès aux données est : APIGRAFIP.PLC

#### 3.13.3.2 Définition d'un groupe

Seuls des groupes privés peuvent être créés sur le serveur OPC.

#### 3.13.3.3 Définition d'un item

Les items désignent en fait les données auxquelles on veut accéder via le serveur OPC. Les noms valides pour un item sont composés de l'adresse IP du poste PLC sur lequel on veut superviser la variable ainsi que du nom de la variable à superviser :

Exemples :

Variable R100 à l'adresse 193.0.1.1

Nom de l'item : !193.0.1.1 !R100

Expression R300+R200 à l'adresse 193.0.1.1

Nom de l'item : !193.0.1.1 !R300+R200

Variable locale WSA10

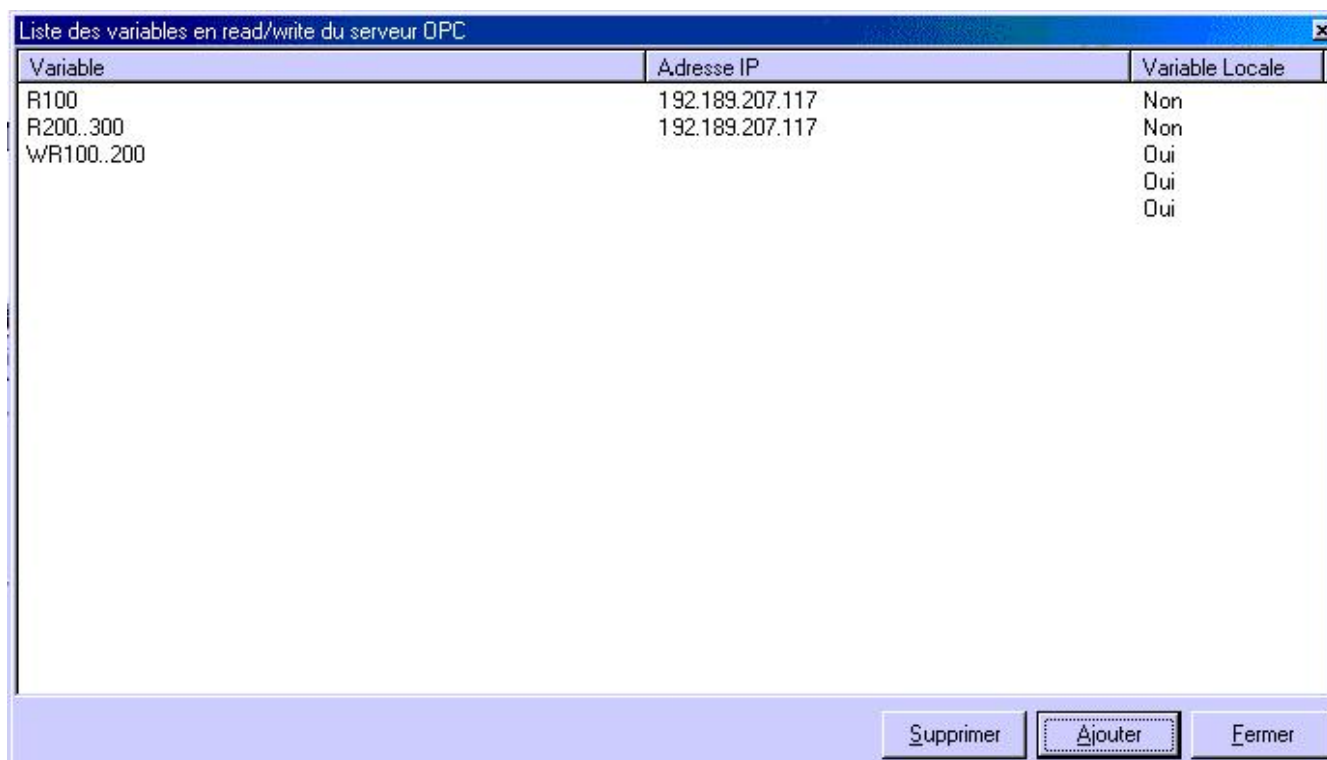
Nom de l'item : WSA10

#### 3.13.3.4 Définition des données en lecture/écriture

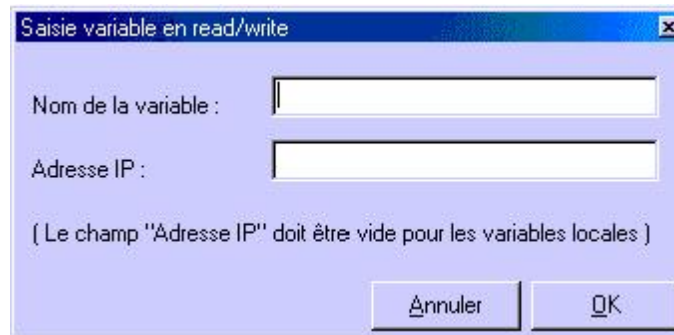
Par défaut, toutes les variables accessibles depuis le serveur OPC sont en lecture seule pour des raisons de sécurité. Cependant, l'utilisateur peut définir une liste de variables en lecture/écriture.

Pour cela, il faut ouvrir la fenêtre principale de SERVAPIOPC. Ensuite, il faut choisir le menu «visualisation » puis «variables read/write pour clients OPC »

La fenêtre suivante s'ouvre alors :



Cette fenêtre montre la liste des variables définies en read/write. Pour ajouter des variables en read/write, cliquer sur le bouton "Ajouter". La fenêtre suivante s'ouvre :



Les noms de variables peuvent être des "plages de variables":

Exemple : R100..200

Dans le cas des variables locales, il faut laisser vide le champ "Adresse IP"

## 3.14 Utilisation de l'atelier en mode indépendant

L'atelier de supervision peut être dans certains cas installé et démarré sans OPTIMA PLC. Dans ce cas, le logiciel possède certaines fonctions supplémentaires comme par exemple la sélection du projet de travail.

### 3.14.1 Installation et démarrage de l'application de Supervision

#### 3.14.1.1 Installation de l'application

L'installation complète s'effectue sur deux postes (au moins).

Le poste 1 est appelé Poste APIGRAF PLC.

Le poste 2 est appelé Poste de supervision WINDOWS.

#### Installation du noyau APIGRAF PLC sur le poste 1

Démarrer le poste.

Le poste APIGRAF doit impérativement être muni d'une carte Ethernet. Le lancement du « packetdriver » (driver associé à la carte réseau) doit être effectué automatiquement au démarrage du poste.

Insérer la première disquette d'installation du noyau APIGRAF PLC dans le lecteur de disquette.

Taper « A :INSTALL »

Suivre les instructions de l'utilitaire d'installation.

Redémarrer le PC .

Taper « APIGRAF » pour lancer le noyau APIGRAF PLC.

Lors du premier lancement, la programme vous demande de configurer l'adresse IP du poste. Choisissez alors l'adresse IP du noyau APIGRAF PLC.

Attention : Les adresses IP du poste du Noyau APIGRAF PLC et du poste de SUPERVISION doivent être compatibles avec les masques de sous-réseau définis sur les deux postes.

Exemple : Si le masque de sous réseau défini sur les 2 postes est 255.255.255.0, les adresses compatibles sont les adresses dont les 3 premiers nombres sont égaux : 193.0.1.1 et 193.0.1.32 par exemple.

La supervision graphique à distance du poste APIGRAF PLC n'est possible que si un projet APIGRAF est en cours d'exécution sur le poste. Par contre, la prise de contrôle du poste APIGRAF PLC est possible à tout moment à l'aide de l'utilitaire PCMASTER (Icône « APIGRAF PLC Distant » sur le bureau). Cet utilitaire vous permet de développer et d'exécuter des projets sur le poste APIGRAF PLC depuis le poste de supervision.

#### Installation des outils de supervision sous Windows

Le poste sous windows® (95-98 ou 2000,NT) doit impérativement être muni d'une carte Ethernet convenablement configurée dans le système.

S'assurer que la configuration réseau du poste supporte bien le protocole TCP/IP et qu'une adresse IP a été définie manuellement dans le système :

Pour cela, se placer dans « MENU DEMARRER >PARAMETRES>PANNEAU DE CONFIGURATION>RESEAU »

Vérifier la présence d'un protocole TCP/IP attaché à la carte Ethernet , cela se traduit généralement par la présence d'une ligne de type : « TCP/IP -> « Nom de la carte Ethernet » ». Si tel est le cas, passer directement au point 6.

Si ce n'est pas le cas, choisir « AJOUTER », choisir ensuite « PROTOCOLE » , puis « MICROSOFT » et enfin « TCP/IP ».

Valider avec « OK ».

Dans la liste des composants réseau de la fenêtre réseau, choisir à l'aide de la souris la ligne correspondant au protocole « TCP/IP-> Carte Ethernet ». La ligne apparaît alors en surbrillance.

Cliquer alors sur le bouton « PROPRIETES ».

Dans l'onglet « ADRESSE IP », cocher « Spécifier une adresse IP ». Choisir alors l'adresse IP du poste et le masque de sous-réseau.

Valider les fenêtres avec « OK ». Si le système demande de redémarrer, redémarrer le poste avant de poursuivre l'installation.

Insérer la première disquette d'installation des outils de supervision APIGRAF sous windows®. (Insérer le CD si le support est un CD)

Dans une fenêtre « explorateur windows® », sélectionner le lecteur de disquette et exécuter « SETUP.EXE ».

Suivre les instructions de l'utilitaire d'installation.

Note : L'installation du noyau APIGRAF PLC peut être effectuée sur plusieurs postes, sachant que le poste de supervision WINDOWS® peut superviser plusieurs noyau APIGRAF PLC.

### 3.14.1.2 Démarrage de l'application

Double-cliquer sur le raccourci OPTIMA VIEW sur le bureau du poste de supervision pour lancer l'atelier de supervision.

Une fenêtre de sélection de projet permet de choisir un projet existant ou d'en créer un nouveau.

Lors du premier lancement, se placer dans le répertoire « PROJETS » et choisir «créer un nouveau projet ».

#### Utilisateurs avancés :

Une fois que des pages de supervision ont été créées, il est possible d'utiliser une ligne de commande (dans un raccourci ou depuis une autre application) pour exécuter une page de supervision, la ligne de commande aura alors la forme suivante :

```
"C:\Program Files\OptimaLog\Optima View\Bin\Exe\SUPERGW.EXE" "C:\Program Files\Optimalog\Optima View\Projets\TEST\Page1.PSGW"
```

Il est également possible de passer des paramètres à la page de supervision à ouvrir :

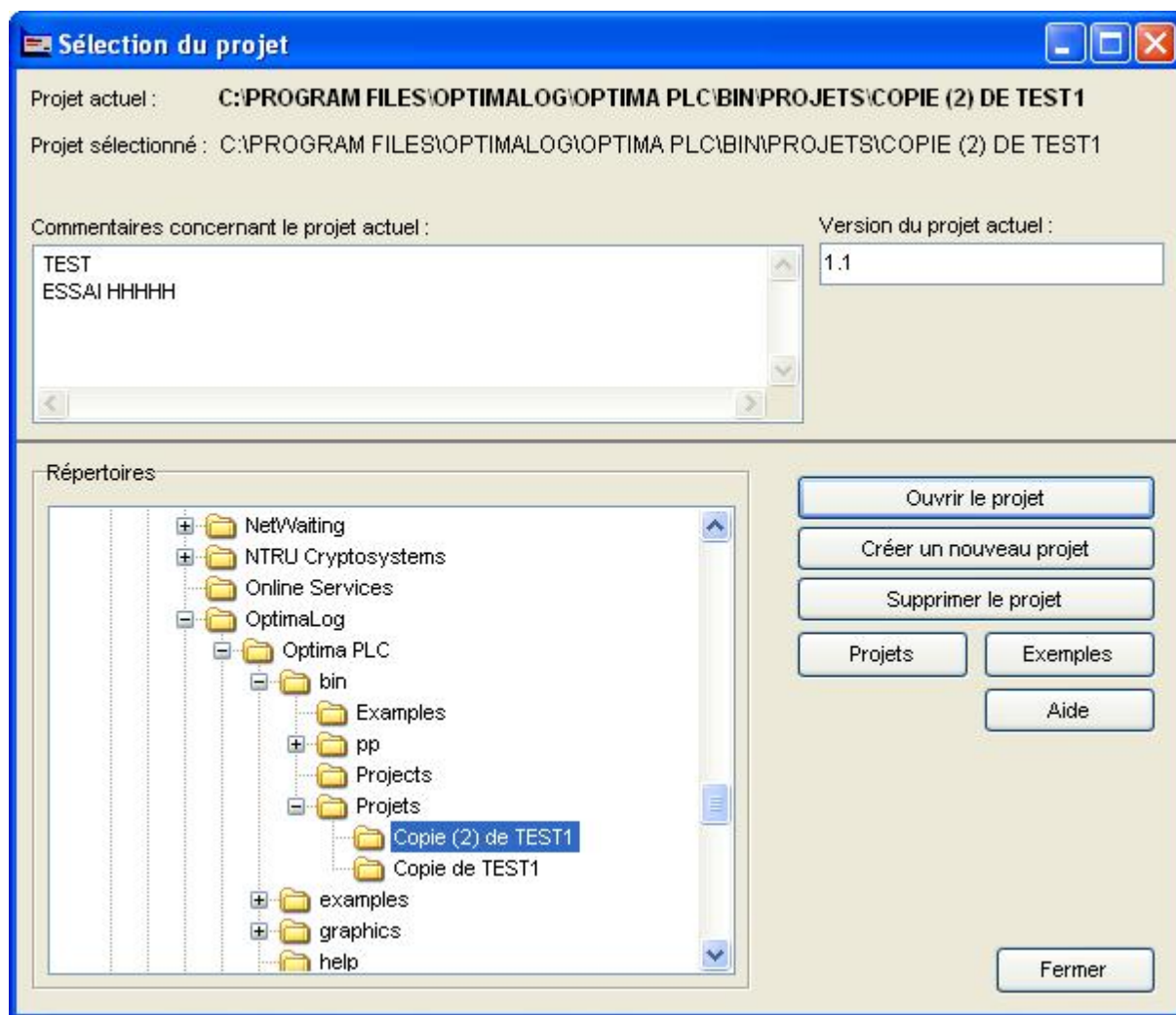
```
"C:\Program Files\OptimaLog\Optima View\Bin\Exe\SUPERGW.EXE" "C:\Program Files\OptimaLog\Optima View\Projets\TEST\Page1.PSGW" 193.0.1.1
```

(Voir [Appel de page avec passage de paramètres](#)<sup>297</sup>)

### 3.14.2 Fenêtre de sélection de projet

#### 3.14.2.1 Sélection d'un projet





La fenêtre de sélection de projet affiche les informations concernant le projet en cours (Nom du projet, commentaires, version, ...). Elle apparaît à chaque démarrage de l'application.

La boîte de dialogue permet de sélectionner le répertoire du projet que l'on veut ouvrir ainsi que le lecteur sur lequel on désire travailler.

La fenêtre présente les contrôles suivants :

#### **Bouton « Ouvrir le projet »**

Ouvre le projet contenu dans le répertoire en surbrillance dans la fenêtre d'exploration des projets.

#### **Bouton « Créer un nouveau projet »**

Crée un nouveau projet dans un sous répertoire du répertoire en surbrillance dans la fenêtre d'exploration des projets.

Attention : La création de projets est seulement autorisée dans le répertoire « PROJECTS » ou dans un des sous répertoires de « PROJECTS »

Si le répertoire courant n'est pas le répertoire "PROJECTS", un message de confirmation demande si on veut créer le projet dans le répertoire "PROJECTS". Répondre "oui" au message de confirmation puis entrer le nom du projet à créer.

#### **Bouton « Supprimer le projet »**

Supprime le projet en surbrillance dans la fenêtre d'exploration des projets. Le projet courant ne peut pas être supprimé.

#### **Bouton « Fermer »**

Ferme la fenêtre de sélection des projets.

#### **Bouton « Projets »**

Ouvre le répertoire des projets.

**Bouton « Exemples »**

Ouvre le répertoire des exemples.

## 4 Runtime Optima PLC

L'exécution d'une application développée avec l'atelier de développement Optima PLC implique l'utilisation du runtime Optima PLC.

Bien qu'une interface homme-machine soit souvent suffisante pour la gestion d'un système, l'environnement du runtime offre des possibilités de visualisation et de modification des variables de l'application.

### 4.1 Généralités

Une fois développée et compilée dans l'atelier Optima PLC, une application est principalement constituée d'un fichier OptimaPLC.prg situé dans le répertoire du projet et contenant le code à exécuter.

Le "runtime" Optima PLC est un programme commun à toutes les applications Optima PLC permettant l'exécution de celles-ci. Il ne s'agit pas d'un interpréteur, mais d'une interface entre le code et Windows d'une part, entre le code et l'utilisateur d'autre part.

Le runtime permet à l'utilisateur de visualiser et de modifier les variables internes de l'application, de suivre le déroulement du programme automate, et d'en assurer la mise au point et le dépannage. Il ne constitue pas une interface conviviale pour l'exploitation journalière par des opérateurs, mais donne accès à des informations importantes pour le développeur et le dépanneur.

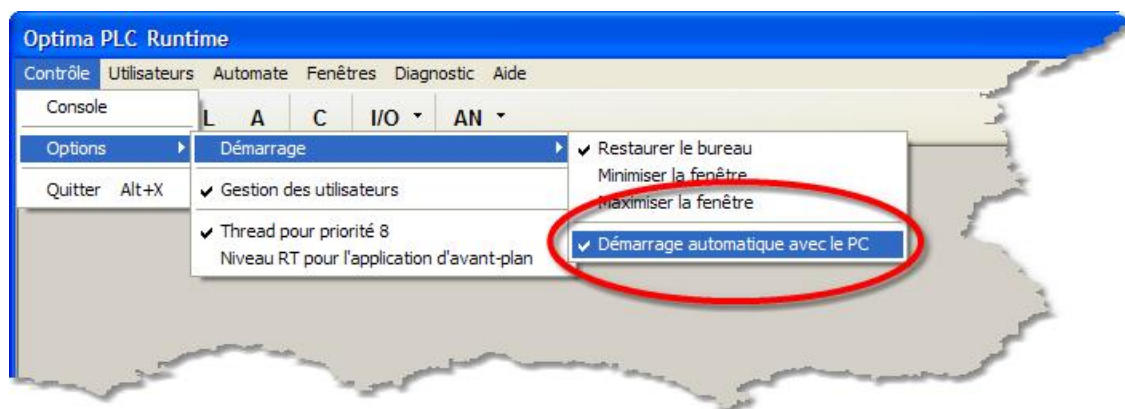
#### 4.1.1 Démarrage de l'application

L'application Optima PLC, une fois compilée, se présente sous la forme d'un fichier OptimaPLC.prg situé dans le répertoire du projet. Plusieurs possibilités existent pour lancer son exécution :

- Lancement direct depuis l'atelier Optima PLC : c'est la méthode la plus simple pendant la phase de mise au point.
- Lancement manuel depuis l'explorateur Windows : double-cliquer sur OptimaPLC.prg pour lancer l'exécution.



- Lancement automatique au démarrage du PC, défini manuellement : créer un raccourci pour le fichier OptimaPLC.prg (menu contextuel par clic droit), et placer ce raccourci dans le répertoire "Démarrage" (Menu Démarrer/Programmes/Démarrage).
- Lancement automatique au démarrage du PC, défini depuis le runtime : utiliser le menu Contrôle/Options/Démarrage/Démarrage automatique avec le PC.



Attention : sous Windows Vista ou Windows Seven, pour permettre un démarrage normal du runtime, il peut être nécessaire de désactiver l'UAC (user account contrôleur), qui peut sinon gêner le démarrage automatique


d'un logiciel nécessitant des droits d'accès au matériel élevés comme Optima PLC (cette désactivation peut être effectuée dans le panneau de configuration, contrôle des comptes utilisateur).

### 4.1.2 Arrêt de l'application

La fermeture de la fenêtre du runtime ne stoppe pas celui-ci, mais le minimise seulement. Son icône reste présente dans la barre de statut :



Pour arrêter complètement le runtime, plusieurs possibilités existent :

- Dans l'atelier Optima PLC, lorsque le runtime a été lancé depuis l'atelier, cliquer sur le bouton .
- Dans le runtime, utiliser le menu Contrôle/Quitter.
- Double-cliquer sur l'icône dans la barre de statut Windows : dans le menu qui apparaît, cliquer sur Quitter.



- Dans le cas d'une fermeture de Windows ou de la session Windows, le runtime est automatiquement stoppé (ne pas forcer immédiatement l'arrêt si cela est proposé par Windows).

A noter qu'une demande d'arrêt par l'utilisateur est toujours suivie d'une demande de confirmation avant l'arrêt complet.

Un arrêt provoque le passage de l'application en phase STOP, durant laquelle certaines tâches programmées peuvent effectuer une finalisation plus ou moins longue. Le runtime stoppe ensuite les drivers de matériels, ce qui peut également prendre un peu de temps, surtout si un problème de communication par exemple est rencontré.

### 4.1.3 Gestion des utilisateurs

Il est possible de restreindre les possibilités d'action dans le runtime en fonction de l'utilisateur. Pour cela, le mode gestion des utilisateurs doit être activé dans le menu Contrôle/Options/Gestion des utilisateurs.

L'activation de ce mode provoque l'apparition du menu Utilisateurs dans la barre de menu.

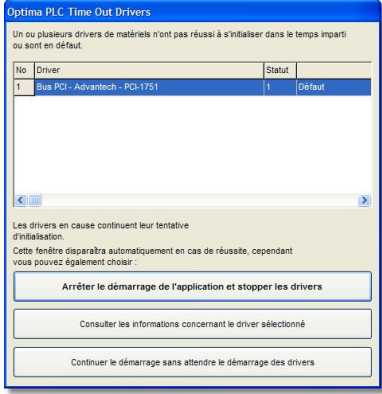
Pour plus d'informations, se reporter au chapitre Gestion des utilisateurs.

### 4.1.4 Statut de l'application

L'état dans lequel se trouve l'application s'affiche dans la barre de statut.

Les différents états sont :

Phase		Interprétation
1	<b>START DRIVERS</b>	Avant de démarrer le programme automate, tous les drivers doivent être initialisés.
2	<b>START DRIVERS TIMEOUT</b>	Si un (ou plusieurs) driver(s) n'a pas pu être initialisé, une liste des drivers non initialisés s'affiche afin d'avertir l'utilisateur. Les tentatives d'initialisation continuent jusqu'à ce que le driver s'initialise correctement, ou que l'utilisateur intervienne.

		
<b>3</b>	<b>START PLC</b>	Les tâches devant s'exécuter dans cette phase d'initialisation du programme automate sont lancées. La phase se termine lorsque toutes ces tâches ont basculé à 1 leur variable StartPLCPhaseOK.
<b>4</b>	<b>RUN</b>	L'application est en cours d'exécution normale.
<b>5</b>	<b>FAULT</b>	L'application est en cours d'exécution mais l'un au moins des éléments matériels liés à l'application (matériel d'entrée/sortie, ...) ne fonctionne pas comme prévu.
<b>6</b>	<b>STOP PLC</b>	Les tâches devant s'exécuter dans cette phase de finalisation du programme automate sont lancées. La phase se termine lorsque toutes ces tâches ont basculé à 1 leur variable StopPLCPhaseOK.
<b>7</b>	<b>STOP DRIVERS</b>	Avant d'arrêter le runtime, tous les drivers doivent être stoppés.
<b>8</b>	<b>STOP DRIVERS TIMEOUT</b>	Si un (ou plusieurs) driver(s) n'a pas pu être stoppé, une liste des drivers non stoppés s'affiche afin d'avertir l'utilisateur. Les tentatives d'arrêt continuent jusqu'à ce que le driver s'arrête correctement, ou que l'utilisateur intervienne.

#### 4.1.5 Performances et temps-réel

Les performances générales de l'application dépendent des options suivantes :

- Activation du noyau temps réel (menu Options/Compilation générale de l'atelier)

Cette option permet d'activer ou non le noyau temps-réel.

S'il est actif, les niveaux de priorité 0 à 7 sont gérés par ce noyau, qui est prioritaire par rapport à tout autre action de Windows. Les performances sont optimales. Par contre il est possible de rencontrer des problèmes de conflit ou de ressource avec ce mode, qu'il est donc conseillé de n'utiliser que sur un PC dédié à l'application.

S'il est inactif, les niveaux de priorité 0 à 9 sont tous gérés par le noyau multitâche préemptif de Windows.

- Tâches Windows de priorité élevée ([menu Contrôle/Options du runtime](#) <sup>334</sup>)

Cette option a pour effet de donner ou non une priorité élevée à l'application.





Si une priorité élevée est donnée, toutes les parties de l'application dépendant de Windows seront prioritaires par rapport aux autres application et aux tâches systèmes de Windows. Si le noyau temps-réel est désactivé, cette option est d'autant plus importante.

Si la priorité normale est laissée, l'application s'exécute de manière classique, avec des périodes pour les tâches automates impossible à maîtriser.

Pour visualiser le niveau des performances, utiliser dans le runtime le [menu Diagnostic/Performances](#) <sup>345</sup>. Cet écran

permet de visualiser les valeurs clés des timings.

Les panneaux affichés à droite ont la signification suivante :

	<p>Performances optimales, noyau temps réel activé, tâches Windows prioritaires.</p>
	<p>Performances optimales des tâches temps-réel, mais les tâches gérées par Windows ne sont pas optimisées (8 et 9).</p>
	<p>Toutes les tâches sont gérées de manière prioritaire par Windows, ce qui permet d'obtenir des performances correctes pour une application d'automatisme classique.</p>
	<p>Toutes les tâches sont gérées par Windows au même niveau de priorité (ou presque) que les autres applications, ce qui ne permet pas de faire fonctionner correctement une application (sauf pour simulation, enseignement, etc...)</p>

## 4.2 Description de l'environnement

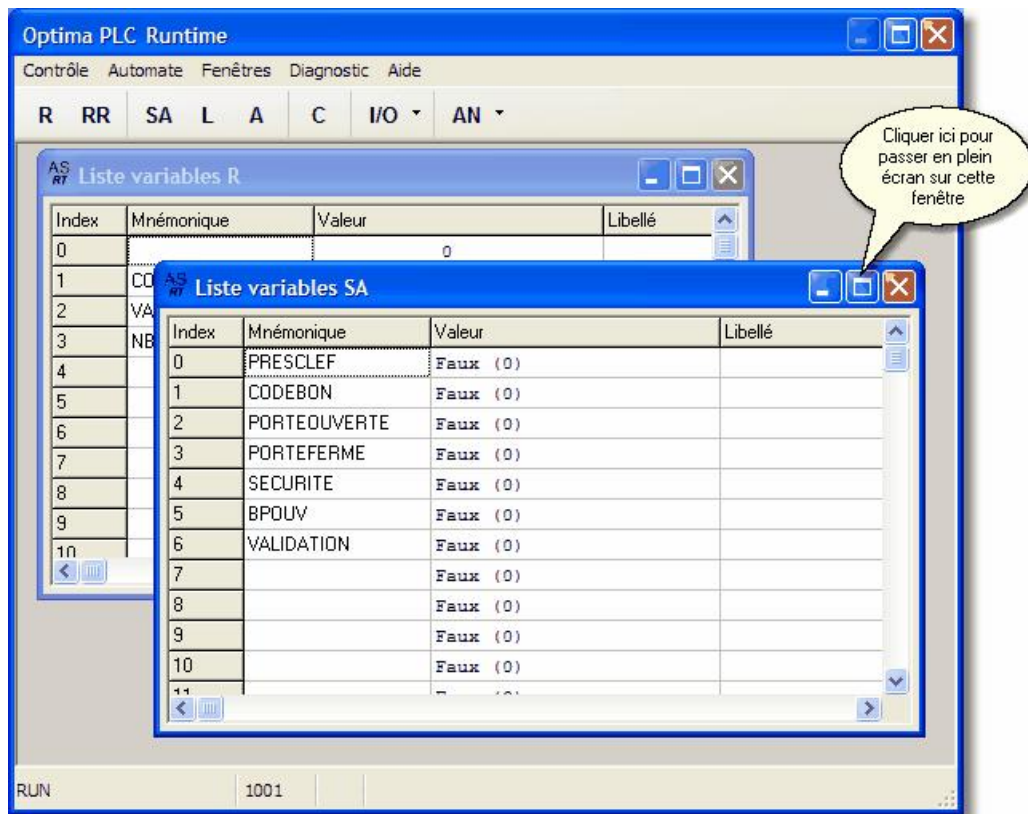
### 4.2.1 Fenêtre de base

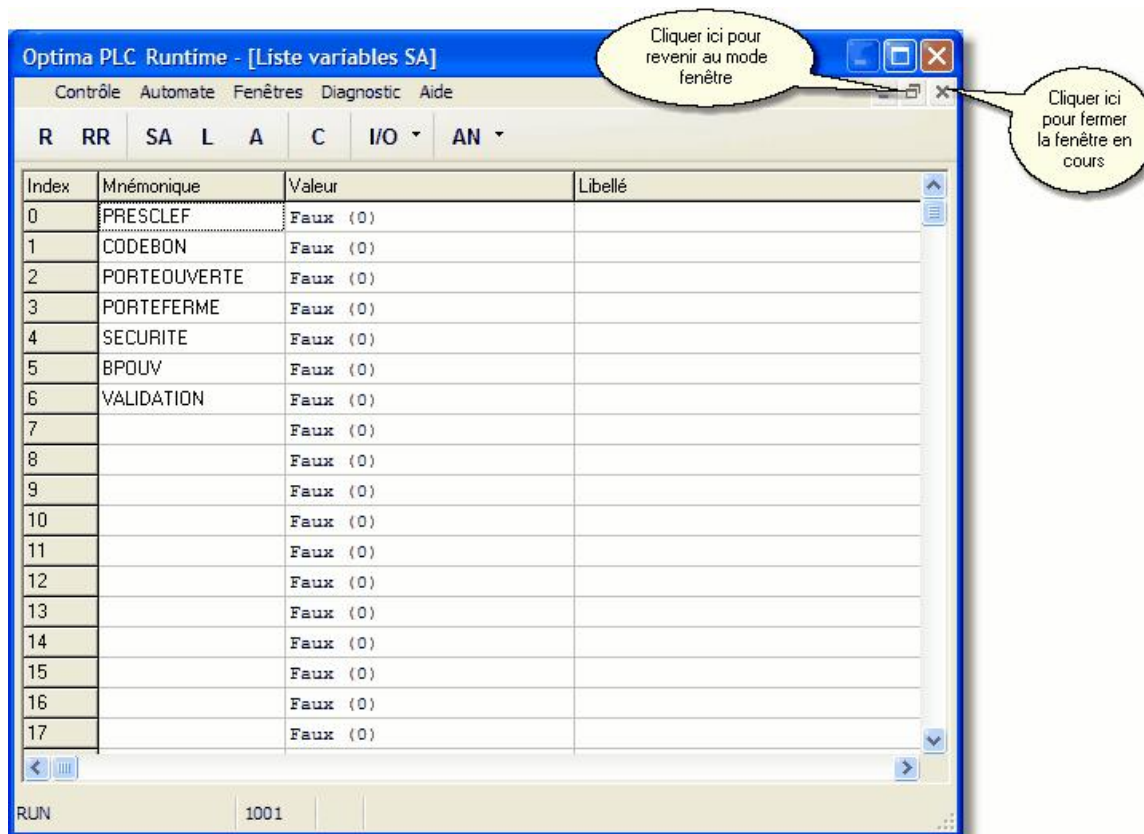
L'environnement du runtime est composé de 4 parties :



- le menu (en haut de la fenêtre)
- la barre d'outil (sous le menu) : elle est composée de boutons permettant l'affichage des différentes variables automates d'Optima PLC
- l'éditeur (au centre de la fenêtre) : pour ouvrir les fenêtres de visualisation
- la barre de statut (en bas de la fenêtre) qui indique l'état de l'application

Les différentes fenêtres de visualisation ou de contrôle ouverte dans le runtime Optima PLC sont incluses dans la fenêtre principale, et peuvent être zoomées selon différents modes :



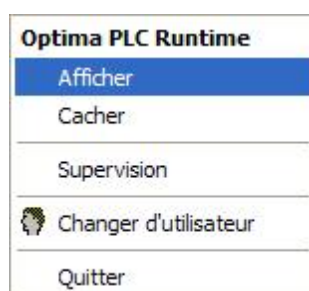


#### 4.2.2 Rappel de la fenêtre

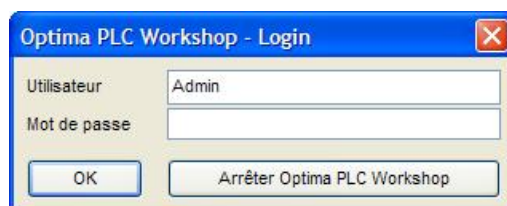
Après une période d'inactivité de l'utilisateur, la fenêtre est automatiquement réduite. Pour la ramener à l'écran, il faut effectuer un double-clic sur l'icône située dans la barre de statut de Windows :



Le menu qui apparaît permet de réafficher la fenêtre du runtime :



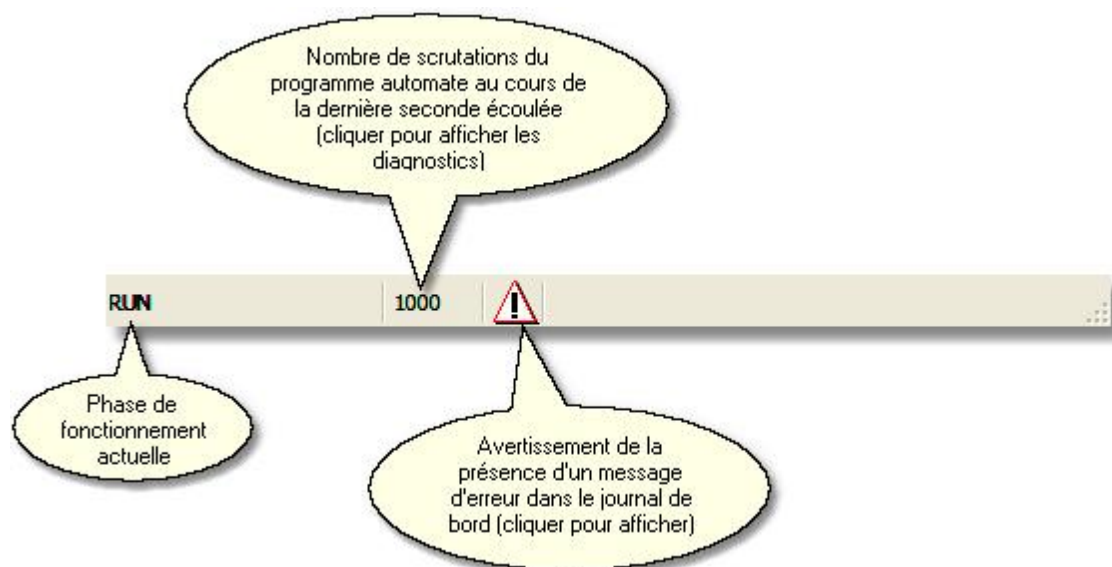
Si la gestion des utilisateurs est activée, il est nécessaire d'entrer son identifiant et mot de passe :



#### 4.2.3 Barre de statut

La barre de statut placée au bas de la fenêtre du runtime fournit quelques informations :





Voir le paragraphe "[Statut de l'application](#)" pour plus d'informations concernant la phase de fonctionnement.

## 4.3 Description des menus

### 4.3.1 Menu Contrôle

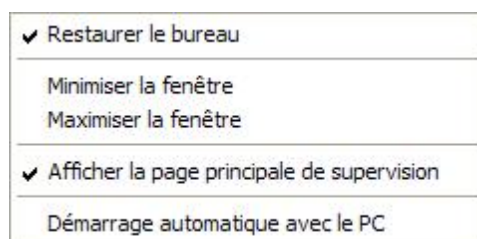
Ce menu permet de modifier les options de fonctionnement du runtime et de quitter l'application.

#### 4.3.1.1 Menu Options

Ce menu donne accès à certaines options de fonctionnement du runtime.

- Démarrage

Cette option donne accès à un nouveau menu :



Option	Effet
Restaurer le bureau	<p>L'aspect du bureau est mémorisé lors de la sortie de l'application pour être restitué au démarrage suivant. Toutes les fenêtres ne sont cependant pas concernées par cette restitution.</p> <p>En mode gestion des utilisateurs, le bureau est sauvegardé et restitué en fonction de l'utilisateur.</p> <p>Les fichiers contenant la description du bureau sont situés dans le sous-répertoire Desk du projet, et leur nom est constitué du nom de l'utilisateur suivi de l'extension .dsk.</p>



Minimiser la fenêtre	Au démarrage, la fenêtre du runtime est minimisée.
Maximiser la fenêtre	Au démarrage, la fenêtre du runtime est maximisée.
Afficher la page principale de supervision	La page de supervision principale (fichier PSGW de même nom que le projet) est affichée au démarrage.
Démarrage automatique avec le PC	<p>Lors du démarrage du PC, le runtime démarre automatiquement avec ce projet.</p> <p>Si un autre projet Optima PLC était configuré en démarrage automatique, le fait de cocher cette option remplace le démarrage du projet précédent par le démarrage du projet courant.</p> <p>Lorsque la case est décochée, plus aucun projet Optima PLC n'est démarré automatiquement.</p> <p>Attention : sous Windows Vista ou Windows Seven, pour permettre un démarrage normal du runtime, il peut être nécessaire de désactiver l'UAC (user account contrôleur), qui peut sinon gêner le démarrage automatique d'un logiciel nécessitant des droits d'accès au matériel élevés comme Optima PLC (cette désactivation peut être effectuée dans le panneau de configuration, contrôle des comptes utilisateur).</p>

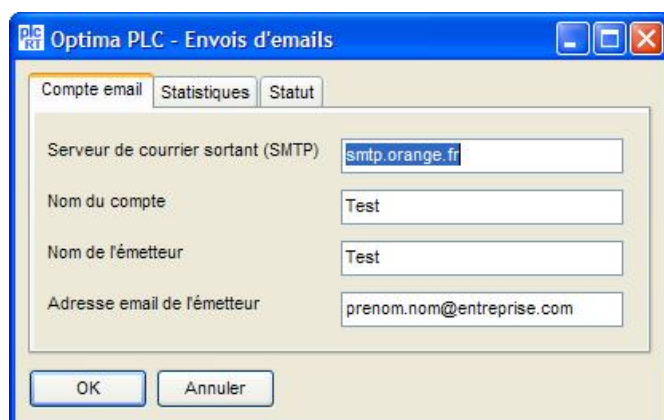
- Gestion des utilisateurs

Cette option permet de choisir le mode de gestion des utilisateurs, dans lequel une identification est demandée préalablement à l'accès aux écrans du runtime. Ce mode offre une meilleure sécurité dans un environnement multi-utilisateurs, comme cela est souvent le cas sur une machine de production.

Lorsque ce mode est activé, les droits attachés à l'utilisateur lui restreignent l'accès à certaines fonctions et menus. De plus, un menu spécial de gestion des utilisateurs est affiché.

- Envoi d'email

Cette option permet de paramétrer et de contrôler la fonction intégrée d'envoi d'emails.



L'onglet "compte email" doit être correctement configuré avec des paramètres de connexion à un serveur d'envoi d'email (SMTP) fonctionnels.

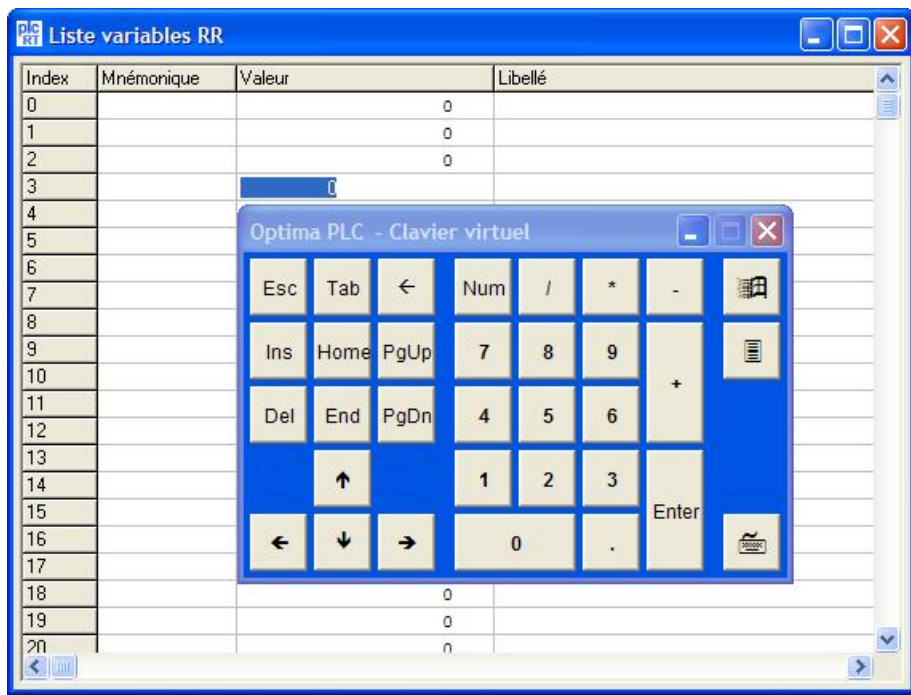
L'onglet statistiques indique le nombre d'emails emis. Enfin, l'onglet Status affiche l'état courant de l'envoi des

emails.

- Clavier virtuel automatique

Cette option, si elle est cochée, indique que le clavier virtuel d'Optima PLC doit être affiché dès qu'une saisie est proposée à l'utilisateur. Ceci est intéressant si l'application est exécutée sur un PC muni d'un écran tactile et sans clavier. L'écran est alors utilisé comme clavier de saisie.

Si la zone de saisie change ou est déplacée, le clavier virtuel est déplacé également. Lorsqu'il n'y a plus de saisie en cours, le clavier est automatiquement caché.



Dans l'exemple ci-dessus, le début de la saisie du registre RR3 provoque l'affichage du clavier virtuel. La fenêtre de ce dernier est placée au plus près de la zone de saisie concernée.

- Tâches Windows de priorité élevée

Cette option permet de donner à la partie Windows de l'application la priorité maximale (Real Time). Ceci est conseillé pour améliorer les performances générales de l'application, en particulier celles non basées sur le noyau temps réel intégré mais sur Windows.

- Période pour tâches de priorité 8

Ce sous-menu, visible uniquement si le noyau temps-réel est activé, permet de définir la période d'exécution des tâches de priorité 8. Pour rappel, ce niveau de priorité n'est pas temps réel, mais géré par Windows. La période par défaut est de 10 ms. Il est possible de faire varier celle-ci de 2 à 20 ms.

- Sous menu "Temps réel"

Ce sous menu permet de paramétrer la tolérance sur la période de base des traitements cycliques. Si la période constatée sort de la tolérance, un message d'avertissement est affiché. La tolérance est définie par un pourcentage appliqué sur la période. Il est conseillé d'effectuer quelques essais avant de fixer une valeur trop basse.

- Sous-menu "Sauvegarde données retain"

Les données retain sont périodiquement sauvegardées dans des fichiers sur disque. Ce sous-menu permet de régler l'intervalle de sauvegarde (en secondes) et le nombre de fichiers conservés.

#### 4.3.1.2 Option Licences

Cette option permet d'afficher les informations courantes concernant la licence d'utilisation du logiciel.

Optima PLC - Licences d'utilisation

Optima PLC

Localisation

☐ Aucune clé

☒ Clé USB locale

☐ Clé USB distante

☐ PC

No de série 105277

Type de licence Gold

Numéro de licence 612002

Début/fin de validité 09/12/2006

Nom utilisateur Dupont Automatismes

Code utilisateur 113

Fin de maintenance logicielle

Fin d'assistance technique

☒ Atelier logiciel

☒ Runtime

☐ Mono-application

☒ OPTIMA View

Nb Max I/O TOR

Nb Max I/O Ana

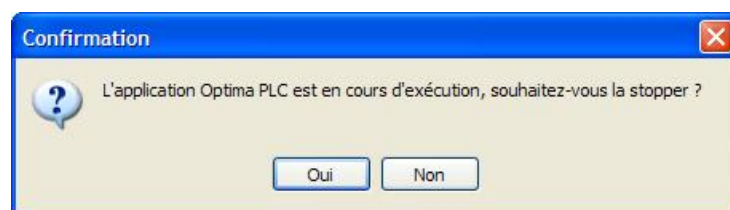
Actualiser

Fermer

Modification Licences

#### 4.3.1.3 Option Quitter

Cette option provoque l'arrêt complet du runtime, après une demande de confirmation :



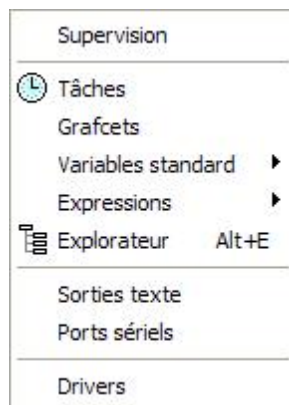
#### 4.3.2 Menu Utilisateurs

Ce menu regroupe les options permettant de gérer la liste des utilisateurs autorisés. Il n'apparaît que si le mode "gestion des utilisateurs" a été activé dans les options.

Les options de ce menu sont communes aux différents logiciels Optimalog. Pour plus d'informations, voir le chapitre Gestion des utilisateurs.

#### 4.3.3 Menu Automate

Il est composé de sous-menus permettant l'accès aux variables et informations diverses disponibles pour le fonctionnement de l'automate.



#### 4.3.3.1 Option Supervision

Cette option provoque l'affichage de la supervision associée à l'application.

Si aucune page de supervision n'est encore affichée, la page principale est chargée et affichée. Cette page correspond au fichier situé dans le sous-répertoire PSGW, de même nom que le projet.

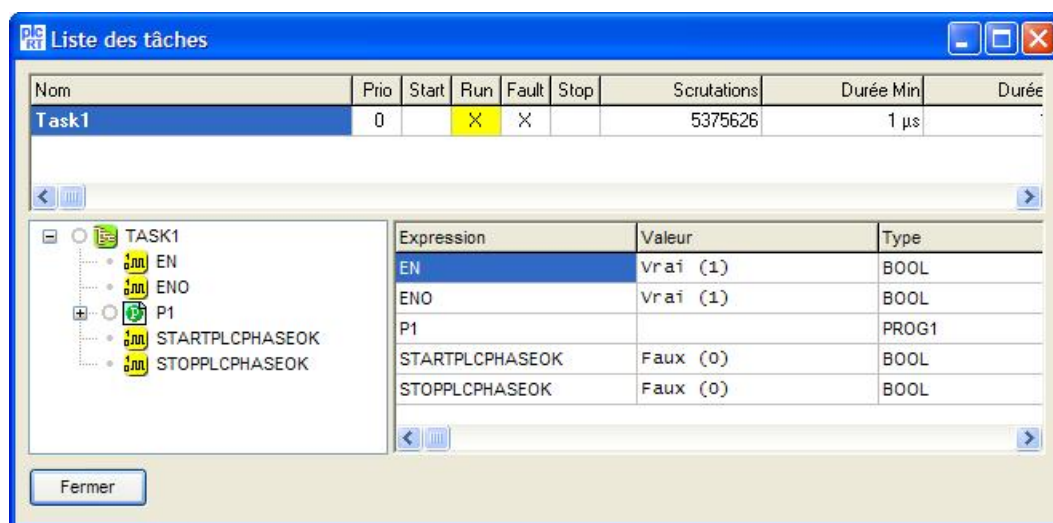
Si des pages sont déjà affichées, cette option provoque leur mise au premier plan.

Lorsque des pages de supervision sont affichées, la sortie de l'application provoque la fermeture des pages.

#### 4.3.3.2 Option Tâches

Il est possible avec cette option de faire apparaître la liste des tâches du programme.

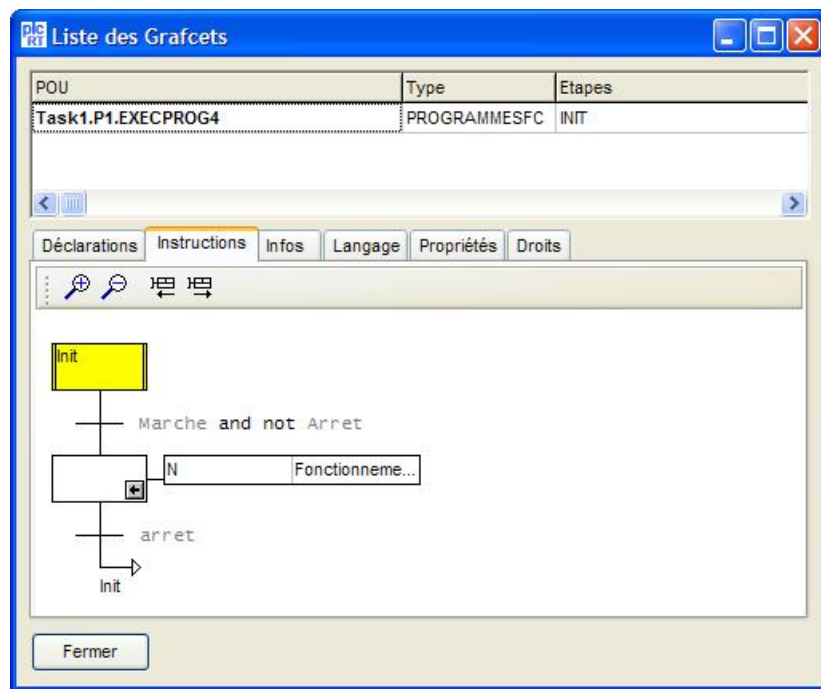
On peut y visualiser différents paramètres propre à chaque tâches comme la durée minimum et maximum d'une scrutation, ou bien encore son degré de priorité par rapport aux autres tâches de l'application.



La sélection d'une tâche dans la liste provoque l'affichage dans la partie basse de la fenêtre d'un explorateur de variables basé sur l'arbre d'exécution de cette tâche.

#### 4.3.3.3 Option Graficets

Il est assez courant de commencer la mise au point ou le dépannage d'un automatisme en vérifiant les séquences du système et les étapes actives. L'objectif de cette option est de faciliter l'accès à ces séquences en les présentant toutes sous forme d'une liste affichant, pour chacune, les étapes actives.



La sélection d'une séquence dans la liste située en partie haute de la fenêtre provoque l'affichage du graphe de la séquence en partie basse. La ou les étapes actives sont affichées dans une couleur différente.

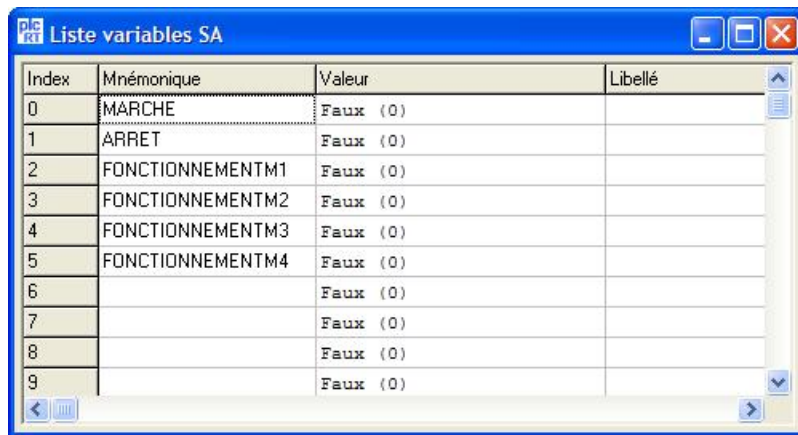
#### 4.3.3.4 Option Variables standard

Ce menu permet de visualiser en temps réel et parfois de modifier l'état des variables automates d'Optima PLC à savoir :

Registres R	Ctrl+R
Registres RR	
Flags SA	
Flags L	Ctrl+L
Entrées TOR I	Ctrl+I
Sorties TOR O	Ctrl+O
Entrées analogiques ANI	
Sorties analogiques ANO	
Chaînes C	

- les registres R
- les registres réels RR
- les flags SA
- les flags L (non modifiables)
- les entrées sorties TOR ou analogiques (peuvent éventuellement être forcées à une valeur si les droits de l'utilisateur le permettent)
- les chaînes C

Voici la fenêtre correspondant aux variables SA :



Index	Mnémonique	Valeur	Libellé
0	MARCHE	Faux (0)	
1	ARRET	Faux (0)	
2	FONCTIONNEMENTM1	Faux (0)	
3	FONCTIONNEMENTM2	Faux (0)	
4	FONCTIONNEMENTM3	Faux (0)	
5	FONCTIONNEMENTM4	Faux (0)	
6		Faux (0)	
7		Faux (0)	
8		Faux (0)	
9		Faux (0)	

Seules les valeurs des variables peuvent être modifiées (les mnémoniques et libellés, définis par programmation, ne peuvent être que visualisés).

Pour modifier une valeur, il faut saisir la nouvelle valeur dans la case appropriée du tableau puis la valider avec la touche "Entrée" du clavier.

Le forçage des entrées sorties TOR s'effectue en sélectionnant l'état à modifier puis en utilisant le menu contextuel qui propose 3 options :

- pas de forçage
- forçage à 0
- forçage à 1

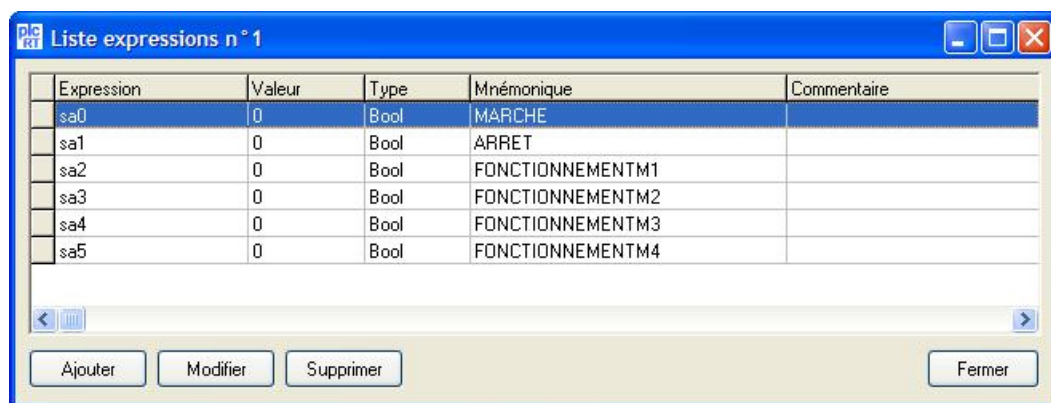
Le forçage des entrées sorties analogiques s'effectue en 2 temps :

- la valeur à affecter doit être saisie dans la colonne "Forçage à" (si cette colonne n'apparaît pas, cela signifie que vous n'avez pas le droit de forcer les valeurs)
- le menu contextuel permet alors de forcer l'entrée ou la sortie à cette valeur

#### 4.3.3.5 Option Expressions

Une "expression" désigne une variable simple ou une formule calculée à partir de variables de l'application.

Les "listes d'expression" constituent un outil de visualisation facilement configurable pour l'utilisateur, permettant de regrouper dans une même fenêtre des variables plus ou moins disparates mais dont la consultation simultanée présente un intérêt pour la mise au point ou le dépannage.



Expression	Valeur	Type	Mnémonique	Commentaire
sa0	0	Bool	MARCHE	
sa1	0	Bool	ARRET	
sa2	0	Bool	FONCTIONNEMENTM1	
sa3	0	Bool	FONCTIONNEMENTM2	
sa4	0	Bool	FONCTIONNEMENTM3	
sa5	0	Bool	FONCTIONNEMENTM4	

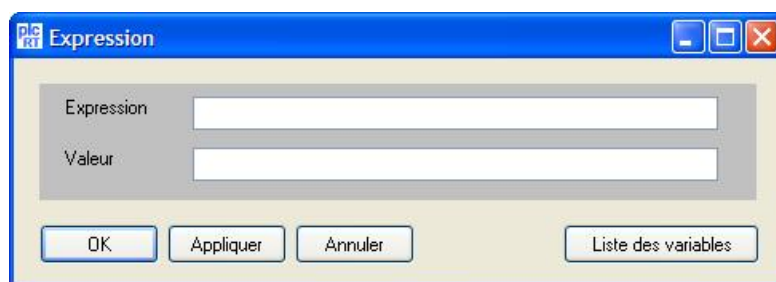
Ajouter   Modifier   Supprimer   Fermer

Toutes les variables prédéfinies ou définies par l'utilisateur peuvent être utilisées dans cet outil.

4 listes d'expressions peuvent être créées.

Dans chacune de ces listes il est possible d'ajouter, de modifier ou de supprimer l'affichage d'une expression. Lorsqu'une expression est affichée dans une liste, sa valeur est mise à jour périodiquement.

L'ajout ou la modification d'une expression font apparaître la fenêtre suivante :



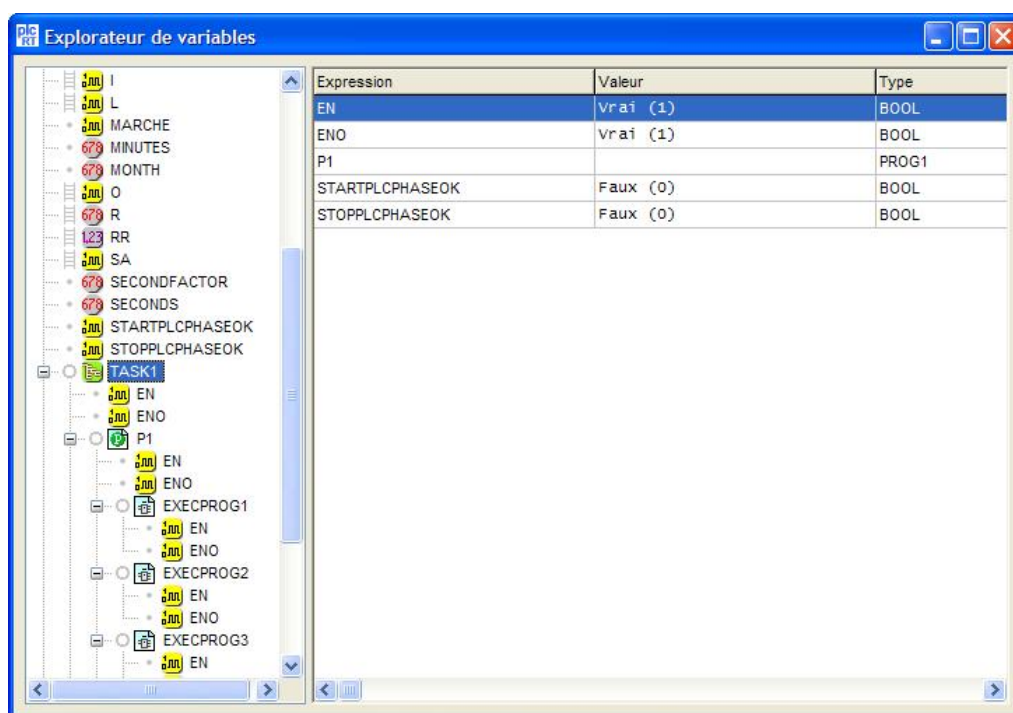
Lorsque le nom de la variable est modifié, la valeur est automatiquement mise à jour (à condition que la variable existe).

Lorsque la valeur est modifiée, elle doit être validée (touche "Entrée" du clavier ou bouton "OK" de la fenêtre) pour être affectée à la variable.

Une expression peut également être créée à partir d'un explorateur de variables en faisant glisser la variables souhaitée au moyen de la souris, et en la lâchant sur la fenêtre des expressions.

#### 4.3.3.6 Option Explorateur

Lorsqu'on ne connaît pas le nom exact d'une expression ou lorsqu'on souhaite simplement consulter sa valeur sans l'ajouter dans une des listes, il est possible d'accéder à la liste complète des expressions de l'application en cliquant sur le bouton "Explorateur".



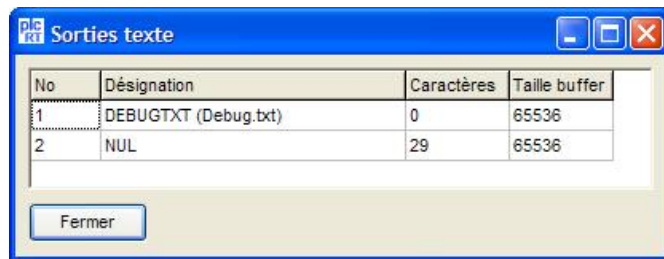
La fenêtre qui apparaît se compose de deux parties :

- un panneau à gauche contenant l'arbre d'exécution.
- Une liste à droite présentant les variables contenues dans le niveau d'arborescence sélectionné.

#### 4.3.3.7 Option Sorties Texte

Les fichiers de "sortie texte" d'Optima PLC permettent de définir simplement dans l'application un flux de sortie "au fil de l'eau", dans lequel le programme peut ajouter du texte destiné à être imprimé ou stocké sur disque.



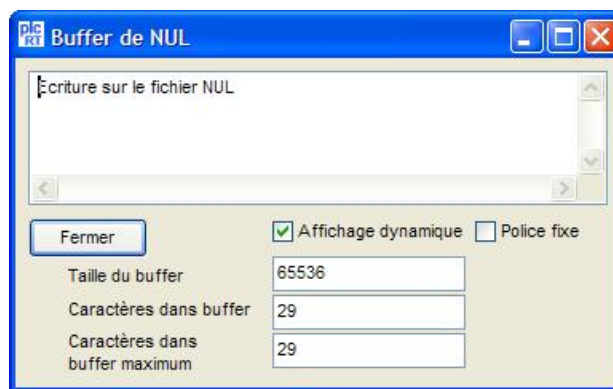


Pour chaque sortie sont mentionnés le nombre de caractères situés dans le buffer, ainsi que la taille de ce buffer.

Deux fichiers de sortie texte sont disponibles de base, NUL (pas de stockage ni d'impression), et DebutTxt (stockage dans le fichier Debug.txt situé dans le répertoire du projet).

- NUL : fichier temporaire créé automatiquement au démarrage de l'application, son contenu peut être visualisé depuis le runtime cependant il est supprimé dès que l'application est arrêtée
- DEBUGTXT : fichier créé (ou ouvert) automatiquement au démarrage de l'application, son contenu peut être visualisé depuis le runtime, il est ensuite sauvegardé à l'arrêt de l'application

Pour visualiser le contenu d'un fichier, double-cliquer sur le nom du fichier.



Une liste permet de consulter le contenu instantané du buffer, qui peut éventuellement évoluer rapidement selon l'usage qui est fait de la sortie dans le programme.

Pour figer cet affichage et consulter plus facilement le contenu d'une sortie fréquemment utilisée, décocher la case "affichage dynamique",

La case "Police fixe" provoque l'utilisation d'une police de pas fixe au lieu d'une police proportionnelle, ce qui est utile si la sortie contient des informations à emplacement fixe sur les lignes.

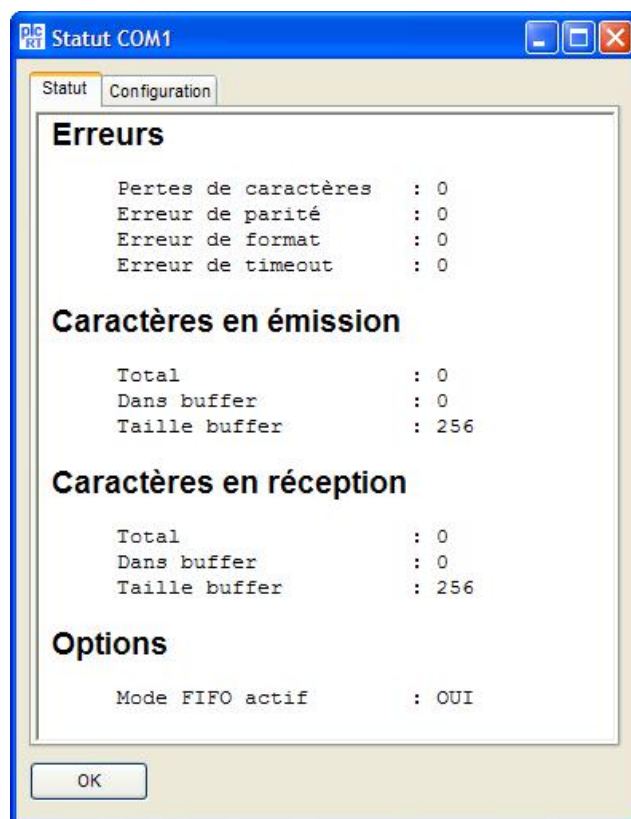
#### 4.3.3.8 Option Ports sériels

Il est possible de visualiser la liste des ports sériels utilisés dans l'application, ainsi que leurs paramètres (qui ne peuvent pas être modifiés ici).



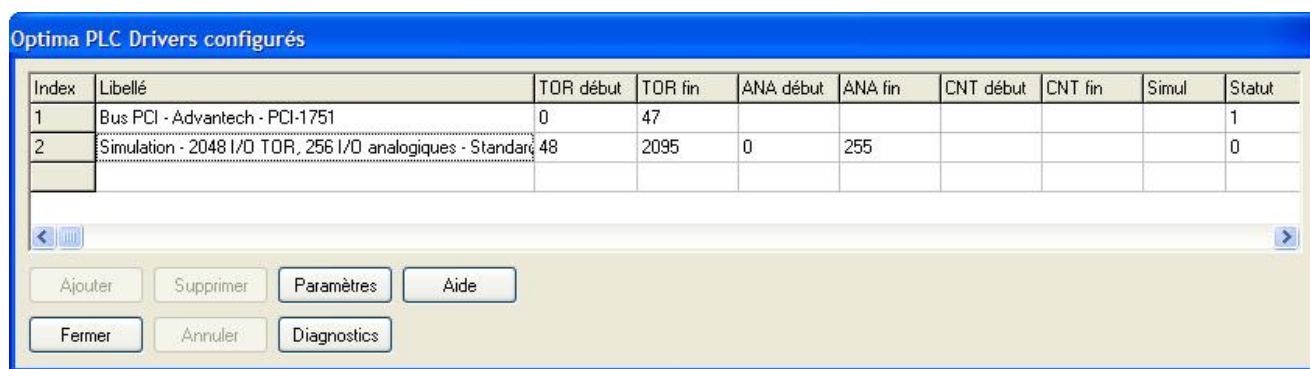


Pour obtenir plus d'informations sur un port COM utilisé, double-cliquer dessus ou bien utiliser le bouton Sélectionner.



#### 4.3.3.9 Option Drivers

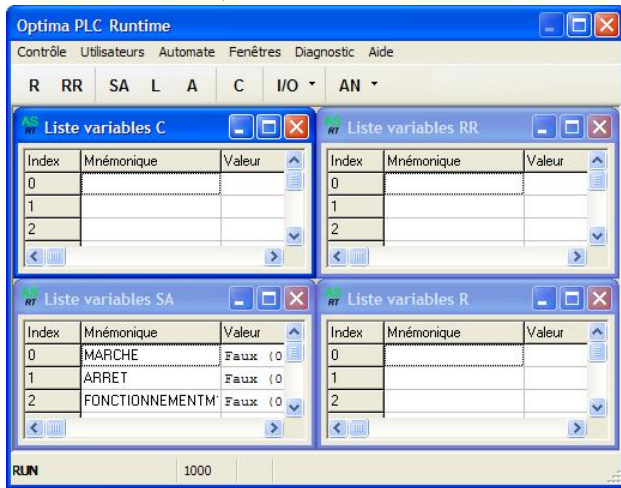
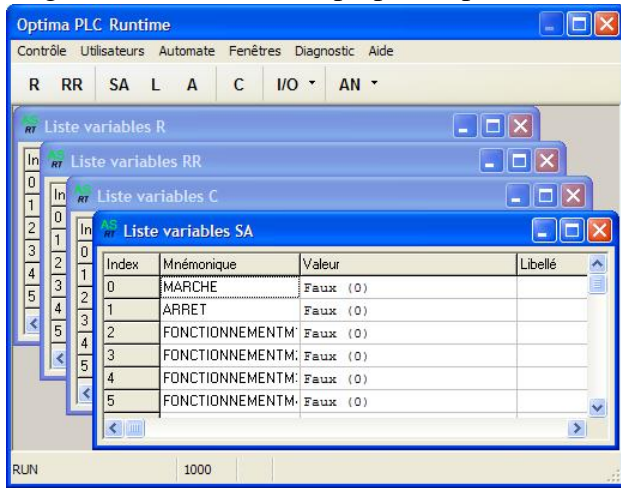
Il est possible ici de visualiser la liste des drivers utilisés dans l'application, ainsi que leurs paramètres.



#### 4.3.4 Menu Fenêtres

Ce menu est dédié à la gestion des fenêtres affichées dans le runtime.



Option	Effet
Tout fermer	Ferme toutes les fenêtres ouvertes dans le runtime.
Mosaïque	<p>Crée une mosaïque avec les fenêtres ouvertes afin que toutes soient visibles (ne concerne pas toutes les fenêtres).</p> 
Cascade	<p>Cascade l'affichage des fenêtres en les superposant partiellement.</p> 
Précédente	Active la fenêtre précédente dans celles affichées.
Suivante	Active la fenêtre suivante dans celles affichées.
Clavier virtuel	<p>Appelle le clavier virtuel Optima PLC, utile avec un écran tactile. Cette option n'apparaît que si l'option "Clavier virtuel automatique" n'est pas cochée dans <a href="#">Contrôle-Options</a> <sup>334</sup> (dans ce cas le clavier virtuel est affiché automatiquement dès qu'une saisie est proposée).</p>
Liste des fenêtres	Affiche un sous menu contextuel permettant de choisir parmi les fenêtres ouvertes celle à activer.

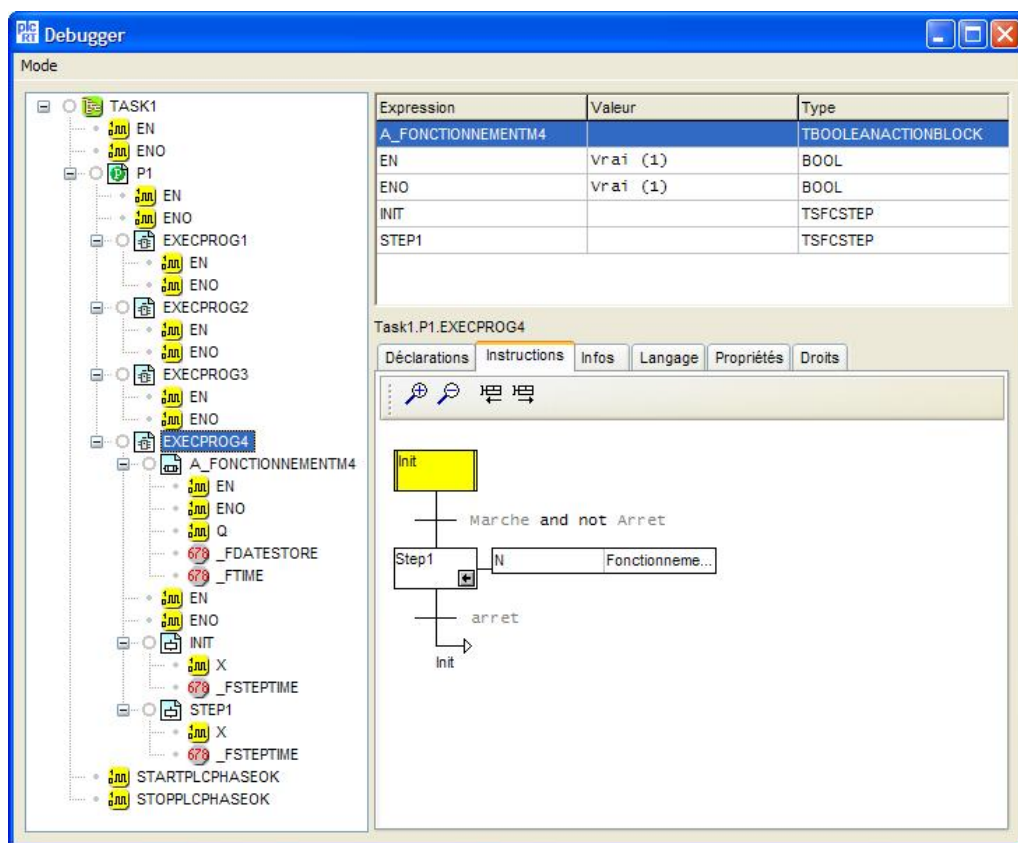
### 4.3.5 Menu Diagnostics

Ce menu donne accès à des fonctions de mise au point diverses.

Debugger	Alt+D
Performances	
Journal de bord	Alt+J
Server OPC	
Liste des sockets	
Générer un enregistrement de diagnostic	
Position d'erreur	

#### 4.3.5.1 Option Debugger

Cette option provoque l'affichage du debugger, qui permet un suivi poussé du fonctionnement du programme.

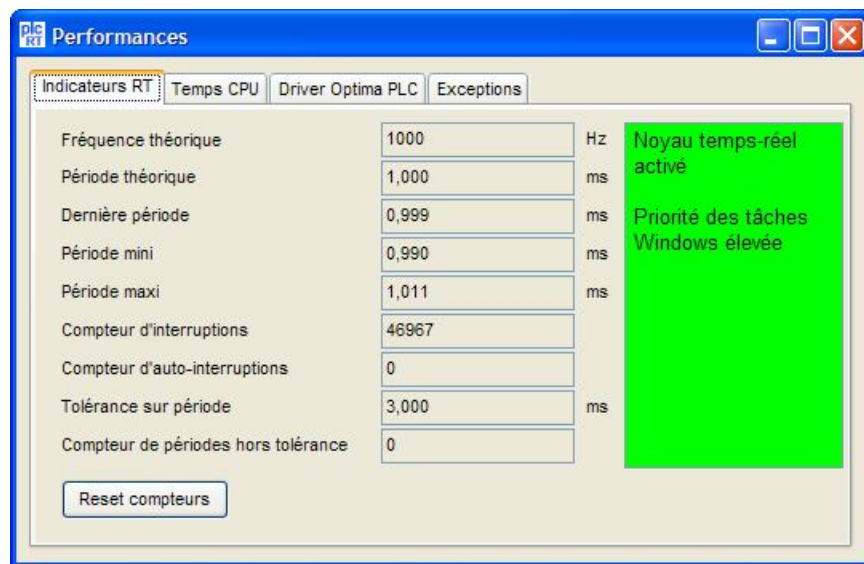


Debugger utilisé avec un POU défini en SFC

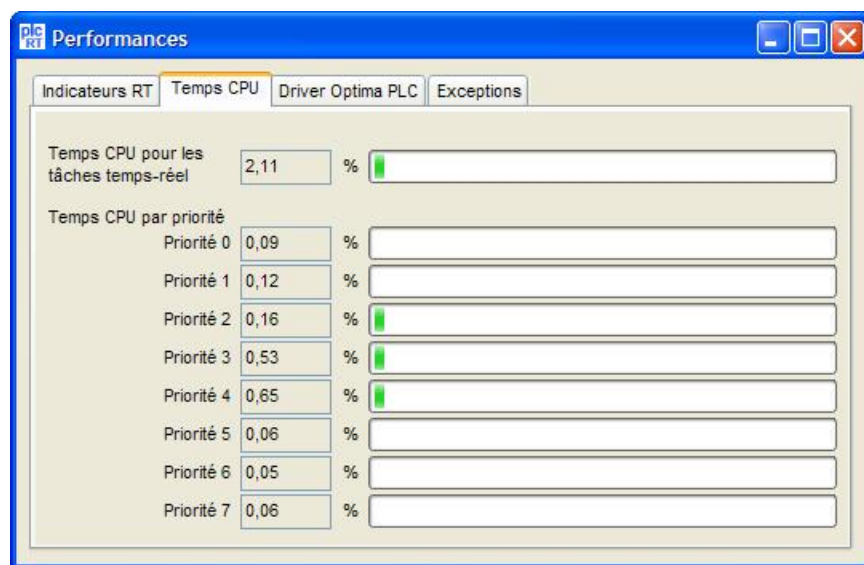
Cet écran comporte un explorateur de variables ainsi qu'un volet de visualisation dynamique des POU sélectionnés.

#### 4.3.5.2 Option Performances

La fenêtre affichée ici présente des informations concernant le fonctionnement temps-réel de l'application et sur l'occupation du CPU par le programme automate.



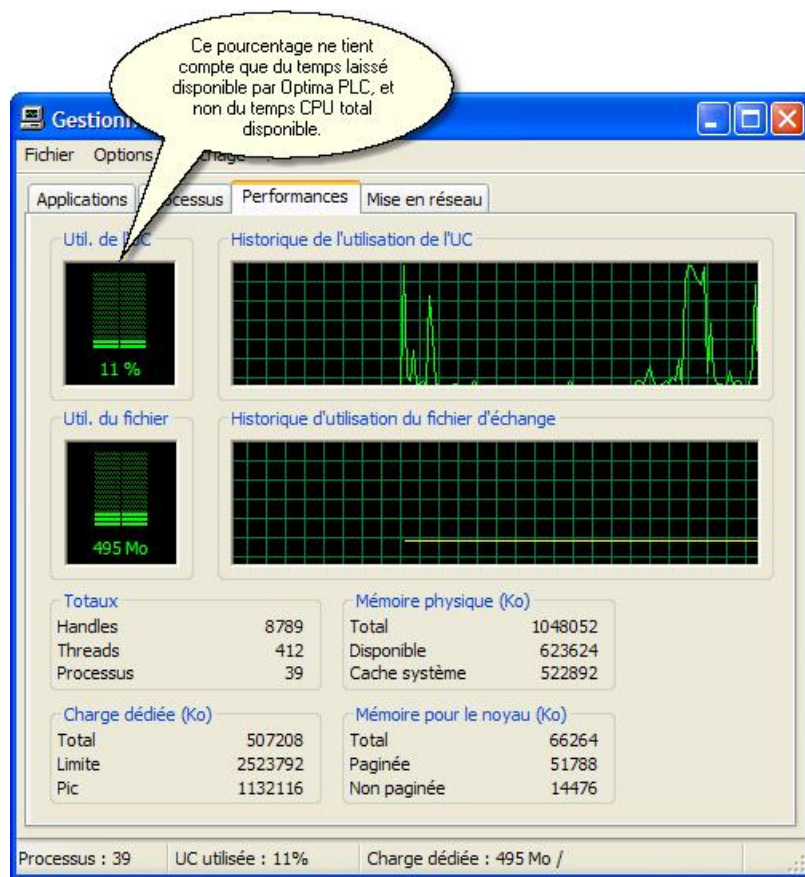
L'onglet "Indicateurs RT" regroupe les principales informations concernant le temps-réel. Le plus important étant le strict respect de la fréquence théorique des tops d'exécution du programme, les points à surveiller sont la période maxi et le compteur de périodes hors tolérance, qui peuvent conduire à diagnostiquer, en cas d'anomalie, la présence d'un matériel ou d'un driver venant perturber les fonctions temps-réel d'Optima PLC.



L'onglet "Temps CPU" permet de contrôler l'usage du CPU effectué par le runtime Optima PLC. Le temps global consommé par les tâches temps-réel est affiché globalement (en pourcentage du temps disponible), puis sont affichées les pourcentage pour chaque niveau de priorité temps-réel (0 à 7).

Le temps CPU consommé par les tâches non temps-réel n'est pas comptabilisé ici (niveaux 8 et 9).

A noter que le pourcentage du temps CPU utilisé, affiché par Windows, correspond au pourcentage calculé par rapport au temps restant non consommé par Optima PLC.

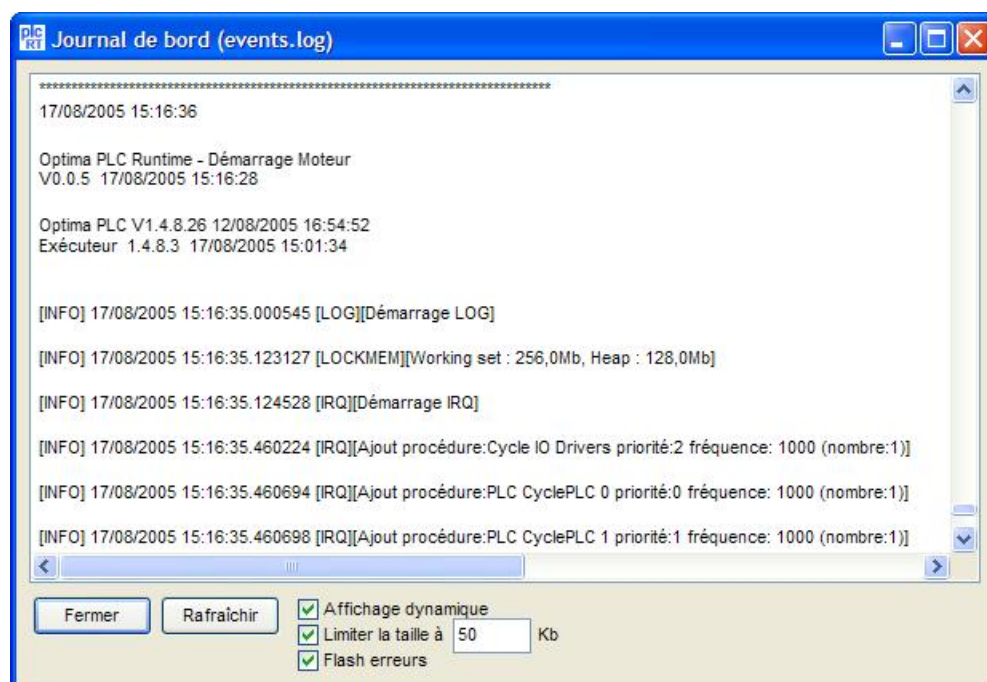


Dans la mesure du possible, il est préférable que le temps total consommé par Optima PLC reste relativement faible et ne dépasse pas en moyenne 30% environ du temps CPU disponible. Si ce pourcentage est dépassé, certaines fonctions de Windows risquent d'être perturbées, en particulier celles faisant un usage important des interruptions, comme les fonctions réseaux par exemple.

Pour réduire le pourcentage d'occupation du CPU par le runtime Optima PLC, un moyen simple consiste à augmenter la période d'exécution des tâches de l'application qui ne nécessitent pas une période aussi faible que la valeur par défaut de 1 milliseconde.

#### 4.3.5.3 Option Journal de bord

Cette option affiche la fenêtre de visualisation du journal de bord du runtime Optima PLC.



Le journal de bord reçoit les messages d'information ou d'erreur du système, mais également des messages définis par le programmeur.

Ces messages sont stockés dans le fichier Events.log situé dans le répertoire d'installation OptimaPLC\log.

L'affichage est effectué en dynamique ; dans le cas où de nombreux messages sont placés dans le journal de bord, décocher la case "Affichage dynamique" pour figer le contenu de la fenêtre.

La case "Limiter la taille à", associée à la saisie située à droite, permet de limiter la taille du fichier Events.log.

Enfin, la case "Flash erreurs" permet d'activer le clignotement de la fenêtre du runtime au cas où un message d'erreur est placé dans le journal de bord, jusqu'à la consultation de celui-ci.

## Position d'origine d'une erreur

Lorsqu'une condition d'erreur survient, celle-ci est signalée dans le journal de bord. Dans l'exemple suivant, une division par 0 :

```
[ERROR] 9/05/2006 18:02:24.710669 [IRQ][Divide Error (Code 200, @ 96D4DA4)]
    96D4DA4 Fichier:C:\FVCS\OptimaPLC\TEMP\_ST_PROG2.ST Proc:_FB_PROG2.EXECUTE Ligne:114
Colonne:6
    10098C41 Fichier:c:/fvcs/OptimaPLC/fpc/plc/st_procs.aps Proc:TFUNCTIONBLOCK__DOEXECUTE
Ligne:1359 Colonne:1
    96CD023 Fichier:C:\FVCS\OptimaPLC\TEMP\_ST_TASK1.ST Proc:_FB_TASK1.EXECUTE Ligne:29
Colonne:1
    10099B98 Fichier:c:/fvcs/OptimaPLC/fpc/plc/st_procs.aps Proc:TTASK__DOEXECUTETASK
Ligne:1705 Colonne:1
    1009B784 (non trouvé)
    100B393B Fichier:c:/fvcs/OptimaPLC/fpc/lib/collec.aps Proc:CALLPOINTERLOCAL Ligne:134
Colonne:1
    100B3D6A Fichier:c:/fvcs/OptimaPLC/fpc/lib/collec.aps Proc:TCOLLECTION__FOREACH Ligne:296
Colonne:1
    1009B8B8 Fichier:c:/fvcs/OptimaPLC/fpc/plc/plc.aps Proc:CYCLEPLC Ligne:190 Colonne:1
```

La première ligne indique la nature de l'erreur, sa date et sa position dans l'espace mémoire. Les lignes suivantes correspondent au contenu de la pile d'appels au moment de l'erreur, et permettent de localiser celle-ci.

Les fichiers mentionnés sont importants : avec l'extension .ST, les fichiers correspondent à des POU de l'application. Un double clic sur la ligne correspondante provoque alors l'affichage du POU dans l'atelier si celui-ci est en fonction à ce moment.

Les fichiers avec l'extension .APS correspondent aux modules internes d'Optima PLC.

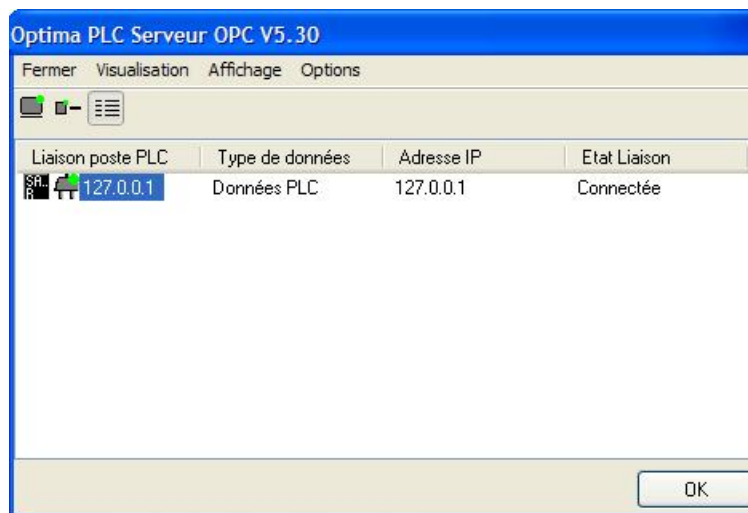
Dans l'exemple ci-dessus, un double clic sur la première ligne en retrait (concernant le fichier \_ST\_PROG2.ST) provoque l'affichage du POU Prog2 dans l'atelier, le curseur placé à l'emplacement du code ayant provoqué la division par 0.

### 4.3.5.4 Option Serveur OPC

Cette option permet l'affichage des informations courantes concernant le serveur OPC intégré à Optima PLC.

Ces informations ne sont disponibles que si le serveur OPC est en fonctionnement, ce qui n'est le cas que si au moins un client tente de s'y connecter. Le client le plus fréquemment connecté est la supervision intégrée à Optima PLC, mais il peut également s'agir d'une application cliente extérieure.

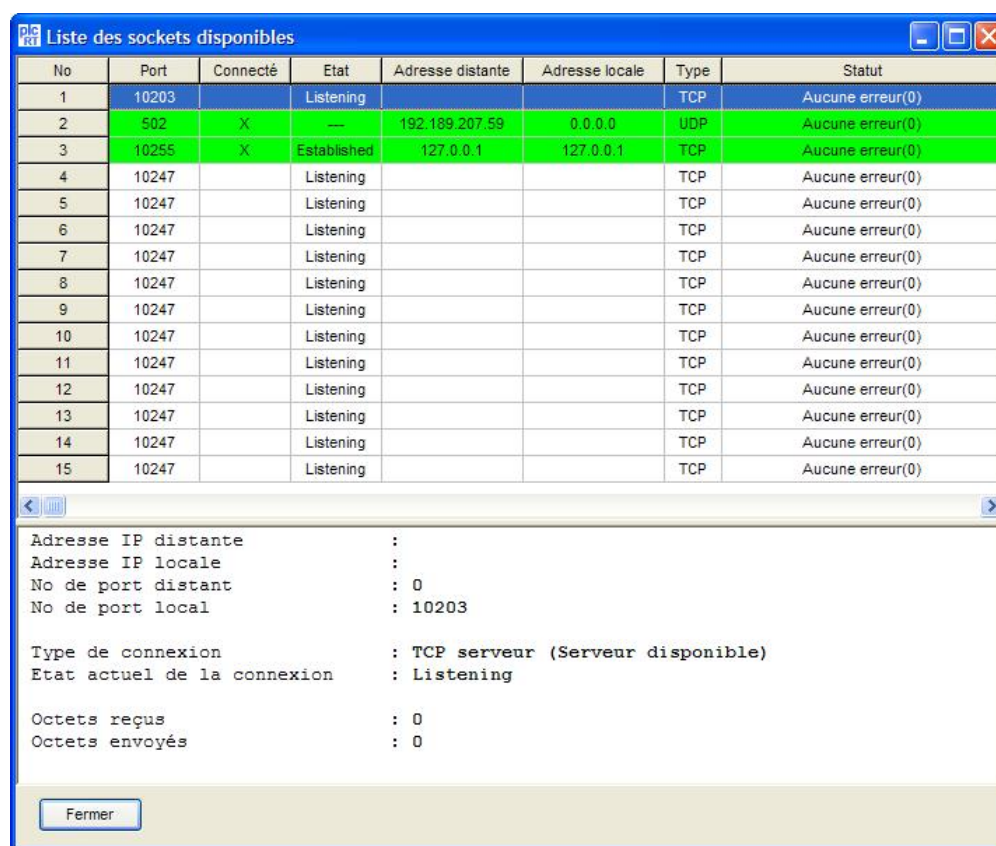




La fenêtre affichée permet de connaître l'état actuel de la connexion. L'adresse IP de l'automate connecté est mentionnée (ici 127.0.0.1 correspond à l'adresse locale, aussi définie par "Local\_Host").

#### 4.3.5.5 Option Liste des sockets

Cette option provoque l'affichage d'une fenêtre de diagnostic listant tous les sockets (liaisons réseaux) gérées par l'application.



Pour chaque socket sont mentionnés diverses informations utiles : son état connecté ou non, l'adresse distante, les numéros de port, etc...

Cette liste contient à la fois les sockets basés sur TCP (liaisons avec protocole d'établissement d'une connexion) et sur UDP (liaisons simples par télégramme, pas de protocole d'établissement de connexion).

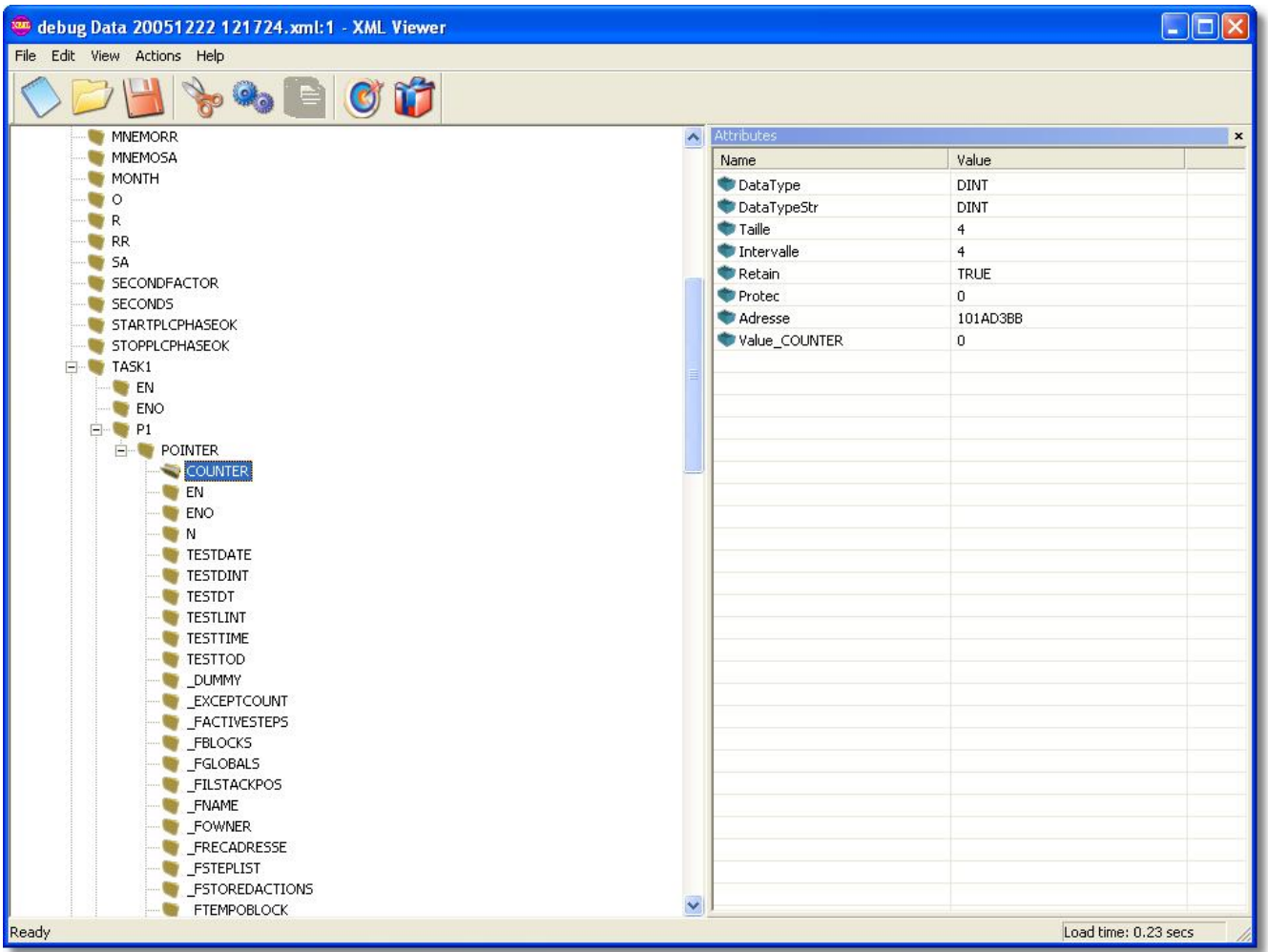
#### 4.3.5.6 Option "Générer un enregistrement de diagnostic"

Cette option provoque la sauvegarde des données complètes de l'automate dans un but de diagnostic ou de mise au point.

Le nom du fichier est "Debug Data" suivi de la date et de l'heure. L'extension est .XML.

Ce fichier est également généré en sortie d'application si l'option "Informations de mise au point" a été sélectionnée pour la compilation.

Le fichier généré peut être consulté dans un programme de visualisation XML, comme par exemple le logiciel gratuit "XML Viewer" de MindFusion (<http://www.mindfusion.org>).

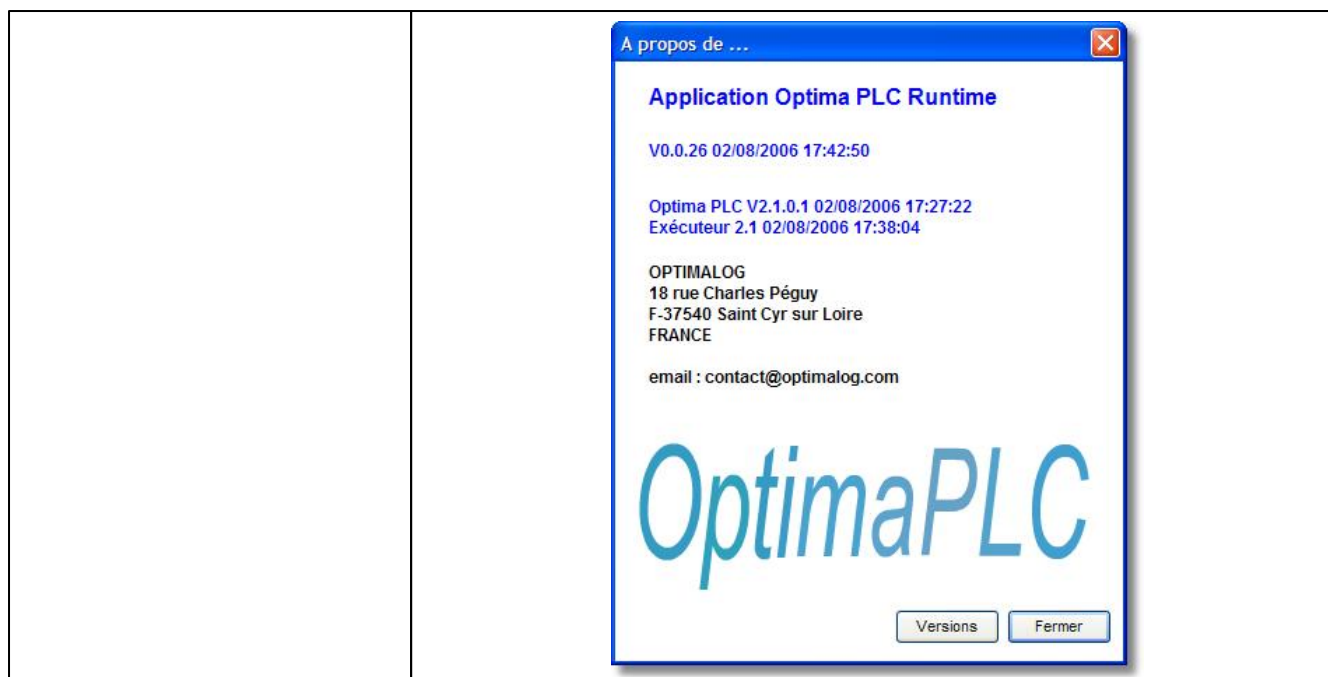


Le fichier ne contient que les données dont la valeur est non nulle, ceci afin de limiter l'encombrement du fichier et son temps de chargement.

4.3.6 Menu Aide

Option Aide	affiche ce fichier d'aide.
Option A propos	Affiche la page de version de l'application et du runtime Optima PLC.

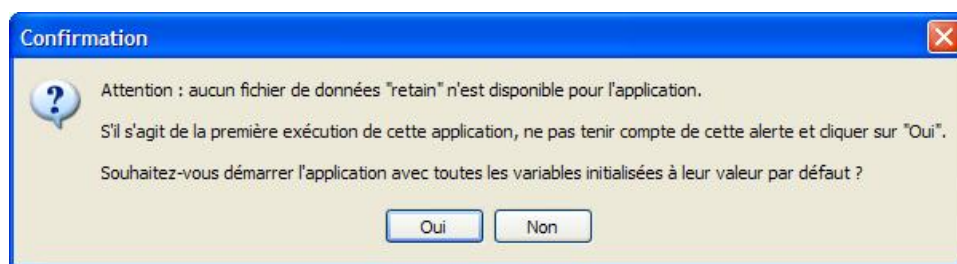




## 4.4 Problèmes possibles au démarrage du runtime

### 4.4.1 Pas de fichier de données "retain" au démarrage de l'application

Au démarrage, l'application charge un fichier de sauvegarde des variables retain. Si aucun fichier n'est disponible, le message suivant est affiché :



Comme le message l'indique, il s'agit d'une situation normale lors du premier démarrage de l'application. Les fichiers retain sont ensuite sauvegardés cycliquement et à la sortie de l'application, l'absence d'un tel fichier est alors une anomalie.

En cliquant sur Oui, les variables retain sont initialisées avec leur valeur par défaut et l'application est lancée. En cliquant sur Non, l'application est stoppée.

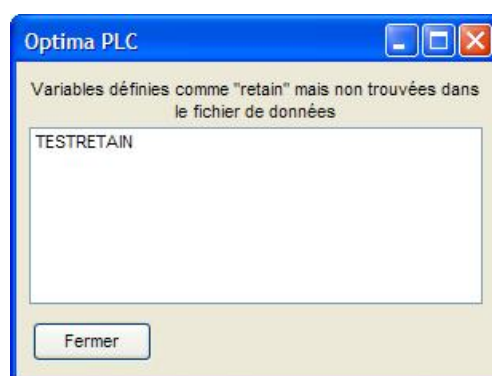
Les fichiers retain sont stockés dans le sous-répertoire "retain" du projet.

### 4.4.2 Modification de la structure des variables "retain"

Si un changement est intervenu dans la déclaration des variables retain depuis le dernier démarrage de l'application, celui-ci est détecté par le runtime. Une fenêtre comme celle qui suit est alors affichée :



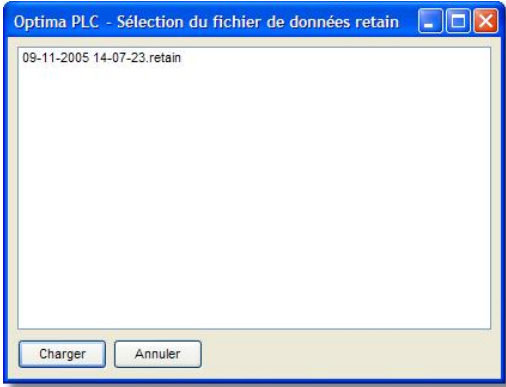
Dans cet exemple, une variable retain nommée "TestRetain" a été ajoutée. Elle n'a donc pas été sauvegardée dans le fichier retain précédent. En cliquant sur le bouton "liste" situé en face du libellé en rouge, la fenêtre suivante est affichée :



L'ensemble des variables concernée par une anomalie est mentionné dans la liste.

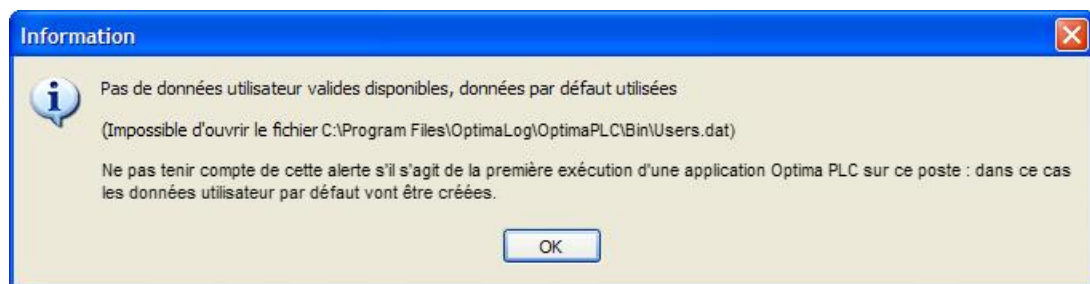
Outre la consultation des listes de variables, les options possibles de la première fenêtre sont les suivantes :

Arrêter l'application	L'application est immédiatement stoppée.
Démarrer sans charger les variables retain	Le fichier de variables retain est ignoré, les variables sont toutes initialisées à leur valeur par défaut.
Démarrer en chargeant uniquement les variables retain correctes	Les variables retain dont la valeur stockée dans le fichier est correcte sont initialisées avec celle-ci, les autres sont initialisées à leur valeur par défaut.
Charger un autre fichier de données	Cette option permet de choisir un autre fichier retain parmi ceux présents.

	 <p>La liste des fichiers disponibles est affichée. Pour en choisir un, placer le curseur dessus puis cliquer Charger.</p> <p>Le fichier est ensuite analysé, et les éventuelles anomalies sont signalées de la même manière que précédemment.</p> <p>Attention : les conséquences du chargement d'un fichier retain non à jour peuvent être importantes selon l'application.</p>
--	---

#### 4.4.3 Absence de données concernant les utilisateurs

Au démarrage, le runtime charge un fichier de données concernant les utilisateurs (program files\OptimaLog\OptimaPLC\bin\users.dat). Si ce fichier n'est pas trouvé, le message suivant est affiché :



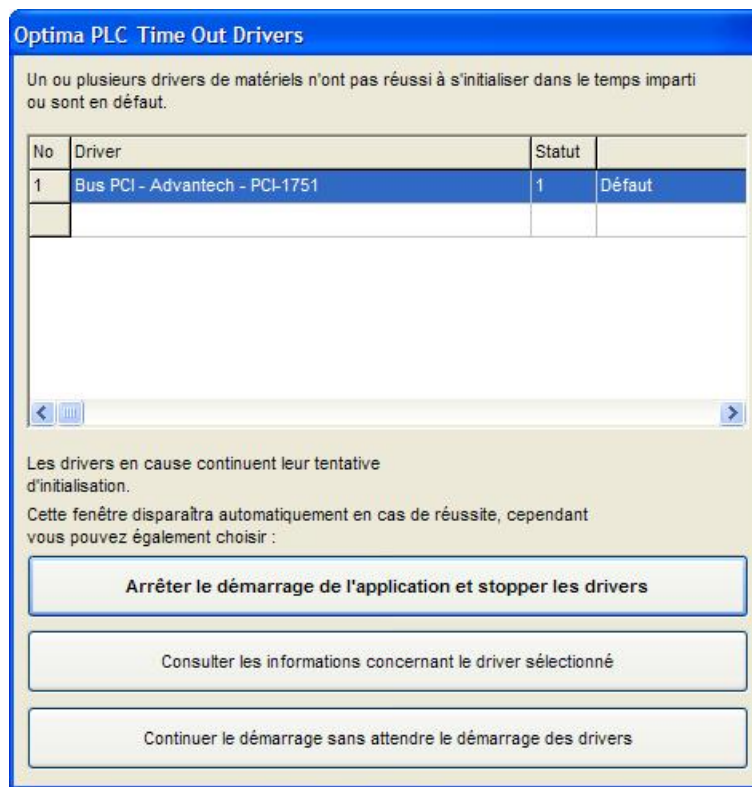
Cette situation est normale lors du premier lancement d'une application Optima PLC sur le PC, dans ce cas ne pas en tenir compte.

S'il ne s'agit pas du premier lancement d'une application, le fichier users.dat est détérioré ou inaccessible.

Après avoir cliqué sur OK, les données d'utilisateur par défaut sont restaurées, et un nouveau fichier users.dat est créé.

#### 4.4.4 Problème de démarrage d'un driver de matériel

Lors du démarrage de l'application, le runtime commence par initialiser les drivers. Au bout de 10 secondes, si un ou plusieurs drivers n'ont pas terminé leur initialisation, la fenêtre ci-dessous est affichée :



La liste des drivers posant problème est affichée. Cependant la tentative d'initialisation continue. Si celle-ci se termine, le driver disparaît de la liste. S'il était le seul de la liste, l'initialisation de l'application continue normalement.

Pendant le temps d'affichage de cette fenêtre, l'application se trouve en phase "[Start drivers timeout](#)<sup>329</sup>". Dans cette phase, aucune tâche automate n'est encore lancée.

L'utilisateur dispose de trois options :

Arrêter le démarrage	Les drivers déjà initialisés sont arrêtés, puis l'application est stoppée.
Consulter les informations concernant le driver sélectionné	La fenêtre de diagnostic concernant le driver est affichée. Les informations qu'elle contient dépendent de chaque driver.
Continuer le démarrage	Après une confirmation, l'application continue son démarrage et passe en mode "Start PLC" puis "Fault".

## 5 Gestion des utilisateurs

### 5.1 Présentation

Après la phase de conception, lorsque l'application et les outils de conception sont installés sur le matériel final, différents intervenants peuvent être amenés à utiliser les fonctions disponibles. Il est alors nécessaire de contrôler les accès aux logiciels afin de filtrer les autorisations en fonction de l'utilisateur.

Chaque logiciel intégré dans Optima PLC dispose d'une gestion avancée commune des utilisateurs.

Cette fonctionnalité peut être activée ou désactivée séparément dans chaque logiciel. Lorsqu'elle est activée, il devient nécessaire de s'identifier avant d'effectuer une action dans le logiciel.

Une liste d'utilisateurs commune aux différents logiciels est tenue à jour. Chaque utilisateur enregistré dispose d'un identifiant (ou "login"), d'un groupe d'appartenance qui définit ses droits, et d'un mot de passe. Lors de la connexion, l'utilisateur doit frapper son identifiant et son mot de passe. En fonction de son groupe, il a ensuite accès ou non aux différentes fonctions des logiciels.

Au bout d'un délai de cinq minutes sans action de l'utilisateur, la connexion est coupée et l'utilisateur doit à nouveau s'identifier. La connexion dans un des logiciels Optimalog entraîne la connexion automatique immédiate dans tous les autres (s'ils utilisent la gestion des utilisateurs) : de cette manière il n'est pas nécessaire de s'identifier chaque fois que l'on passe d'un logiciel à l'autre.

Les groupes d'utilisateurs sont définis de la manière suivante :

Groupe	Prérogatives
<b>Admins</b>	Groupe permettant l'ensemble des opérations possibles et permises dans le cadre de la licence installée. Le concepteur et installateur d'une application doit appartenir à ce groupe pour programmer et définir la configuration.
<b>Maintenance</b>	Groupe autorisé à la plupart des opérations d'édition et de configuration, hormis la programmation.
<b>Supervisors</b>	Groupe autorisé à visualiser et éditer les paramètres de fonctionnement d'une application.
<b>Operators</b>	Groupe dont les accès se limitent aux visualisations des paramètres et aux actions prévues par le concepteur.

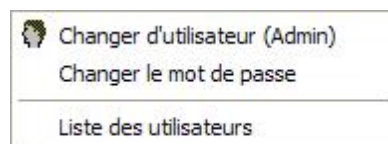
### Attention

L'utilisateur Admin (du groupe Admins) est créé par défaut avec le mot de passe Admin. Lorsque la gestion des utilisateurs est désactivée dans un logiciel, cet utilisateur est pris par défaut.

Il est **fortement recommandé** de modifier le mot de passe d'accès par défaut de l'utilisateur Admin.

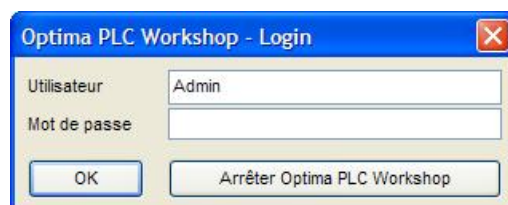
## 5.2 Menu Utilisateurs

Ce menu regroupe les options permettant de gérer la liste des utilisateurs autorisés. Il n'apparaît que si le mode "gestion des utilisateurs" a été activé dans les options (ces options peuvent varier d'un logiciel à l'autre).



### 5.2.1 Option Changer d'utilisateur

Cette option a pour effet immédiat de déconnecter l'utilisateur en cours, et provoque l'affichage de la fenêtre de connexion.



Il n'est plus possible d'utiliser le logiciel sans frapper un identifiant et un mot de passe corrects.

### 5.2.2 Option Changer le mot de passe

Cette option permet à l'utilisateur courant de modifier son mot de passe d'accès. Par sécurité, le mot de passe précédent doit être ressaisi, et le nouveau doit être saisi deux fois.

### 5.2.3 Option Liste des utilisateurs

Cette option provoque l'affichage de la liste des utilisateurs.

Nom	Groupe
Admin	Admins
Béatrice	Operators
Jacques	Supervisors
Lucie	Admins
Paul	Maintenance

Si l'utilisateur courant dispose de droits suffisants, il est possible de créer (bouton Nouveau), modifier ou supprimer des utilisateurs de la liste. Cependant, un utilisateur ne peut agir que sur les comptes qui lui sont subalternes (unique exception : un admin peut créer un autre admin, mais ne peut plus ensuite le modifier).

En cliquant sur Nouveau, la fenêtre de saisie des informations de l'utilisateur est affichée :

Tous les champs sont initialement vides. Le nom d'utilisateur ne peut être entré qu'à l'occasion de cette création, il ne peut plus être modifié par la suite.

## 6 Programmation

La programmation avec Optima PLC reprend les principes définis dans la norme IEC 61131-3. Des types de variables ainsi qu'un certain nombre de fonctions ont été ajoutées afin d'offrir des facilités supplémentaires.

D'une façon générale, un programme se compose de :

- déclarations de types
- déclarations de variables
- instructions (instructions simples, appels de fonctions, appels de POU...)
- commentaires (encadrés par "(" et ")" selon la norme, Optima PLC accepte également "/" en début de commentaire)

## 6.1 Types de données

### 6.1.1 Entiers

Les entiers définis dans la norme sont :

Type	Description	Taille en bits	Plage de valeurs	
<b>SINT</b>	Entier court	8	-128	127
<b>INT</b>	Entier	16	-32768	32767
<b>DINT</b>	Entier double	32	$-2^{31}$	$2^{31}-1$
<b>LINT</b>	Entier long	64	$-2^{63}$	$2^{63}-1$
<b>USINT</b>	Entier court non signé	8	0	255
<b>UINT</b>	Entier non signé	16	0	65535
<b>UDINT</b>	Entier double non signé	32	0	$2^{32}-1$
<b>ULINT</b>	Entier long non signé	64	0	$2^{64}-1$

Exceptionnellement, une variable peut être définie de type ANY\_INT, par exemple dans les entrées d'une fonction. La variable est alors compatible avec tous les types entiers. Elle est stockée comme une variable de type LINT.

Les valeurs entières peuvent être écrites sous un format décimal (exemple : +65, -57, 0...) ou dans une des bases suivantes :

- binaire : les valeurs sont alors précédées de 2# (exemple : 2#00110101)
- octal : les valeurs sont alors précédées de 8# (exemple : 8#020)
- hexadécimal : les valeurs sont alors précédées de 16# (exemple : 16#C2)

Pour plus de lisibilité, le caractère "\_" peut être éventuellement inséré au milieu des chiffres (exemple : +1\_000\_000).

### 6.1.2 Réels

Les réels définis dans la norme sont :

Type	Description	Taille en bits	Plage de valeurs
<b>REAL</b>	Nombre réel	32	$\pm 1.5 \times 10^{-45} \dots 3.4 \times 10^{38}$

<b>LREAL</b>	Nombre réel long	80 Ceci est une différence par rapport à la norme, qui définit pour ce type une taille de 64 bits	$\pm 3.6 \times 10^{-4951} \dots 1.1 \times 10^{4932}$
--------------	------------------	--	--

Exceptionnellement, une variable peut être définie de type ANY\_REAL. La variable est alors compatible avec tous les types réels. Elle est stockée sous la forme d'une variable de type LREAL.

Les valeurs s'écrivent avec un point pour séparer la valeur entière des décimales et le symbole pour l'exposant est "e" ou "E" (exemple : -45.68E7).

### 6.1.3 Durée

Le type utilisé pour les durées est le type TIME.

Les valeurs de temps sont normalisées avec les lettres :

- d : jours
- h : heures
- m : minutes
- s : secondes
- ms : millisecondes

Elles peuvent s'écrire suivant 2 formats :

- forme courte : précédée de T#, la valeur s'écrit sans discontinuité entre les nombres et les lettres (exemple : T#3d45m2s)
- forme longue : précédée de TIME#, la valeur s'écrit en séparant les binômes valeur/lettre par le signe "\_" (exemple : TIME#2d\_7h\_10m)

Les valeurs de type TIME sont stockées sur 64 bits avec une résolution de la microseconde. La durée maximale est d'environ 300000 années.

### 6.1.4 Horodatage

Les types date et temps définis dans la norme sont :

Type	Description	Taille en bits	Format
<b>DATE</b>	Date du calendrier	64	D#année-mois-jour ou DATE#année-mois-jour
<b>TIME_OF_DAY</b>	Heure de la journée	64	TOD#heure:minute:seconde ou TIME_OF_DAY#heure:minute:seconde
<b>DATE_AND_TIME</b>	Date et heure	64	DT#année-mois-jour-heure:minute:seconde ou DATE_AND_TIME#...

Les valeurs s'écrivent en précisant le type suivi de "#" puis en séparant les champs de la date par "-" et en séparant les champs de l'heure par ":" (exemple : le 10 mai 2004 à 11h09 s'écrit DT#2004-05-10-11:09:00).

Toutes les valeurs d'horodatage sont stockées sur 64 bits avec une résolution de la microseconde.



## 6.1.5 Chaîne de caractères

Le type dédié aux chaînes de caractères est le type "string".

La chaîne s'écrit entre 2 signes ' , et peut contenir les caractères de contrôle suivants :

Caractère de contrôle	Interprétation
<b>\$</b>	Utiliser le caractère \$ dans une chaîne
<b>'</b>	Utiliser le caractère ' dans une chaîne
<b>\$L ou \$l</b>	Saut de ligne (line feed)
<b>\$N ou \$n</b>	Nouvelle ligne
<b>\$P ou \$p</b>	Saut de page (form feed)
<b>\$R ou \$r</b>	Retour charriot (carriage return)
<b>\$T ou \$t</b>	Tabulation

Le type string occupe 256 bytes et peut contenir un maximum de 255 caractères. Le premier byte de la zone de données occupée est la "taille dynamique" de la chaîne, c'est à dire le nombre de caractères significatifs.

Une variable de type string se comporte également comme un tableau de CHAR, chaque élément du tableau, de 0 à 255, pouvant être affecté séparément.

Le type string peut être dérivé pour produire des chaînes de longueur plus courte.

### Exemple

la chaîne 'Prix d\$un logiciel :\$L\$R1000\$\$'  
 permet l'affichage suivant :      Prix d'un logiciel :  
    1000\$

## 6.1.6 Caractère

Le type caractère correspond à une valeur ASCII codée sur 8 bits.

Les caractères peuvent être spécifiés avec la même syntaxe que les chaînes de caractère de longueur 1.

## 6.1.7 Cordon de bits

Les entiers définis dans la norme sont :

Type	Description	Taille en bits	Valeurs
<b>BOOL</b>	booléen	1 (format binaire utilisé 8 bits)	0 (ou false) ou 1 (ou true)
<b>BYTE</b>	chaîne de 8 bits	8 bits	
<b>WORD</b>	chaîne de 16 bits	16 bits	
<b>DWORD</b>	chaîne de 32 bits	32 bits	
<b>LWORD</b>	chaîne de 64 bits	64 bits	

Comme pour les entiers, il est possible d'écrire les informations binaires sous forme décimale, binaire (2#) ou hexadécimale (16#).

Pour plus de lisibilité, le caractère "\_" peut être inséré au milieu des chiffres (exemple : 2#1000\_0100).

### 6.1.8 Types génériques

Les types génériques ne peuvent pas être utilisés pour déclarer des variables. Ces types sont mentionnés ici car ils sont utilisés dans la déclaration des blocs standard (exemple : le bloc AND peut être utilisé avec tous les cordons de bits).

Les types de données sont structurés suivant une hiérarchie qui permet de définir des types génériques qui regroupent des types semblables. Ces types sont :

ANY	
	ANY_NUM
	ANY_REAL
	LREAL, REAL
	ANY_INT
	LINT, DINT, INT, SINT
	ULINT, UDINT, UINT, USINT
	ANY_BIT
	LWORD, DWORD, WORD, BYTE, BOOL
	STRING
	ANY_DATE
	DATE_AND_TIME, DATE, TIME_OF_DAY
	TIME

### 6.1.9 Types dérivés

Il est possible de créer de nouveaux types de données dérivés des types prédéfinis.

La déclaration d'un nouveau type est encadrée par les mots clés "TYPE" et "END\_TYPE".

#### Exemple

```
TYPE
  TypeExemple : string;
END_TYPE
```

Les types peuvent être définis par la déclaration simple d'une variable ou en utilisant des formats de données plus structurés tels que :

- [les structures de données](#) <sup>360</sup>
- [les types énumérés](#) <sup>361</sup>
- [les plages de valeurs](#) <sup>361</sup>
- [les tableaux](#) <sup>361</sup>

#### 6.1.9.1 Structure de données

La structure de données est un ensemble de données de types différents. Sa déclaration est encadrée par les mots clés STRUCT et END\_STRUCT.

#### Exemple

```
TYPE StructureExemple :
  STRUCT
    Variable1deStructureExemple : DINT;
    Variable2deStructureExemple : DWORD;
    Variable3deStructureExemple : TIME;
  END_STRUCT
END_TYPE
```

```
END_STRUCT
END_TYPE
```

Si une variable STR est ensuite déclarée avec le type StructureExemple, il sera possible d'utiliser les expressions suivantes :

```
STR.Variable1deStructureExemple
STR.Variable2deStructureExemple
STR.Variable3deStructureExemple
```

### 6.1.9.2 Types énumérés

Il est possible d'affecter des noms à des valeurs (c'est-à-dire des états) pour un type de variable donné.

Les types énumérés permettent ainsi une meilleure lisibilité du programme ; d'autre part le contrôle de type effectué par le compilateur minimise le risque d'erreur, par rapport à l'utilisation de valeurs numériques.

Dans la déclaration du type, les noms des différents états sont mentionnés entre parenthèses et séparés les uns des autres par une virgule. Pour l'utilisation d'une valeur, il suffit d'écrire le nom du type suivi du signe # et de préciser l'état.

#### Exemple

```
TYPE
    ExempleTypeEnumere : (ARRET, MARCHE, DEFAULT);
END_TYPE
...
EtatSysteme:= ExempleTypeEnumere#ARRET;
```

### 6.1.9.3 Plages de valeurs

Il est possible de limiter les valeurs possibles d'une variable en définissant un type avec une plage de valeurs. Pour définir la plage, il suffit d'écrire entre parenthèses, après le type, les valeurs de début et de fin de plage séparées par 2 points consécutifs.

#### Exemple

```
TYPE
    ExemplePlageValeurs : INT (-4..+22);
END_TYPE
```

### 6.1.9.4 Tableaux de données

Les mots clés ARRAY et OF sont utilisés pour définir les tableaux de données. Les dimensions du tableau sont définies entre crochets. Chaque dimension est définie par un indice de début séparé par 2 points consécutifs de l'indice de fin. Le signe "," permet de séparer la définition de 2 dimensions. Le type des données du tableau est indiqué après le mot OF.

#### Exemple

```
TYPE
    ExempleTableau2D : ARRAY[1..5,2..10] OF DWORD;
END_TYPE
```

## 6.2 Variables prédéfinies dans Optima PLC

### 6.2.1 Tableaux de variables

Il s'agit de tableaux de valeurs de différents types, permettant de simplifier la programmation et la déclaration de données. Ces variables peuvent être directement consultées et même modifiées dans le runtime dans des pages spéciales.

Dans la plupart des cas, la taille des tableaux est définie par une valeur modifiable dans les options du projet (Constantes Système).

Nom	Indice de début	Indice de fin	Type	Usage	Accès direct
<b>R</b>	0	Défini dans Constantes Système	DINT	Registres numériques entiers	%MD0.x
<b>RR</b>	0	Défini dans Constantes Système	LREAL	Registres numérique réels	
<b>SA</b>	0	Défini dans Constantes Système	BOOL	Flags modifiables dans les tables par l'opérateur	%MX0.x
<b>L</b>	0	Défini dans Constantes Système	BOOL	Flags	%MX1.x
<b>C</b>	0	Défini dans Constantes Système	STRING	Chaînes de caractères	%MS0.x <i>voir note</i>
<b>DT</b>	0	Défini dans Constantes Système	TIME	Temporisations. La valeur de ces variables, une fois chargée avec un temps, est diminuée automatiquement au fil du temps jusqu'à atteindre 0.	%MT0.x <i>voir note</i>
<b>A</b>	0	Défini dans Constantes Système	BOOL	Flags d'alarmes (provoque l'enregistrement et l'apparition d'un message)	%MX2.x
<b>I</b>	0	Dépend des drivers installés	BOOL	Entrées booléenne	%IX0.x
<b>ANI</b>	0	Dépend des drivers installés	LREAL	Entrées analogiques	%IR0.x <i>voir note</i>
<b>O</b>	0	Dépend des drivers installés	BOOL	Sorties booléennes	%QX0.x
<b>ANO</b>	0	Dépend des drivers installés	LREAL	Sorties analogiques	%QR0.x <i>voir note</i>

*note* : cette syntaxe d'accès direct est propre à Optima PLC.

## Mnémoniques

Les variables prédéfinies peuvent être utilisées directement dans le programme sans faire l'objet de déclaration. Cependant, il est possible de leur associer un mnémonique et un texte descriptif. Le mnémonique peut être utilisé dans les blocs en lieu et place de la variable prédéfinie. Le texte descriptif est quant à lui affiché dans les fenêtres des variables prédéfinies du runtime.

Par exemple, voici la définition d'un mnémonique et d'un descriptif pour la variable R10 :

```
CounterBad AT R10 : dint;    // compteur de pièces mauvaises
```

Le mnémonique CounterBad et le texte descriptif "compteur de pièces mauvaises" (commentaire situé en fin de ligne de déclaration) sont maintenant associés au registre R10.

La déclaration suivante, en accord total avec la norme IEC61131, est aussi possible et donne le même résultat :

```
CounterBad AT %MD0.10 : dint;    // compteur de pièces mauvaises
```

Pour être correctement pris en compte, ce genre de déclaration doit se trouver dans une clause VAR..END\_VAR d'un bloc de variables globales. D'autre part, le type de variable précisé doit impérativement être celui de la variable prédéfinie.

## Tableaux des mnémoniques et des textes descriptifs

Les tableaux de chaînes de caractères "MnemoXX" et "TextXX" contiennent respectivement les mnémoniques et textes descriptifs des variables prédéfinies faisant l'objet d'une définition de mnémoniques.

De plus une variable "Text"mnémonique est également créée.

Dans le cas de l'exemple cité plus haut, les variables suivantes sont donc disponibles :

Variable	Contenu
MnemoR[10]	"CounterBad"
TextR[10]	"compteur de pièces mauvaises"
TextCounterBad	"compteur de pièces mauvaises"

## 6.2.2 Cas particuliers

### 6.2.2.1 Alarmes A

Les alarmes A sont des variables booléennes prédéfinies dans un tableau. Leur intérêt est de permettre une gestion simple de messages de défauts destinés à l'opérateur (une autre gestion plus élaborée est proposée avec le logiciel **Optim'Alarm**).

Lorsque l'une des alarmes est mise à 1 ou à 0, une ligne est ajoutée dans le journal de bord de l'application. De plus, lors du passage à 1, une fenêtre d'information contenant les alarmes actives peut être affichée (voir les options du runtime).

Pour être pleinement exploitées, les alarmes doivent être utilisées avec déclaration de mnémonique et de commentaire :

#### Exemple de déclaration

```
FaultAirPression at A5 : bool; // Défaut de pression d'air
```

Dans cet exemple, la mise à 1 de FaultAirPression provoquera l'affichage de "Défaut de pression d'air" dans la fenêtre des alarmes actives. Ce message sera également mentionné dans le journal de bord.

### 6.2.2.2 Temporisations DT

Les temporisations DT sont des variables de type Time prédéfinies dans un tableau. Leur particularité est que leur valeur, si elle n'est pas nulle, est automatiquement diminuée du temps écoulé jusqu'à atteindre la valeur 0.

Cette possibilité rend particulièrement simple la programmation de temporisations. Il suffit en effet de charger une variable DT avec le temps souhaité, puis de tester son passage à 0. Il est possible également de tester une valeur intermédiaire.

## 6.2.3 Variables diverses

Variables horodatrices (toutes de type DINT) :

<b>Day</b>	quantième du mois (1 à 31)
<b>Month</b>	numéro du mois (1 à 12)
<b>Year</b>	année en cours
<b>Hours</b>	heure de la journée
<b>Minutes</b>	minutes dans l'heure courante
<b>Seconds</b>	secondes dans l'heure courante
<b>DayOfYear</b>	numéro du jour dans l'année
<b>WeekNb</b>	numéro de la semaine dans l'année

## 6.3 Déclarations

### 6.3.1 Déclarations de variables

La déclaration de variables s'effectue entre 2 mots clés, le premier définit le type de déclaration, le second END\_VAR clôture la partie déclaration.

La déclaration se compose du nom de la variable suivi de ":" puis du type de variable et termine par ";".

La syntaxe utilisée ici est celle de la norme IEC 61131-3.

#### Exemple

```
VAR
    Variable1 : DINT;
END_VAR
```

Les variables peuvent être :

- globales : les variables sont accessibles depuis tous les POU situés après la déclaration. Dans Optima PLC, il existe un POU spécifique pour ce type de variable qui permet de déclarer en global sans avoir à redéfinir à nouveau la variable comme étant une variable externe (comme la norme le prévoit) au moment de son utilisation.
- locales : les variables sont utilisables uniquement dans le POU où elles ont été déclarées (onglet "Déclarations" dans Optima PLC)

Les variables sont ensuite classées en fonction de leur type d'utilisation :

<b>VAR</b>	Fonction, bloc de fonction, bloc de programme : les variables sont locales (visibles uniquement dans le POU) et sont sauvegardées entre 2 appels du POU (variables statiques).
	POU de variables globales : les variables sont globales et accessibles depuis les autres POU déclarés après.
<b>VAR_TEMP</b>	les variables sont temporaires, leur valeur n'est pas sauvegardée entre 2 appels du POU.
<b>VAR_INPUT</b>	les variables d'entrée sont les paramètres du POU, elles doivent être affectées avant ou lors de l'appel du POU, mais ne peuvent pas être modifiées dans le POU
<b>VAR_OUTPUT</b>	les variables de sortie sont les résultats d'exécution du POU, elles ne peuvent être affectées que par le POU
<b>VAR_IN_OUT</b>	les variables d'entrée-sortie sont des paramètres qui peuvent être modifiés dans le POU.
<b>VAR_GLOBAL</b>	les variables sont globales (autorisé uniquement dans un bloc de programme). Cette syntaxe est celle de la norme, mais l'utilisation de variables globales avec un POU de variables est plus pratique.
<b>VAR_EXTERNAL</b>	les variables sont locales au POU mais sont liées à des variables globales définies dans un bloc de programme avec VAR_GLOBAL.
<b>Accès direct en mémoire</b>	la nom de la variable commence par % et est suivi de 2 lettres dont l'interprétation est pour la première le type de zone mémoire, pour la seconde la longueur des données (voir tableau ci-dessous). Une valeur d'indice est ensuite ajoutée à ces 3 caractères.  <b>Exemple</b> : %MB100 : la variable est le 100ème octet en mémoire interne

Première lettre	Interprétation	Deuxième lettre	Interprétation
<b>I</b>	Emplacement mémoire en	<b>X</b>	Bit

	entrée		
<b>Q</b>	Emplacement mémoire en sortie	<b>B</b>	Octet (8 bits)
<b>M</b>	Mémoire interne	<b>W</b>	Mot (16 bits)
		<b>D</b>	Mot double (32 bits)
		<b>L</b>	Mot long (64 bits)

Codage pour accès direct en mémoire

### Cas des fonctions

Dans le cas d'une fonction, les possibilités de déclaration sont réduites. En particulier, la déclaration VAR (variables statiques) n'est pas autorisée.

## 6.3.2 Attributs de variables

Les variables peuvent avoir des attributs additionnels définis par les mots clés suivants :

- **RETAIN** : indique que la valeur des variables doit être conservées lorsqu'il y a disparition de la puissance du PLC (permet une sauvegarde en cas de coupure de courant)

### Exemple

```
VAR RETAIN
    Statistique1 : LINT;
END_VAR
```

- **CONSTANT** : la variable a une valeur constante pendant l'exécution du programme, cette valeur est spécifiée après le type grâce au signe " := " :

### Exemple

```
VAR CONSTANT
    NombrePerceusesDansSysteme : SINT := 4;
END_VAR
```

- **AT** : pour donner un nom à une variable en spécifiant son emplacement mémoire ; dans Optima PLC il est ainsi possible d'utiliser ce mot clé pour affecter un mnémonique à une variable automate.

### Exemple

```
VAR
    NumeroMachine    AT    %MB1;
    ArretMachine      AT    SA0;
END_VAR
```

## 6.3.3 Instances de POU

Pour pouvoir utiliser un bloc de fonction depuis un autre POU, il faut en créer une instance dans la partie déclaration. Cette déclaration est tout à fait similaire la déclaration d'une variable quelconque.

Attention, l'instance d'un bloc ne doit pas être définie dans les variables temporaires (VAR\_TEMP)

**Exemple** : le programme Machine1 utilise une instance du bloc de fonction MiseSousPuissance

```
VAR
```

```
MiseSousPuissanceMachine1 : MiseSousPuissance;
END_VAR
```

**Remarque** : le fait de déclarer une instance de POU ne signifie pas que les instructions de ce POU seront exécutées ; pour exécuter cette partie de programme, l'appel de l'instance "MiseSousPuissanceMachine1" doit faire partie des instructions du programme.

## 6.4 Valeurs constantes

### 6.4.1 Constantes numériques

Description	Exemples
Libellés entiers	-12 0 123_456 +986
Libellés réels	-12.0 0.0 0.4560 3.14159_26
Libellés réels avec exposants	-1.34E-12 ou -1.34e-12 1.0E+6 ou 1.0e+6 1.234E6 ou 1.234e6
Libellés en base 2	2#1111_1111 (255 décimal) 2#1110_0000 (240 décimal)
Libellés en base 8	8#377 (255 décimal) 8#340 (240 décimal)
Libellés en base 16	16#FF ou 16#ff (255 décimal) 16#E0 ou 16#e0 (240 décimal)
"Zéro" et "un" booléens	0 1
VRAI ou FAUX booléens	FALSE TRUE

Important : les valeurs entières en base 10 sont compatibles avec les types numériques (SINT, USINT, INT, ...), alors que les valeurs entières en base 2, 8 ou 16 sont considérées comme des cordons de bits. Pour outrepasser cette restriction, utiliser des constantes typées (voir plus loin) :

```
INT#16#12    valeur entière codée en hexadécimal.
WORD#100     cordon de bits codé en décimal
```

### 6.4.2 Constantes chaînes de caractères

Une chaîne de caractères est une séquence de 0 à n caractères, commençant et se terminant par une apostrophe (').

La séquence du signe \$ suivi de deux chiffres hexadécimaux correspond à la représentation hexadécimale d'un caractère.

Exemple	Explication
"	Chaîne vide
'A'	Chaîne de longueur 1 contenant le caractère A
' '	Chaîne de longueur 1 contenant le caractère "espace"
'\$'	Chaîne de longueur 1 contenant le caractère "apostrophe".
'\$R\$L' '\$0D\$0A'	Chaînes de longueur 2 contenant les caractères CR et LF.
'\$\$1.00'	Chaîne de longueur 5 correspondant à \$1.00



Les séquences constituées du caractère \$ suivi d'un caractère spécial sont décrites ci-dessous.

Combination	Interpretation
\$\$	Caractère Dollar
\$'	Apostrophe
\$L ou \$l	LF (changement de ligne)
\$N ou \$n	Changement de ligne
\$P ou \$p	FF (changement de page)
\$R ou \$r	CR (retour curseur)
\$T ou \$t	Tabulation

### 6.4.3 Constantes de durée

Description	Exemples
Libellés de durée sans soulignement Préfixe court	T#14ms T#-14ms T#14.7s T#14.7m T#14.7h t#14.7d t#25h15m t#5d14h12m18s3.5ms
Préfixe long	TIME#14ms TIME#-14ms time#14.7s
Libellés de durée avec soulignement Préfixe court	t#25h_15m t#5d_14h_12m_18s_3.5ms
Préfixe long	TIME#25h_15m time#5d_14h_12m_18s_3.5ms

### 6.4.4 Constantes de date

Préfixes pour les libellés :

Description	Préfixe
Libellés de date Préfixe long	DATE#
Libellés de date Préfixe court	D#
Libellés d'heure du jour Préfixe long	TIME_OF_DAY#
Libellés d'heure du jour Préfixe court	TOD#
Libellés de date et d'heure du jour Préfixe long	DATE_AND_TIME#
Libellés de date et d'heure du jour Préfixe court	DT#

#### Exemples

Préfixe long	Préfixe court
DATE#1984-06-25 date#1984-06-25	D#1984-06-25 d#1984-06-25
TIME_OF_DAY#15:36:55.36 time_of_day#15:36:55.36	TOD#15:36:55.36 tod#15:36:55.36
DATE_AND_TIME#1984-06-25-15:36:55.36 date_and_time#1984-06-25-15:36:55.36	DT#1984-06-25-15:36:55.36 dt#1984-06-25-15:36:55.36

## 6.4.5 Valeurs constantes typées

L'utilisation de valeurs constantes présente parfois une difficulté concernant leur type, celui-ci n'étant pas explicite.

Par exemple, la valeur constante 10 est compatible avec les types suivants : SINT, USINT, UINT, INT, UDINT, DINT, LINT, ULINT. La valeur constante 1 est en plus compatible avec le type BOOL.

Ces valeurs numériques n'ont donc pas de type précis lors de leur évaluation, le compilateur autorisant cette incertitude le plus longtemps possible.

Il est cependant possible de forcer le type de toute valeur constante en préfixant la constante avec le type désiré suivi de #, de la même manière que pour les valeurs TIME, DATE, etc.

### Exemples

INT#15	valeur entière 15
INT#16#12	valeur entière 18
BYTE#10	cordon de bits 2#1010
BOOL#1	valeur booléenne true

## 6.5 Initialisation de variables

Les variables peuvent être initialisées de 2 façons :

- lors de la déclaration du type ou de la variable

La valeur précisée ici correspond à la valeur initialisée lors d'un redémarrage de l'application si la variable n'est pas "RETAIN". Si la variable est "RETAIN", la valeur n'est utilisée pour l'initialisation que lorsque aucune donnée retain la concernant n'est disponible.

- Dans une tâche exécutée en phase START, permettant des initialisation avant le passage en RUN.

### 6.5.1 Initialisation des types et variables simples

Un type ou une variable peut être initialisé au moment de sa déclaration.

La valeur est alors affectée grâce au signe ":", et est placée à la fin de la ligne de déclaration.

#### Exemple

```
TYPE
  VarInitialisee      :      DINT      := 22;
END_TYPE
```

#### Cas particulier : initialisation des tableaux

Pour initialiser un tableau, les valeurs doivent être séparées par des virgules ; lorsque plusieurs éléments consécutifs reçoivent la même valeur, il suffit de préciser le nombre d'éléments à initialiser et d'indiquer entre parenthèses la valeur à affecter.

**Exemple** : le tableau est initialisé avec les valeurs 1 pour le 1er élément, 0 pour les 3 suivants et 10 pour le dernier élément

```
VAR
  TableauStatistiques : ARRAY[0..5] := 1 , 3(0), 10;
END_VAR
```

### 6.5.2 Initialisation de types et variables multi-éléments

Il est possible d'initialiser des variables de type multi-éléments.

#### Initialisation des tableaux

Pour initialiser un tableau, les valeurs doivent être séparées par des virgules ; lorsque plusieurs éléments consécutifs reçoivent la même valeur, il suffit de préciser le nombre d'éléments à initialiser et d'indiquer entre parenthèses la valeur à affecter.

**Exemple** : le tableau est initialisé avec les valeurs 1 pour le 1er élément, 0 pour les 3 suivants et 10 pour le dernier élément

```
VAR
    TableauStatistiques : ARRAY[0..5] := 1 , 3(0), 10;
END_VAR
```

Tous les éléments ne doivent pas obligatoirement être initialisés.

### Initialisation de structure

Chaque champ de la structure peut être initialisé. Lors de la déclaration, après avoir mentionné le type de la variable, les différents champs sont mentionnés entre parenthèses avec les valeurs affectées à chacun d'eux. Tous les champs ne doivent pas obligatoirement être initialisés.

### Exemple

```
TYPE Limites
    STRUCT
        Minimum : DINT;
        Maximum : DINT;
        Moyenne : DINT;
    END_STRUCT
END_TYPE

VAR
    LimitesTension : Limites(Minimum:= 0, Maximum:= 10);
END_VAR
```

## 6.6 Programmation en texte structuré (ST)

Le langage ST est un langage évolué structuré, proche du langage Pascal, permettant de définir des traitements et des algorithmes complexes.

### 6.6.1 Expressions

Une expression est une construction syntaxique, destinée à être évaluée, afin de fournir un résultat correspondant à l'un des types de données définis.

Les expressions sont constituées d'opérateurs et d'opérandes.

Les opérateurs sont résumés dans le tableau ci-dessous.

	Opération	Symbole	Priorité
<b>1</b>	Parenthésage	(expression)	Maximale
<b>2</b>	Evaluation de fonction	identifiant (argument list)  exemple : LN(A), MAX(X,Y), etc.	
<b>3</b>	Exponentiation	**	
<b>4</b>	Négation	-	
<b>5</b>	Complément	NOT	
<b>6</b>	Multiplication	*	
<b>7</b>	Division	/	
<b>8</b>	Modulo	MOD	

<b>9</b>	Addition	+	
<b>10</b>	Soustraction	-	
<b>11</b>	Comparaison	< , > , <= , >=	
<b>12</b>	Egalité	=	
<b>13</b>	Inégalité	<>	
<b>14</b>	ET Booléen	&	
<b>15</b>	ET Booléen	AND	
<b>16</b>	OU exclusif Booléen	XOR	
<b>17</b>	OU booléen	OR	Minimale

L'évaluation d'une expression consiste à appliquer les opérateurs aux opérandes dans l'ordre défini par la priorité.

Par exemple, si A, B, C et D sont du type Int et ont les valeurs 1, 2, 3 et 4, alors

B-C\*ABS(D)                      prend la valeur 9  
 (A+B-C)\*ABS(D)                prend la valeur 0

## 6.6.2 Instructions

Résumé des types d'instruction.

	Instructions	Exemples
<b>1</b>	Affectation	A := B; CV := CV+1; C := SIN(X);
<b>2</b>	Appel d'un bloc de fonction	CMD_TMR ( IN:=%IX5, PT:=T#300ms); A := CMD_TMR.Q;
<b>3</b>	RETURN Sortie d'une fonction ou d'un bloc de fonction	RETURN;
<b>4</b>	IF	D := B*B - 4*A*C; IF D < 0.0 THEN NROOTS := 0; ELSIF D = 0.0 THEN NROOTS := 1; X1 := - B/(2.0*A); ELSE NROOTS := 2; X1 := (- B + SQRT (D))/(2.0*A); X2 := (- B - SQRT (D))/(2.0*A); END_IF;
<b>5</b>	CASE Sélection parmi plusieurs valeurs	TW := BCD_TO_INT(THUMBWHEEL); TW_ERROR := 0; CASE TW OF 1,5 : DISPLAY := OVEN_TEMP; 2 : DISPLAY := MOTOR_SPEED; 3 : DISPLAY := GROSS - TARE; 4,6..10: DISPLAY := STATUS (TW - 4); ELSE DISPLAY := 0;  TW_ERROR := 1; END_CASE;

		QW100 := INT_TO_BCD(DISPLAY);
6	FOR Répétition d'une boucle n fois.	J := 101; FOR I := 1 TO 100 BY 2 DO IF WORDS[I] = 'KEY' THEN J := I; EXIT; END_IF; END_FOR;
7	WHILE Répétition tant que la condition est vraie.	J := 1; WHILE J <= 100 & WORDS[J] <> 'KEY' DO J := J+2; END_WHILE;
8	REPEAT Répétition jusqu'à ce que la condition soit vraie.	J := -1; REPEAT J := J+2; UNTIL J = 101 OR WORDS[J] = 'KEY' END_REPEAT;
9	EXIT Permet de sortir d'une boucle itérative avant que la condition terminale soit satisfaite.	EXIT;
10	Instruction vide	;

### 6.6.2.1 Affectation

L'affectation permet de remplacer la valeur d'une variable par le résultat de l'évaluation d'une expression.

La partie gauche d'une affectation est une variable, celle de droite est une expression dont le type du résultat correspond à celui de la variable.

#### Exemple

```
A:= B + C;
```

### 6.6.2.2 Appel de fonction

L'appel d'une fonction est effectué comme partie de l'évaluation d'une expression.

#### Exemple

La fonction Func a deux paramètres entiers, et renvoie un réel.

```
X= Func( A, B ) * 1.5;
```

## 6.7 Programmation en liste d'instructions (IL)

Un programme IL est constitué d'une liste d'instructions élémentaires, proches de l'assembleur.

### 6.7.1 Opérateurs, modificateurs, opérandes

Chaque instruction doit commencer sur une nouvelle ligne. Elle est constituée d'un opérateur, éventuellement d'un ou plusieurs modificateurs, et d'un opérande.

Une instruction peut être précédée d'une "étiquette" suivie de ":", et suivie d'un commentaire.

Etiquette	Operateur (suivi éventuellement d'un ou	Opérande	Commentaire
-----------	---	----------	-------------

	plusieurs modificateurs)		
START:	LD	I100	( * PUSH BUTTON * )
	ANDN	SA100	( * NOT INHIBITED * )
	ST	O80	( * FAN ON * )

Remarque : lorsqu'une étiquette est définie, il ne doit pas y avoir d'espace entre l'étiquette et le symbole ":".

Les instructions manipulent l'"accumulateur", ou "résultat courant". Il s'agit d'une variable pouvant prendre les différents types de données. Ce résultat peut être mis provisoirement de côté pendant l'évaluation d'un autre résultat, ce qui conduit à la notion classique de "pile".

Le fonctionnement général d'une instruction IL, nommée par exemple OP, est le suivant :

Résultat:= Résultat OP opérande

Par exemple, l'instruction :

AND SA10

doit être interprétée comme :

Résultat:= Résultat AND SA10

	Opérateur	Modificateurs	Opérande	Signification
1	LD	N		Donne au résultat courant le type et la valeur de l'opérande
2	ST	N		Range la valeur du résultat courant dans l'opérande. Le résultat courant et l'opérande doivent avoir le même type.
3	S	Note 3	BOOL	Si le résultat courant vaut 1, force l'opérande à 1.
	R	Note 3	BOOL	Si le résultat courant vaut 1, force l'opérande à 0.
4	AND	N, (	BOOL	ET booléen
5	&	N, (	BOOL	ET booléen
6	OR	N, (	BOOL	OU booléen
7	XOR	N, (	BOOL	OU exclusif booléen
8	ADD	(		Addition
9	SUB	(		Soustraction
10	MUL	(		Multiplication
11	DIV	(		Division
12	GT	(		Comparaison >
13	GE	(		Comparaison >=
14	EQ	(		Comparaison =
15	NE	(		Comparaison <>
16	LE	(		Comparaison <=
17	LT	(		Comparaison <
18	JMP	C, N	Etiquette	Saut à l'instruction qui suit l'étiquette
19	CAL	C, N	Nom	Appel d'un bloc de

				fonction
20	RET	C, N		Fin d'exécution du bloc
21	)			Evaluation d'une opération différée

Un modificateur est un élément qui vient modifier l'opérateur, il se place à la fin de l'opérateur (ne pas mettre d'espace entre les deux). Exemple : ORN.

Le modificateur N provoque une négation de l'opérande.

Le modificateur C indique que l'instruction ne doit être effectuée que si le résultat courant est égal à 1.

Exemple :

```
JMPC Etiquette1
```

Cette instruction provoque le saut à l'instruction qui suit Etiquette1 seulement si le résultat courant est égal à 1.

Enfin, le modificateur ( permet de différer l'opération. Par exemple :

```
LD R10
ADD( R11
MUL 10
)
```

Doit être interprété comme :

```
Résultat:= R10
Résultat:= Résultat + (R11 * 10)
```

Les modificateurs ( peuvent être imbriqués plusieurs fois.

L'opérateur ) doit apparaître autant de fois que le modificateur (.

Exemple :

```
LD R5
ADD( R10
MUL R11
DIV( R12
ADD R13
)
)
```

## 6.7.2 Appel de fonctions

Les fonctions doivent être lancées en inscrivant le nom de la fonction dans le champ opérateur. Le résultat de la fonction est placé dans le résultat courant.

Les fonctions peuvent être appelées de façon formelle ou informelle.

### Appel formel :

Le nom de la fonction est suivi du modificateur (, viennent ensuite les affectations des paramètres d'entrée (une affectation par ligne, écrite au format ST sans ; à la fin), la fin des affectations est signalée par l'opérateur ).

Exemple :

Soit une fonction CALCUL comportant 2 paramètres d'entrée Valeur1 et Valeur2.  
L'appel formel de cette fonction est le suivant :

```
CALCUL(
Valeur1:= 10
```

```
Valeur2:= 5
)
```

### Appel informel :

Le résultat courant est utilisé comme premier paramètre de la fonction. Le nom de la fonction est éventuellement suivi d'un second paramètre.

Exemples :

```
LD 30.2
COS          (* calcul du cosinus dans le résultat courant *)
LD 10
CALCUL 5     (* équivalent à l'exemple décrit dans la partie Appel formel *)
```

## 6.7.3 Appel de blocs de fonction

Un appel de bloc de fonction peut se faire sous différentes formes, comme décrit dans le tableau suivant.

La déclaration suivante est supposée faite dans ces exemples :

```
VAR
    C10: CTU;
END_VAR
```

	Description	Instructions
1	CAL avec liste d'entrées	CAL C10(CU:=%IX10, PV:=15)
2	CAL avec instruction LD et ST pour les entrées	LD 15 ST C10.PV LD %IX10 ST C10.CU CAL C10
3	Utilisation d'opérateurs d'entrées (voir tableau suivant)	LD 15 PV C10 LD %IX10 CU C10

Opérateurs d'entrées pour les blocs de fonctions standards :

	Operateurs	Type de bloc de fonction
4	S1, R	SR
5	S, R1	RS
6	CLK	R_TRIG
7	CLK	F_TRIG
8	CU, R, PV	CTU
9	CD, LD, PV	CTD
10	CU, CD, R, LD, PV	CTUD
11	IN, PT	TP
12	IN, PT	TON
13	IN, PT	TOF

Ces opérateurs effectuent le chargement de l'entrée de bloc de même nom avec la valeur du résultat courant, qui doit être de même type. Le bloc est ensuite appelé.

Exemple :

```
LD 100000
PV CTD1 (* Chargement du décompteur avec la valeur 100000 *)
LD 1
```



## 6.8 Programmation en langage Apigraf

Le langage Apigraf est un langage dédié à la programmation automate, basée sur des conditions booléennes textuelles simples.

### 6.8.1 Instruction booléenne type

Une instruction booléenne Apigraf peut se décomposer en trois colonnes :

Condition	Action	Commentaire
-----------	--------	-------------

La largeur de chaque zone d'écriture n'a pas d'importance particulière. Seules deux règles doivent être respectées : les conditions doivent commencer en début de ligne, à l'inverse des actions qui ne doivent pas commencer en début de ligne.

#### Exemple

```
|Marchel & \ArretUrgence      Moteur1      ! démarrage Moteur 1
|SINON                        \Moteur1      ! arrêt moteur 1
|                               ! (la partie SINON est
|                               ! facultative)
```

#### Explication

- **Condition**

expression booléenne écrite obligatoirement en début de ligne.

- **Action**

si la condition est vraie, les actions précisées dans cette partie sont exécutées. La partie "ACTION" peut comporter plusieurs lignes, chaque ligne contenant une seule action.

Une ACTION doit être précédée d'au moins un blanc ou d'une tabulation. En effet, au contraire des conditions, les actions ne doivent pas commencer en début de ligne.

Pour les ACTIONS, on parle aussi de "PROCESS".

Attention : dans les langages de la norme IEC 61131-3, le mot "action" se rapporte à un concept un peu différent de celui d'Apigraf : il s'agit ici d'une instruction quelconque effectuant quelque chose en réponse à une condition.

- **Commentaire**

Ils sont toujours précédés du signe "!".

Dans l'exemple précédent, la condition :

```
Marchel & \ArretUrgence
```

est l'expression booléenne conditionnant l'action à effectuer. Si celle-ci est vraie (à 1), la sortie "Moteur1" est activée, sinon elle est désactivée (instruction "\Moteur1").

La partie "SINON" est facultative dans une condition. Le mot sinon peut être placé ou non en début de ligne.

#### Exemple

```
|Marchel & \ArretUrgence
|                               Moteur1
|                               Lampel
|Sinon                        \Moteur1      ! cette partie est
```

| \Lampel ! facultative

## 6.8.2 Condition (expression booléenne)

Une condition, en APIGRAF, est une expression booléenne dont le résultat est vrai (1) ou faux (0). Ce résultat découle d'opérations logiques sur des valeurs booléennes ou numériques.

Les parenthèses "(" et ")" permettent d'encadrer les conditions, de manière à obtenir leur évaluation correcte. Les opérateurs booléens sont les suivants :

<b>&amp;</b>	ET logique
<b>#</b>	OU logique
<b>\</b>	Négation logique
<b>XOR, &lt;&gt;</b>	OU exclusif
<b>=</b>	égalité de deux valeurs booléennes

Les conditions formées de valeurs booléennes combinées par ces opérateurs sont correctes. signification logique des signes = (égal) et <> (différent)

### • Egalité

Les deux écritures suivantes sont équivalentes :

```
| (L1 = L2)                action1
| (L1 & L2) # (\L1 & \L2)   action1
```

### • Différence

Les deux écritures suivantes sont équivalentes :

```
| (L1 <> L2)                action1
| (L1 & \L2) # (\L1 & L2)   action1
```

### • Exemples de conditions

```
| L1                ! vrai si L1 vrai
| L1 <> A2           ! vrai si (L1 et non A2) ou (non L1 et A2)
| L1 & \SA2 & (L3 = L4) ! vrai si L1 et non SA2 et
|                  ! ((L3 et L4) ou (non L3 et non L4))
```

Dans les conditions ci-dessus, les valeurs booléennes sont des variables booléennes, comme les flags L, les alarmes A, les SA, ou les E/S : c'est le cas le plus simple.

### • Test sur valeurs numériques

Cependant, les valeurs booléennes peuvent être aussi le résultat de comparaison sur des valeurs numériques :

```
| R10 > 500                action1
```

action1 sera activée si R10 est supérieur à 500

```
| L1 & (R10 > 1000) & (R1 + R2 - R3 = 0)
```

Ici, le résultat de la comparaison de R10 avec 1000 donne une valeur booléenne directement utilisable dans la condition. Des calculs numériques intermédiaires peuvent aussi être effectués, comme ici  $R1 + R2 - R3 = 0$  le résultat du calcul est comparé à "0" (voir les opérateurs numériques ci-après).

*Les conditions doivent toujours commencer en début de ligne.*

*Les conditions portant sur des variables numériques doivent être parenthésées si elles sont combinées avec d'autres conditions.*

## 6.8.3 Exemple de conditions et d'actions

### Exemples de conditions correctes

```
|DT6 = 5000           ! vraie si DT6 est égale à 5000
|R4 * 2 = R7 div 3    ! vraie si 2 fois R4 est égale à R7 divisé par 3
|Timer <= 25000       ! vraie si Timer est plus petit ou égal à 25000
|(I5 # (DT5 = 0) # \SA5) & (Timer < R6) & \I7
```

#### Rappel

L'expression booléenne définissant une condition doit *obligatoirement* commencer en début de ligne. Une comparaison numérique doit être placée entre parenthèses si elle est combinée à une autre par un opérateur :

```
|R5 > 500             ! correct
|R5 > 500 & SA2       ! incorrect
|(R5 > 500) & SA2     ! correct
```

### Exemples d'actions correctes

#### Rappels

Le "process" est l'ensemble des actions qui seront activées ou non en fonction de la condition qui précède. Une instruction du process est toujours précédée d'espaces ou d'une tabulation, qui indiquent qu'il s'agit d'une action à effectuer (à la différence des conditions qui elles, commencent en début de ligne).

#### Actions sur variables booléennes

```
|      O5             ! active la sortie 5
|      \HL3           ! désactive la sortie HL3
|      \O5,8          ! désactive les sorties 5 et 8
|      O7 = SA4 & (I3 # I4) ! O7 est vrai si
|                      ! l'équation booléenne à droite
|                      ! du signe est vraie
|      L4             ! active le flag logique
|                      ! L4 à "vrai"
|      \SA6           ! désactive l'interrupteur logique SA6
|      SA5,6,7,20..30 ! active les interrupteurs logiques
|                      ! 5,6,7 et 20 à 30
|      L3 = \L3       ! inverse l'état de L3
```

Ce qui précède est valable pour la modification de toute variable booléenne du type A, O, I, L, SA, mnémoniques d'E/S, ou autre.

De façon générale, si X et Y sont deux variables booléennes, l'écriture des actions suivantes :

```
|      X              ! passage de X à 1
|      \X             ! passage de X à 0
|      X = Y          ! X prend l'état de Y
|      X = \Y         ! X prend l'état inverse de Y
|      X = \X         ! l'état de X est inversé (\X
|                      ! est placé dans X)
```

Dans une affectation de valeurs booléennes avec le signe =, le premier membre est une variable, et le second est une expression, qui peut avoir la syntaxe des expressions décrites plus haut. Par exemple, l'action suivante est correcte :

```
|      X = L1 & L2 & (R5 > 1000)
```

Enfin, les variables indicées prédéfinies ont la particularité de pouvoir être modifiées globalement :

```
|      L1,2,3         ! les flags L1, L2 et L3 sont mis à 1
|      \SA10..20       ! les SA de SA10 à SA20 sont mis à 0
|      A1,2,3,10..30   ! les alarmes A1, A2, A3, et A10
|                      ! à A30 sont mises à 1
```

```
|      \O0..O100      ! les sorties O0 à O100 sont
|                      ! mises à 0
```

### Actions sur variables numériques

```
|      R5 = 6           ! la valeur 6 est chargée
|                      ! dans le registre numérique R5
|      R6 = Horloge - 1000 ! la valeur Horloge - 1000
|                      ! est chargée dans le registre numérique R6
|      DT6 = R8 * 5      ! le résultat de la
|                      ! multiplication de 5 par le contenu du
|                      ! registre R8 est
|                      ! chargé dans la
|                      ! temporisation DT6
|      inc R3           ! incrémentation de R3
|                      ! équivalent à R3 = R3 + 1
|      R7 = R7 + R8      ! addition du contenu des
|                      ! registres R7 et R8, le résultat est chargé
|                      ! dans R7
|      inc R7,R8         ! idem ligne précédente,
|      dec R3           ! décrémentation de R3
|                      ! équivalent à R3 = R3 - 1
```

### Remise à 0 (RAZ) de variables

Les variables booléennes peuvent être remise à 0 (Faux) par une instruction du type (voir plus haut) :

```
|      \L5             ! L5 = Faux
```

Par extension, la même syntaxe peut être utilisée pour la remise à 0 des variables numériques ou chaînes de caractères.

### Exemples

```
|      \R5             ! R5 = 0
|      \G3             ! G3 = 0
|      \R5,10,20..30   ! R5,R10,R20 à R30 = 0
|      \C1..C5,C10     ! C1 à C5,C10 vides (= '')
|      \ANO4,5,6       ! ANO4 à 6 = 0
```

### Appels de blocs de fonctions ou de fonctions sans valeur de retour

L'appel s'effectue en faisant précéder le bloc du mot clé "DO".

### Exemples

```
|      do CopyR( R100, R200, 20 ) ! copie les registres R100 à R119 vers R200 à R219
|                                   ! (appel de fonction sans valeur de retour)
|
|      do BlockX( In1:= 3, In2:= 10 ) ! appel du bloc de fonction BlockX avec les paramètres
In1 et In2
|      R10 = BlocX.Out              ! copie de la variable de sortie Out de BlockX dans R10
```

## 6.8.4 Détection de fronts

Les expressions décrites au paragraphe précédent sont vraies ou fausses selon leur état. Cela signifie qu'une condition comme :

```
|SQ10          action1
```

est vraie tant que SQ10 est à 1, fausse sinon. Action1 est effectuée tant que SQ10 est à 1, et ceci à chaque exécution du séquentiel.

Si l'action est un simple changement d'état de variable, cela ne pose pas de problème particulier : le fait de forcer une variable dans un état où elle se trouvait déjà n'a aucune conséquence.

Par contre, si l'action est un appel de fonction, par exemple, cela peut être gênant, surtout si cet appel ne doit être fait qu'une fois au moment où la condition devient vraie.

Dans certains cas ce que l'on désire tester, ce n'est donc pas l'état d'une condition, mais son changement d'état : on parle de détection de "front" :

- front montant : passage d'une condition de 0 à 1

```

1          +-----+
0 -----+
    
```

- front descendant : passage d'une condition de 1 à 0

```

1 -----+
0          +-----+
    
```

- front : changement d'état d'une condition.

Trois fonctions sont disponibles en Apigraf pour la détection de front sur une expression booléenne :

- DFM ( <expression booléenne> )

détection de front montant. La condition est vraie si l'expression passe de 0 à 1.

- DFD ( <expression booléenne> )

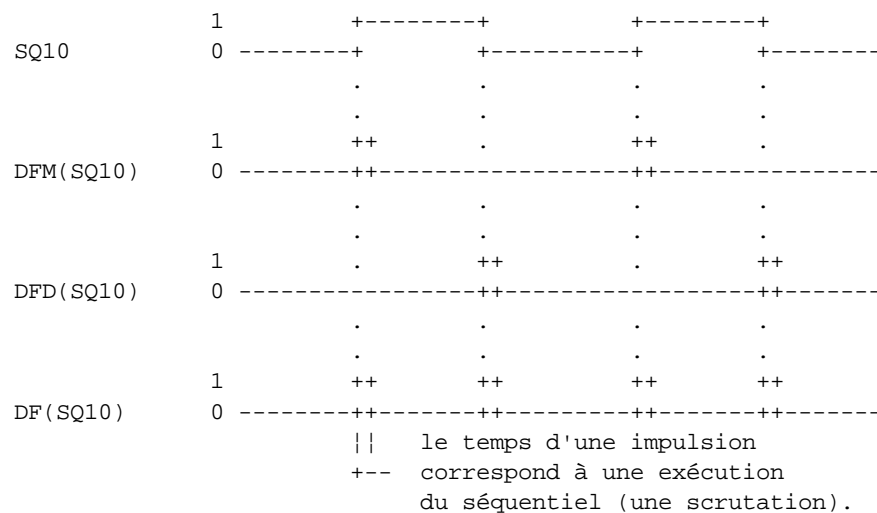
Détection de front descendant. La condition est vraie si l'expression passe de 1 à 0.

- DF ( <expression booléenne> )

Détection de front montant ou descendant (changement d'état). La condition est vraie si l'expression change d'état.

Les expressions commençant par l'une de ces fonctions ne sont actives que pendant la durée d'une seule exécution du séquentiel.

Voici par exemple le chronogramme d'une condition sans puis avec détection de front:



## Exemples

```

|DFM(SQ10)          ! condition vraie (1) si SQ10 passe
|                  ! de 0 à 1
|DFD(L10 & (R10 > 0)) ! condition vraie (1) si l'expression
|                  ! L10 & (R10 > 0) passe de 1 à 0
|DF(SQ10)           ! condition vraie (1) si SQ10 change
|                  ! d'état
    
```

Certaines précautions doivent être prises concernant l'utilisation de ces fonctions de détection de front. En particulier, elles doivent être appelées à chaque scrutation du séquentiel, de manière à pouvoir comparer la valeur de la scrutation actuelle à celle de la scrutation précédente. Ceci implique les règles suivantes (leur non-respect peut entraîner des fonctionnements erratiques) :

- Les fonctions de détection de front ne doivent pas être utilisées dans des étapes de GRAFCET, mais seulement en combinatoire (en dehors des séquences). Il faut remarquer que dans une séquence, ce genre de détection peut toujours être effectué autrement en rajoutant par exemple un pas de séquence ou en associant la détection de front en combinatoire (en dehors d'une séquence GRAFCET) à un flag testé dans le pas.
- Pour être correctement évaluées, ces fonctions ne doivent pas être utilisées au milieu d'une condition, mais toujours au tout début de la condition (en raison du mécanisme d'optimisation de l'évaluation des conditions, avec lequel l'évaluation est arrêtée dès que le résultat final est certain). Une seule de ces fonctions peut donc apparaître dans une condition.

Une détection de front doit toujours être placée en début de ligne.

Les utilisations suivantes sont déconseillées ou interdites :

- Dans une séquence SFC, ou une action, ou un bloc appelé sporadiquement.
- A l'intérieur d'une condition

```
|DFM(L1) # DFM(L2) # DFM(L3)
|                               L10
```

Si un front montant est détecté sur L1, DFM(L2) et DFM(L3) ne sont pas évaluées car le résultat de la condition est obligatoirement vrai. Lors des scrutations suivantes, la détection ne sera pas faite correctement.

L'écriture suivante serait correcte :

```
|DFM( L1 # L2 # L3 )
|                               L10
```

...mais ne donne pas le résultat souhaité : si par exemple L1 est déjà à 1, le passage de L2 de 0 à 1 ne provoque pas de détection de front, puisque l'expression  $L1 \# L2 \# L3$  ne varie pas (elle est à 1).

Dans un cas comme celui-ci, la solution serait d'écrire :

```
|DFM(L1)      L10
|DFM(L2)      L10
|DFM(L3)      L10
```

### 6.8.5 Mode d'évaluation des conditions booléennes

Les expressions booléennes sont évaluées de la gauche vers la droite. Dès que le résultat d'une expression est connu de façon certaine, l'évaluation est arrêtée. Ce mécanisme d'optimisation (commun aux autres langages textuels d'Optima PLC) permet d'augmenter considérablement la vitesse d'exécution du séquentiel.

### 6.8.6 Opérations numériques

Un éventail complet d'opérations booléennes, numériques ou sur chaînes de caractères sont disponibles, ce qui permet d'effectuer des opérations assez complexes directement dans le programme automate.

#### ATTENTION

Le signe = joue deux rôles différents suivant l'endroit où il est écrit.  
dans une condition il signifie la comparaison :

```
|R5 = R6          action1      ! si R5 est égal à R6
|                               ! faire action1
```

dans une action il joue le rôle d'affectation (écriture) :

```
|condition1      R5 = R6      ! si condition1 est vraie
```

```
|
|                                     ! alors le contenu de R6 est
|                                     ! affecté (écrit dans) à R5
R5 = R6          R7 = R8          ! si les registres R5 et R6
|                                     ! sont égaux, alors copier
|                                     ! R8 dans R7.
```

## Opérations sur variables numériques

<b>+</b>	addition de deux valeurs
<b>-</b>	soustraction de deux valeurs
<b>*</b>	multiplication de deux valeurs
<b>/ ou div</b>	division de deux valeurs numériques
<b>mod</b>	modulo

## Exemples

```
|      R20 = R6 + 4
|      R1  = R6 * R7
|      R2  = R8 div R9
|      R2  = 10 / 100
|      R2  = real_to_dint(10.0 / 100.0)
|      RR3 = 10 / R100      ! OK, mais erreur si R100 = 0
|                          ! (résultat réel)
```

## Opérateurs Inc et Dec

Les opérateurs suivants agissent sur des variables numériques entières:

<b>inc</b>	incréméntation
<b>dec</b>	décréméntation

Ces opérateurs agissent directement sur le contenu des variables et en modifient la valeur. Leur utilisation augmente la rapidité d'exécution du programme automate, cependant le programme perd en lisibilité.

## Exemples

```
|      inc R3      ! équivalent à R3 = R3 + 1
|      dec R4      ! équivalent à R4 = R4 - 1
|      inc R5, 1000 ! équivalent à R5 = R5 + 1000
|      dec R6, R7+R8 ! équivalent à R6 = R6 - (R7+R8)
```

## Comparaison de valeurs numériques

<b>=</b>	égalité
<b>&lt;&gt;</b>	différent
<b>&lt;</b>	inférieur
<b>&gt;</b>	supérieur
<b>&lt;=</b>	inférieur ou égal
<b>&gt;=</b>	supérieur ou égal

## Exemples

```
| (R1 < R2 + 3)      action1      ! action1 activée si
|                                     ! R1 est plus grand que (R2+3)
```

## Fonctions scientifiques sur valeurs numériques

Les fonctions standards disponibles dans les langages de la norme IEC 61131-3, celles définies dans les bibliothèques Optima PLC et celles définies par l'utilisateur peuvent être appelées dans une expression. Les types de données doivent être respectés.

### 6.8.7 Ecriture sur fichier de sortie texte

Les instructions "Write" et "WriteLn" permettent d'envoyer du texte sur les sorties texte présentes dans le système (imprimantes ou fichiers disque).

Les écritures sont toujours effectuées en ASCII, à la fin de la sortie (en fin de fichier si la sortie est un fichier disque).

"WriteLn" effectue en plus de "Write" un saut à la ligne suivante.

Les deux instructions attendent une suite de paramètres entre parenthèses, dont le premier est le nom d'interface (identificateur) de la sortie (par exemple LPT1).

#### Exemple

```
|L10      WriteLn (LPT1, R1, Tab, R2, Tab, R3)
|          \L10
```

Lorsque L10 passe à 1 (vrai), la sortie suivante est effectuée sur LPT1 (1ère imprimante parallèle) : valeur du registre R1, une tabulation, valeur du registre R2, une tabulation, enfin la valeur du registre R3. Un retour à la ligne suivante est ensuite effectué (séquence CR LF).

Les valeurs numériques peuvent être écrites en spécifiant un nombre de caractères ; des espaces sont alors ajoutés à gauche pour obtenir ce nombre de caractères si nécessaire.

#### Exemple

```
|          WriteLn (LPT1, R1:10, R2:10)
```

Ici, la sortie de R1 est effectuée sur 10 caractères, suivie de R2 sur également 10 caractères. Les deux valeurs sont calées à droite, donc complétées à gauche par des espaces.

### 6.8.8 Instructions Grafcet

Une structure est prévue pour définir des séquences dans le langage.

Les numéros de "STEP" de séquence sont stockés dans des variables de type "SequenceApigraf", qui doivent être définies dans les déclarations globales ou locales (il n'y a pas de variables G prédéfinies comme dans Apigraf IP).

A un instant donné, au cours de l'exécution, un seul pas peut être actif pour chaque séquence.

#### 6.8.8.1 Déclaration d'une variable de séquence

Dans les variables globales (ou locales s'il s'agit d'un bloc de fonction), insérer une déclaration du type :

```
G10 : SequenceApigraf;
```

Le type SequenceApigraf est défini dans la librairie System, qui doit donc être chargée.

#### 6.8.8.2 Définition du code d'une séquence

La syntaxe de début d'une séquence (dans le séquentiel) est la suivante:

```
| SEQUENCE xxx
```

où "xxx" indique le nom de la variable de la séquence. Le mot SEQUENCE doit toujours être placé en début de ligne.

#### 6.8.8.3 Définition d'une étape de séquence

l'instruction "STEP y" démarre la zone du programme où vont être écrites les conditions/actions associées à cette étape de la séquence GRAFCET en cours :

```
| STEP y
```



"y" indique le numéro de l'étape. C'est un entier pouvant varier de 0 à 255.

Après l'instruction STEP sont placées les instructions booléennes du type conditions/actions. Les conditions ne sont testées que lorsque la séquence est dans l'étape correspondante.

Juste après l'instruction STEP, une suite d'actions peut être définie sans condition : pour que ces actions soient exécutées, il suffit que cette étape de séquence soit active.

Les actions inconditionnelles placées juste après le mot STEP sont exécutées une seule fois, lorsque l'étape devient active. Ceci permet de charger une temporisation, par exemple, uniquement lors de l'entrée dans l'étape.

Lorsqu'il est placé en début de ligne, le mot STEP indique le début d'une nouvelle étape. S'il est placé en retrait, il s'agit d'une instruction de saut à une autre étape de la séquence (voir exemple ci-après).

#### 6.8.8.4 Fin d'une séquence

Lorsque tous les pas de séquence ont été écrits, la fin de la séquence est définie par l'instruction "FIN" :

```
| END      xxxx
```

où "xxx" doit être le même nom que dans l'instruction SEQUENCE xxx de début de séquence.

Le mot END doit toujours être placé en début de ligne.

#### 6.8.8.5 Saut à une étape de la séquence

Lorsqu'il s'agit de passer d'un pas à un autre dans une même séquence, l'instruction STEP doit être utilisée, à l'intérieur d'une partie TRANSITION.

##### Exemple

```
| ....
| !-----
| STEP 4
|           KA1           ! démarrage moteur (action)
|
| transition           ! début de la partie transition
|                       ! du PAS 4 vers les autres PAS
|
| L5 & L10      STEP 5    ! saut du pas 4 au pas 5
|
| !-----
| STEP 5
| ....
```

##### Explication

Dans cette séquence, (L5 & L10) constitue la transition (réceptivité) pour passer de l'étape 4 à l'étape 5.

La partie TRANSITION peut contenir plusieurs transitions. Cependant, si une transition vide est définie (pas de condition, seul le STEP de destination est indiqué), une seule transition est alors autorisée.

#### 6.8.8.6 Exemples de séquences

La syntaxe générale d'une séquence est la suivante:

```
| !-----
| SEQUENCE G0           ! Début de la séquence G0
| !-----
| STEP 0    ! début du pas 0
| ...
| transition
| ...
| !-----
| STEP 1    ! début du pas 1
| ...
| transition
| ...
```

```

|!
|STEP 2      ! début du pas 2
|...
|transition
|...
|!-----
|END G0      ! Fin de la séquence G0
|!-----
|
|!-----
|SEQUENCE G1      ! Début de la séquence G1
|!-----
|STEP 0
|...
|transition
|...
|!-----
|STEP 1
|...
|transition
|...
|!-----
|END G1      ! Fin séquence G1
|!-----

```

### Exemple

```

! Raz clignotant si arrêt

Arret          \G1          ! raz séquence GRAFCET G1
               \Lampe       ! extinction lampe|
!-----
SEQUENCE G1          ! début de la séquence G1
!-----
STEP 0

transition
Marche & \Arret
               STEP 1          ! passage au pas suivant
!-----
STEP 1
               Lampe          ! allumage lampe
               DT0 = R0       ! temporisation lampe allumée
transition
DT0 = 0          STEP 2          ! passage au pas suivant
!-----
STEP 2
               \Lampe       ! extinction de la lampe
               DT0 = R1       ! temporisation lampe éteinte
transition
DT0 = 0          STEP 0          ! retour au début
!-----
END G1          ! fin de la séquence G1
!-----

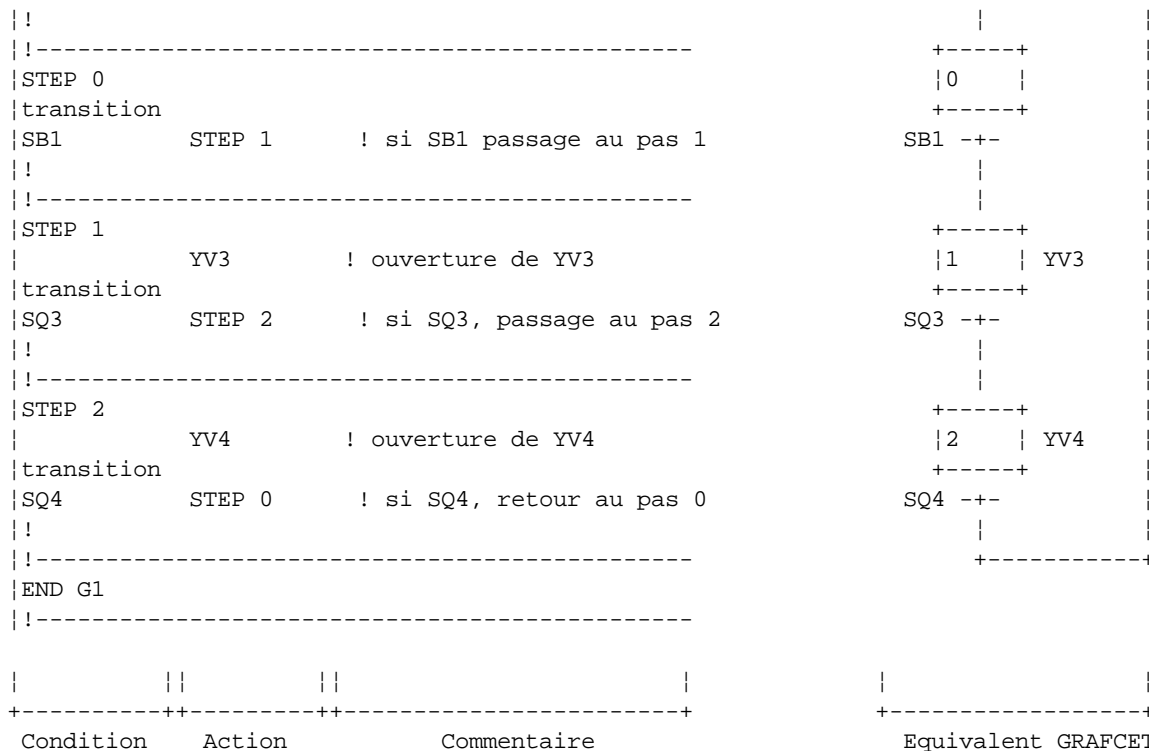
```

#### 6.8.8.7 Transposition d'un Grafcet en langage Apigraf

```

|! si arrêt d'urgence, arrêt de la séquence G1
|ArretUrgence    \G1
|
|!-----+
|SEQUENCE G1                                         +-----+

```



### 6.8.9 Différences avec le langage Apigraf originel

Un certain nombre de différences entre le langage Apigraf d'origine et celui implémenté dans Optima PLC doivent être prises en compte lors du portage d'une application d'Apigraf PLC vers Optima PLC.

En général, les noms de variables ou de fonctions sont systématiquement anglicisés.

<b>Déclarations</b>	Les déclarations de mnémoniques et de variables doivent être effectuées avec la syntaxe IEC 61131-3 dans la partie déclaration des POU concernées. Les mots-clés "TITRE", "MESSAGE", "E/S", "UTILISE", "SEQUENTIEL", etc. du langage Apigraf ne sont plus utilisables ici.
<b>Entrées/sorties TOR E et S</b>	Le préfixe d'accès aux entrées et sorties est modifié comme suit : Exxx -> Ixxx Sxxx -> Oxxx ES[xxx] ne peut plus être utilisé, à la place utiliser I[xxx] ou O[xxx] Les nouveaux préfixes correspondent maintenant systématiquement aux préfixes utilisés en anglais.
<b>Entrées/sorties analogiques EAN et SAN</b>	Le préfixe d'accès aux entrées et sorties analogiques est modifié comme suit : EANxxx -> ANIxxx SANxxx -> ANOOxxx AN[xxx] ne peut plus être utilisé, utiliser à la place ANI[xxx] ou ANO[xxx] Attention : les valeurs sont maintenant du type LREAL, et non plus de type DINT. Les nouveaux préfixes correspondent maintenant systématiquement aux préfixes utilisés en anglais.
	Cette variable indiquant la première scrutation de

<b>InitSequentiel</b>	<p>l'automate n'a plus cours dans Optima PLC. Pour effectuer une initialisation, une variable créée avec une valeur initiale définie à 1 peut être testée puis modifiée de manière similaire à InitSequentiel :</p> <pre>var   InitSequentiel : bool := true; end_var</pre> <p>Autre manière, une tâche exécutée en phase StartPLC peut être utilisée.</p>
<b>Temporisations DT</b>	<p>Le type des temporisations standard n'est plus DInt, mais TIME.</p> <p>Il est conseillé dans les nouvelles applications d'utiliser la syntaxe de la norme du type T#xxxx.</p> <p>Pour faciliter le portage de programme Apigraf existant, il est toutefois autorisé d'affecter directement une valeur numérique en millisecondes aux temporisations, ou d'utiliser une telle valeur en comparaison.</p>
<b>Grafjets</b>	Voir plus loin les différences spécifiques au grafjet.
<b>Horloge, HorlogeInterne</b>	Ces variables ne sont plus disponibles. Utiliser à la place la fonction Timer (en microsecondes).
<b>Variables horodatrices</b>	<p>Utiliser les variables :</p> <p>Year, Month, Day, DayOfWeek, Hours, Minutes, Seconds, WeekNb, DayOfYear.</p> <p>Les variables FacteurDate et FacteurSeconde sont disponibles sous les noms DateFactor et SecondFactor avec des valeurs compatibles avec celles d'Apigraf. Il est conseillé cependant d'utiliser désormais les types de variables horodatrices définies dans la norme IEC 61131-3.</p>
<b>Division</b>	<p>Dans la norme, les symboles / et <b>div</b> sont équivalents. Dans le langage Apigraf, pour compatibilité avec les applications précédentes, ces symboles sont différenciés :</p> <ul style="list-style-type: none"> <li>• / donne un résultat réel</li> <li>• <b>div</b> donne un résultat entier si les opérandes sont entiers, réel sinon.</li> </ul>
<b>Fonctions numériques</b>	<p>Certaines fonctions particulières ont été modifiées en accord avec la norme :</p> <p>ArcTan -&gt; ATan</p> <p>Tan (fonction tangente autrefois inexistante)</p>
<b>Fonctions CopieR, CopieL, SommeR</b>	<p>Ces fonctions sont anglicisées : CopyR, CopyL, SumR.</p> <p>Attention : SumR renvoie maintenant une valeur de type LInt.</p>
<b>Ecris, EcrisLigne</b>	Ces instructions sont toujours disponibles en français, mais il est conseillé d'utiliser les équivalents anglais Write et WriteLn.
<b>InclureApigraf</b>	L'inclusion de fichier n'est pas possible dans Optima PLC. Pour obtenir une structure similaire, avec des paramètres éventuels, le fichier inclus doit être transformé en fonction indépendante.

Différences propres à la syntaxe grafcet

<b>Pas de table G prédéfinie</b>	<p>La table Gx définie dans Apigraf n'existe pas dans Optima PLC. Pour définir une séquence Apigraf dans un POU Optima PLC, la variable de séquence doit être définie par le programmeur, soit dans les variables globales, soit dans les variables du POU. Le type de variable est obligatoirement "SequenceApigraf".</p> <pre>var     G10 : SequenceApigraf; end_var</pre> <p>Cette déclaration permet ensuite de définir une séquence G10. Le nom des variables n'est pas nécessairement de la forme Gx.</p>
<b>Mots clés anglicisés</b>	<p>Les mots clés sont tous anglicisés :</p> <p>PAS → STEP</p> <p>APPEL → CALL</p> <p>FIN → END</p> <p>Les autres mots clés (SEQUENCE, TRANSITION) sont inchangés.</p>
<b>Pas d'option de recherche de stabilité</b>	<p>Cette option n'existe pas dans Optima PLC.</p>
<b>Test de l'activité d'une séquence</b>	<p>La condition "G10" insérée dans une expression permettait de tester l'activité de G10, c'est à dire de savoir si une étape autre que 0 était active. Cette possibilité n'étant plus autorisée, il faut utiliser la fonction ApiSeqActive :</p> <pre>ApiSeqActive(G10)</pre> <p>renvoie true si la séquence n'est pas à l'étape 0.</p>

## 6.9 Programmation avancée

Ce chapitre décrit certaines techniques particulières d'usage courant.

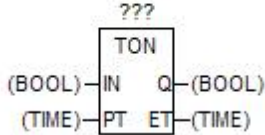
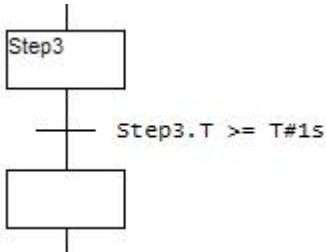
### 6.9.1 Techniques de gestion du temps

Avec tout système de développement d'automatisme se pose inévitablement la question de la meilleure manière de gérer le temps et les temporisations.

Dans Optima PLC, la gestion du temps peut être effectuée de différentes manières.

#### Méthodes standards de la norme IEC 61131-3

<b>Blocs standards TON, TOF, TP</b>	<p>Ces blocs sont définis par la norme IEC 61131-3, et permettent de gérer des temporisations sur fronts montants ou descendants.</p> <p>Ce sont des blocs de fonction, il doivent donc normalement être déclarés dans les variables pour être utilisés dans un bloc. Leur utilisation est possible dans tous les langages d'Optima PLC.</p> <p>TON est par exemple défini comme suit :</p>
---	---

	
<b>Temps écoulé pour une étape</b> <b>***.T</b>	<p>Le temps écoulé dans une étape de diagramme SFC peut être utilisé dans les transitions suivantes, comme dans cet exemple :</p>  <p>Cette syntaxe est définie dans la norme IEC 61131-3. Le temps n'est pas obligatoirement une constante, il peut être contenu dans une variable de type TIME, ou être le résultat d'une expression.</p>

### Méthodes propres à Optima PLC

<b>Temporisations DT</b>	<p>Ces temporisations sont des variables indexées de type Time, particulières en ce sens que dès le chargement avec un temps donné, la valeur décroît au fil du temps jusqu'à atteindre 0.</p> <p>Il suffit donc de charger une variable DT avec un temps, puis d'attendre que la valeur soit repassée à 0.</p> <p>Exemple en langage ST (clignotement du flag L5 à 1 Hz) :</p> <pre> if DT5 = T#0           // le temps est-il écoulé ? then L5:= not L5;      // inversion de la valeur du flag     DT5:= T#500ms      // rechargement de la temporisation end_if; </pre> <p>Ces variables sont héritées du langage et de l'automate Apigraf PLC. Elle peuvent cependant être utilisées dans tous les langages disponibles dans Optima PLC.</p>						
<b>Fonctions renvoyant la valeur courante du temps :</b> <b>Timer, Current_DT,</b> <b>Current_Date,</b> <b>Current_TOD.</b>	<p>La gestion du temps peut être parfois réalisée en mémorisant la valeur courante du temps, puis en effectuant la différence entre la valeur courante et la valeur mémorisée.</p> <p>Les fonctions Timer (Time) et Current_DT peuvent être utilisées pour cela.</p> <p>Attention toutefois à la fonction Current_DT : si elle présente l'avantage de renvoyer un temps absolu, elle est sensible à un changement de date et heure provoqué par l'utilisateur sur le PC, ou par un changement d'heure saisonnier, ce qui peut fausser les mesures de temps (voir fonctions UTC).</p> <p>La fonction Timer quant à elle renvoie un temps dont l'origine est indéfinie (ce qui ne permet de l'utiliser que dans les différences de temps écoulé), mais qui n'est pas affectée par un changement de date et d'heure.</p>						
<b>Variables horodatrices</b>	<p>Les variables suivantes permettent une gestion avancée du temps :</p> <table> <tr> <td>Year</td><td>Année courante</td></tr> <tr> <td>Month</td><td>Mois courant</td></tr> <tr> <td>Day</td><td>Quantième du mois</td></tr> </table>	Year	Année courante	Month	Mois courant	Day	Quantième du mois
Year	Année courante						
Month	Mois courant						
Day	Quantième du mois						

	Hours	Heures courantes
	Minutes	Minutes courantes
	Seconds	Secondes courantes
	DayOfYear	Numéro du jour dans l'année
	DayOfWeek	Numéro du jour dans la semaine (0 = Dimanche)
	WeekNb	Numéro de la semaine dans l'année
	DateFactor	Numérotation absolue du jour (compatibilité Apigraf)
	SecondFactor	Numérotation absolue de la seconde (compatibilité Apigraf)

#### Méthodes basées sur le temps UTC propres à Optima PLC (Temps Universel Coordonné)

<b>Fonctions renvoyant le temps UTC</b> <b>Current_DT_UTC,</b> <b>Current_Date_UTC,</b> <b>Current_TOD_UTC</b>	Ces fonctions s'utilisent de manière identique à celles citées plus haut, mais renvoient un temps indépendant du lieu ou de la période dans l'année (heure d'été). Elles sont idéales pour le stockage d'un "timestamp" absolu.																		
<b>Variables horodatrices UTC</b>	La structure UTC permet d'obtenir le temps courant :  <table> <tr> <td>UTC.Year</td><td>Année courante</td></tr> <tr> <td>UTC.Month</td><td>Mois courant</td></tr> <tr> <td>UTC.Day</td><td>Quantième du mois</td></tr> <tr> <td>UTC.Hours</td><td>Heures courantes</td></tr> <tr> <td>UTC.Minutes</td><td>Minutes courantes</td></tr> <tr> <td>UTC.Seconds</td><td>Secondes courantes</td></tr> <tr> <td>UTC.DayOfWeek</td><td>Numéro du jour dans la semaine (0 = Dimanche)</td></tr> <tr> <td>UTC.DateFactor</td><td>Numérotation absolue du jour (compatibilité Apigraf)</td></tr> <tr> <td>UTC.SecondFactor</td><td>Numérotation absolue de la seconde (compatibilité Apigraf)</td></tr> </table>	UTC.Year	Année courante	UTC.Month	Mois courant	UTC.Day	Quantième du mois	UTC.Hours	Heures courantes	UTC.Minutes	Minutes courantes	UTC.Seconds	Secondes courantes	UTC.DayOfWeek	Numéro du jour dans la semaine (0 = Dimanche)	UTC.DateFactor	Numérotation absolue du jour (compatibilité Apigraf)	UTC.SecondFactor	Numérotation absolue de la seconde (compatibilité Apigraf)
UTC.Year	Année courante																		
UTC.Month	Mois courant																		
UTC.Day	Quantième du mois																		
UTC.Hours	Heures courantes																		
UTC.Minutes	Minutes courantes																		
UTC.Seconds	Secondes courantes																		
UTC.DayOfWeek	Numéro du jour dans la semaine (0 = Dimanche)																		
UTC.DateFactor	Numérotation absolue du jour (compatibilité Apigraf)																		
UTC.SecondFactor	Numérotation absolue de la seconde (compatibilité Apigraf)																		

### 6.9.2 Affichage d'informations, d'erreurs, de demandes de confirmation

La librairie Optima PLC "Messages" comporte des fonctions et blocs de fonction permettant de faire surgir à l'écran un texte d'information ou encore une demande de confirmation.

Certains paramètres sont communs à ces fonctions :

Nom	Type	Description
<b>LogEvent</b>	<b>BOOL</b>	Ce paramètre peut être mis à 1 pour que le système enregistre le message dans le journal de bord. A 0, le message n'est pas enregistré. La valeur par défaut est 1.
<b>HelpTopic</b>	<b>STRING</b>	<p>Ce paramètre permet de définir l'accès à une aide concernant le message affiché.</p> <p>La valeur passée doit être vide si aucune aide n'est disponible. Si une aide est disponible, la valeur doit correspondre à un topic d'un fichier .chm situé dans le répertoire de l'application (si possible UserMessages.chm). Eventuellement, les topics d'aide peuvent se trouver répartis dans plusieurs fichiers .chm.</p> <p>Lorsque le message est affiché, le bouton Aide peut être cliqué pour provoquer l'ouverture de la fenêtre d'aide.</p>

### 6.9.3 Utilisation des tableaux de mesures standards

Les "tableaux de mesures" sont des structures définies interactivement dans le menu d'options de l'atelier Optima PLC. Leur intérêt réside principalement dans le fait qu'ils sont à la base des affichages de courbes dans la supervision Optima PLC.

Une fois le tableau "TabData" (par exemple) défini dans la liste des tableaux de mesure, il est possible d'utiliser les expressions suivantes dans un programme :

Expression	Description	Type
<b>TabData</b>	Référence au tableau, utilisée dans certaines fonctions	UDINT
<b>TabData[index]</b>	accès direct à une valeur du tableau	DINT
<b>TabData.Size</b>	Capacité de stockage dans le tableau	DINT
<b>TabData.Count</b>	Nombre de valeurs actuellement stockées dans le tableau	DINT
<b>TabData.Min</b>	Valeur minimale placée dans le tableau	DINT
<b>TabData.Max</b>	Valeur maximale placée dans le tableau	DINT

La bibliothèque standard "TabSamples" contient un certain nombre de fonctions de manipulation des tableaux de mesure.

La syntaxe décrite dans le tableau ci-dessus s'applique uniquement aux tableaux de mesure définis dans les options. Pour les tableaux créés dynamiquement, seules les fonctions de la librairie TabSamples peuvent être utilisées.

## 6.9.4 Utilisation des fichiers de sortie texte

Les "fichiers de sortie texte" sont des structures définies interactivement dans le menu d'options de l'atelier Optima PLC.

A la différence des fichiers généraux manipulés au moyen de la librairie "FILES", ces fichiers permettent uniquement de réaliser des sorties dans des fichiers de texte, au fil de l'eau. Cependant ces sorties sont nettement facilitées, en particulier grâce aux fonctions de la librairie "TEXTOUT".

Une fois le fichier de sortie défini, celui-ci peut être utilisé dans le programme, en particulier avec les fonctions extensibles (le nombre de paramètres est variables) TOWrite et TOWriteLn. Les variantes TOWriteCSV, TOWriteLnCSV permettent d'écrire directement des lignes au format CSV ("comma separated values", valeurs séparées par des virgules). Les variantes TOWriteTab et TOWriteLnTab permettent d'écrire directement au format TSV ("tab separated values", valeurs séparées par des tabulations). Les formats CSV et TSV sont généralement utilisés pour générer des fichiers de type tableur dans un format ouvert.

Le premier paramètre à passer pour chacune de ces fonctions est le fichier lui-même (utiliser le nom spécifié dans "interface fichier"). Ensuite, autant de paramètres que souhaité peuvent être passés, chacun avec un type simple quelconque. Les fonctions TOWriteLn ajoutent en plus un saut de ligne à la suite des champs imprimés.

Le texte envoyé sur ces fichiers de sortie peut être consulté pendant l'exécution du runtime, avec l'option [Sorties Texte](#)<sup>[34]</sup>.

## 7 Modifications "en ligne"

Optima PLC est en mesure de modifier une application en cours de fonctionnement sans l'interrompre. Certaines règles, décrites ci-après, doivent cependant être strictement respectées .

### 7.1 Application

#### 7.1.1 Modifications du programme de l'application








Les POU d'une application peuvent être modifiés en ligne. Une option du menu Compilation permet ce type de modification ([Compilation et modification en ligne](#)<sup>[21]</sup>).

Cependant des restrictions importantes peuvent être imposées selon le type de POU. Une règle générale est que les modifications autorisées ne doivent pas influencer sur la structure globale des données de l'application. Ainsi, il n'est pas possible d'ajouter ou de supprimer des variables, ou d'en modifier la taille ou le format.

Cette règle est relativement restrictive, car elle implique qu'il est impossible d'ajouter ou de retirer des blocs de fonctions. De même, les variables "statiques" des blocs de fonctions (variables d'entrée, de sortie et internes) ne peuvent pas être modifiées. Mais la modification du code s'exécutant dans les POU est bien souvent largement suffisante lors d'une mise au point.

Voici pour chaque type de POU les restrictions applicables.



	Type de POU	Modifications autorisées	Modifications interdites
	<b>Tâche</b>	Modification du trigger et de la période	Modification de la liste des programmes appelés dans la tâche. Modification de la priorité de la tâche.
	<b>Variables globales</b>	Ajout et modification de variables déclarées avec "AT" : il s'agit en général de la définition de mnémoniques. De nouveaux mnémoniques peuvent être ajoutés sans problème.  Modification du nom des variables globales normales.  Passage de variables normales à "retain" et inversement. La modification n'est cependant pas prise en compte avant la recompilation et exécution normale de l'application.	Modification des variables globales normalement déclarées (type, ajout ou suppression de variables, etc...)
	<b>définitions de types</b>	Modification des éléments d'un type énuméré.	Aucune autre modification n'est autorisée, à moins qu'elle ne porte sur un type non utilisé dans l'application.
	<b>bloc de programme</b>	voir bloc de fonction	voir bloc de fonction
	<b>bloc de fonction</b>	Modification des variables temporaires ("VAR_TEMP").  Toutes modifications du code, sauf dans le cas du SFC (graphique et textuel).  Modification du code et du mode d'appel des actions en SFC.  Inhibition et désinhibition d'appel d'actions en SFC.  Ajout d'UNE action dans les étapes SFC vides.	Modification des variables d'entrée, de sortie et des variables internes ("VAR").  Modification de la structure d'un diagramme SFC (ajout ou suppression d'étapes, de transitions et d'appels d'actions).
	<b>fonction</b>	Toutes modifications	
	<b>sous-groupe</b>	Toute modification concernant l'ordre ou la hiérarchie des POUs	Suppression de blocs de fonction ou de programme utilisés dans l'application.

### 7.1.2 Modalités de chargement des modifications

Les modifications effectuées et compilées pendant le fonctionnement de l'application sont placées dans un fichier de code d'extension .prg, comme le module d'application OptimaPLC.prg standard. Ce type de module de modification est nommé OPLCXXXX.prg, XXXX étant remplacé par la première valeur numérique disponible parmi les fichiers déjà présents, en commençant par 0000.

Le runtime prend en charge la demande de chargement du module de modification effectuée par l'atelier Optima PLC. Une confirmation préalable (désactivable) est nécessaire.

Le chargement, une fois effectué, est signalé dans le fichier events.log. Le code chargé en mémoire est alors exécuté depuis les deux modules OptimaPLC.prg et OPLC0000.prg (par exemple).

#### Chargement d'une nouvelle modification en ligne

Si un précédent module de modification était déjà présent lors du chargement d'un nouveau module, celui est retiré de la mémoire après le basculement vers le nouveau module, de cette manière un module de modification au maximum est chargé à un moment donné. En effet, un module reprend l'ensemble des modifications effectuées depuis le démarrage de l'application.

#### Démarrage en dehors de l'atelier d'une application modifiée

Si l'application ainsi modifiée est ensuite démarrée sans passer par l'atelier Optima PLC, le module OptimaPLC.prg est tout d'abord chargé normalement, puis le dernier module de modification est chargé à son tour. Si ce module a été supprimé manuellement sur le disque entre-temps, une confirmation est demandée pour démarrer sans les modifications en ligne effectuées, ou pour stopper le démarrage.

#### Suppression automatique des modules de modification en ligne

Lors d'une compilation normale produisant un fichier OptimaPLC.prg, les fichiers OPLCXXXX.prg éventuellement présents dans le répertoire sont supprimés.

## 7.2 Modifications de la supervision

Les écrans de supervision peuvent être modifiés et testés sans problème pendant le fonctionnement de l'application.

Toutefois, un écran n'est pris en compte par l'application que lors du prochain chargement de celui-ci suivant la modification effectuée.

## 8 Exemples

Le but de ces applications est de permettre une prise en main du logiciel Optima PLC par des exemples simples et ludiques.

### 8.1 Création d'un programme

Pour créer une application sur Optima PLC, il faut tout d'abord créer un projet (voir Chapitre : Création d'un projet). Appeler par exemple ce programme "Démarrage Moteur".

### Création des POUs

Les POUs (Programming Organization Unit) sont des applications qui permettent de créer :

- Tâche : POU de cadencement de l'exécution des blocs de programme
- Variables globales : POU de déclarations de variables disponibles pour l'ensemble des POUs de programmation
- Types : POU de déclarations de type de variables disponibles pour l'ensemble des POUs de programmation
- Bloc de programme : POU de programmation principal

- Bloc de fonction : sous-programme, avec ou sans valeur de retour, qui mémorise l'état de ses variables internes entre 2 appels, ainsi pour un même jeu de paramètres d'entrée le résultat renvoyé par ce bloc peut être différent (exemple du PID : pour une même consigne la sortie de ce bloc de régulation évolue jusqu'à ce qu'elle atteigne la consigne)
- Fonction : sous-programme dont la valeur de retour reste identique pour un même jeu de paramètres d'entrée (pas de mémorisation des variables internes)
- Sous-groupe : POU de structuration de la bibliothèque

Vous pourrez remarquer qu'il existe déjà deux POU dans la bibliothèque, ils composent le programme minimum.

Pour commencer, il vous faut déclarer les variables de votre système. Pour cela, créer un POU de variables globales que vous nommerez "Variable".

## 8.2 Exemple 1 : Démarrage moteur

### 8.2.1 Fonctionnement

Cette exemple est basé sur la mise en fonctionnement d'un moteur.

C'est à dire : L'appui sur "marche" provoque la mise en rotation du moteur et permet à l'auto-maintien du système de s'activer. Un appui sur "arrêt" permet la mise hors service du système.

En booléen l'équation du moteur s'écrit :

Fonctionnement := (Marche+Fonctionnement). not Arrêt

Cette équation est donc programmée avec les quatre langage de programmation actuellement existant sur Optima PLC

#### 1. ProgrammeST : Programmation en langage structuré

Le langage ST (Structured Text) est un langage textuel de haut niveau dédié aux applications d'automatisation. Ce langage est principalement utilisé pour décrire les procédures complexes, difficilement modélisables avec les langages graphiques.

#### 2. ProgrammeFBD : Programmation en diagramme de bloc de fonction (FBD)

Le langage FBD (Function Block Diagram) est un langage graphique. Il permet la construction d'équation complexes à partir des opérateurs standards, de fonctions ou de blocs fonctionnels.

#### 4. ProgrammeLD : Programmation en langage Ladder (ou langage à contact)

Le langage LD (Ladder Diagram) est une représentation graphique d'équation booléennes combinant des contacts pour les entrées et des relais pour les sorties.

#### 5. ProgrammeSFC : Programmation en Langage SFC

Le langage SFC (Sequential Function Chart), ou GRAFCET, est un langage graphique utilisé pour décrire les opérations séquentielles. Le procédé est représenté comme une suite d'étapes reliée entre elles par des transitions.

### 8.2.2 Déclarations

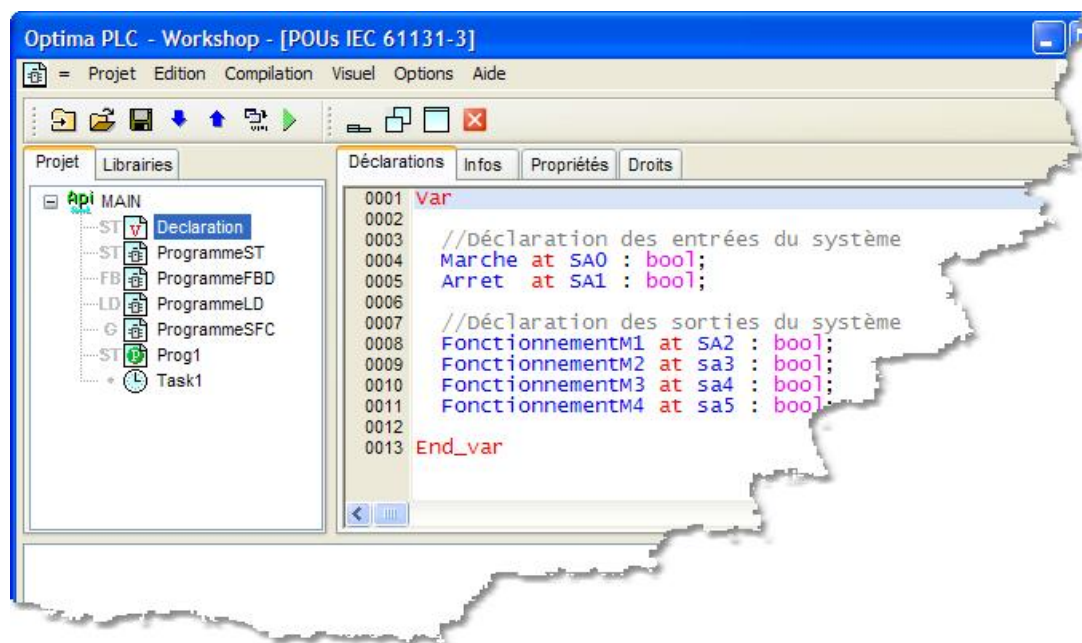
Il est nécessaire dans un premier temps de déclarer les variables qui seront les entrées et les sorties des programmes.

Pour cela créez un POU, choisissez l'option "Créer des variables globales" en faisant un clic droit sur MAIN dans votre arborescence.

Il suffit maintenant de déclarer toutes les variables nécessaires.

Par exemple :

- Marche correspond à la dénomination de la variable
- SA0 correspond à sa variable interne prédéfinie éventuellement utilisée (non obligatoire)
- bool correspond à son type, ici c'est un booléen.



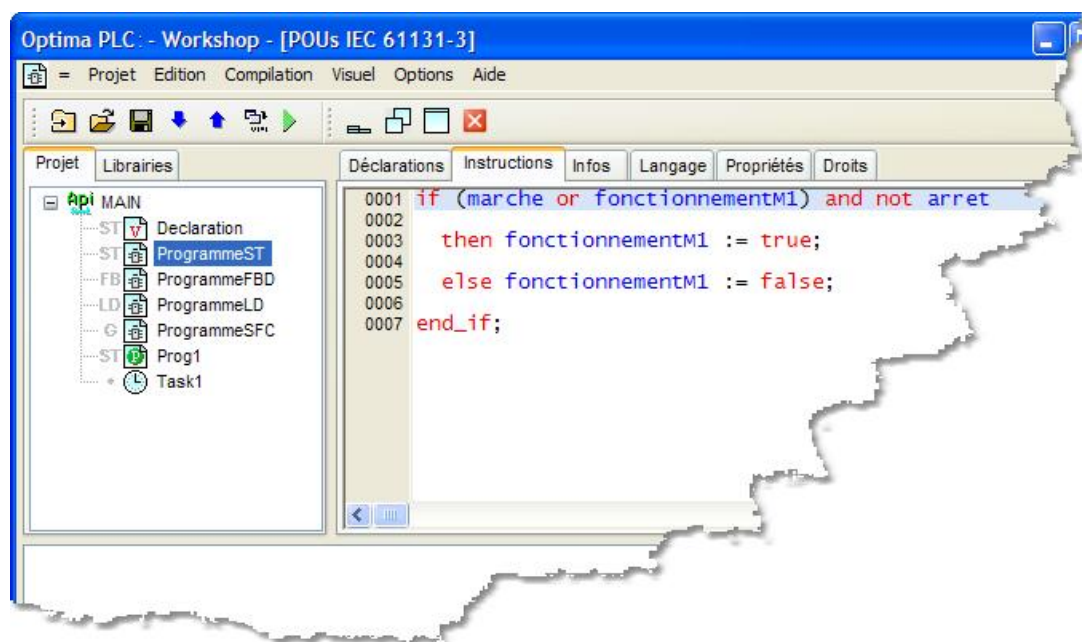
### 8.2.3 ProgrammeST

Le premier programme est un programme en langage structuré.

Faites un clic droit sur MAIN, choisissez "Créer un bloc de fonction", appelez-le ProgrammeST, choisissez l'option langage ST, puis validez.

Il faut maintenant programmer l'équation booléenne du [Fonctionnement](#) <sup>393</sup>

Tapez les instructions suivantes comme indiquées sur la copie d'écran.



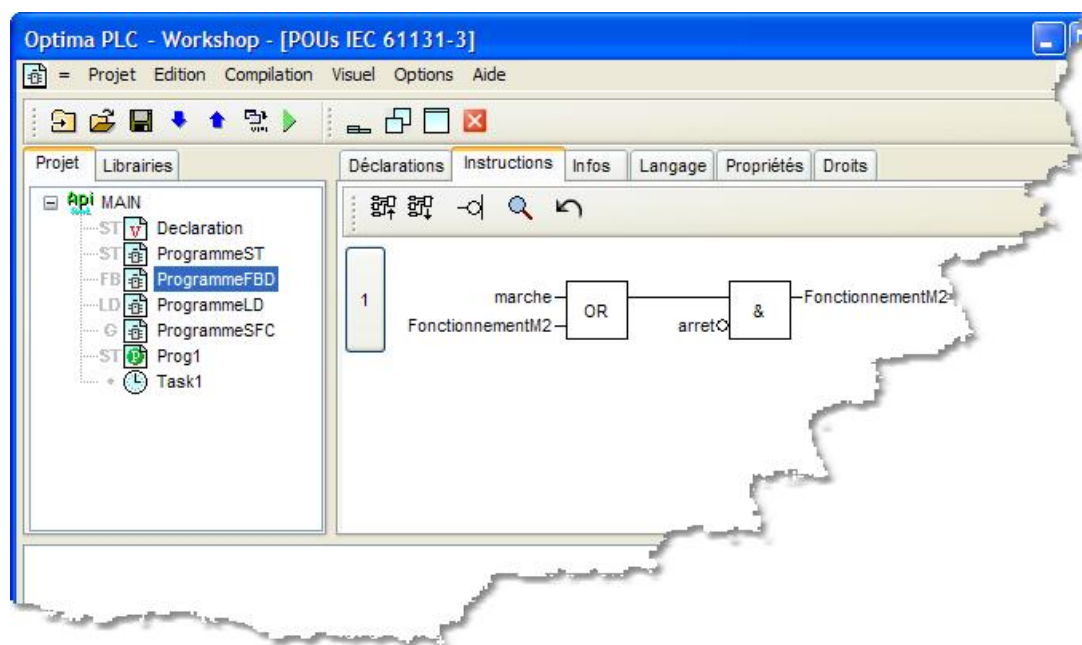
## 8.2.4 ProgrammeFBD

Le programme suivant est un programme en langage FBD.

Faites un clic droit sur MAIN, choisissez "Créer un bloc de fonction", appelez-le ProgrammeFBD, choisissez l'option langage FBD, puis validez.

Il faut maintenant programmer l'équation booléenne du [Fonctionnement](#) <sup>393</sup>

Tapez les instructions suivantes comme indiquées sur la copie d'écran.



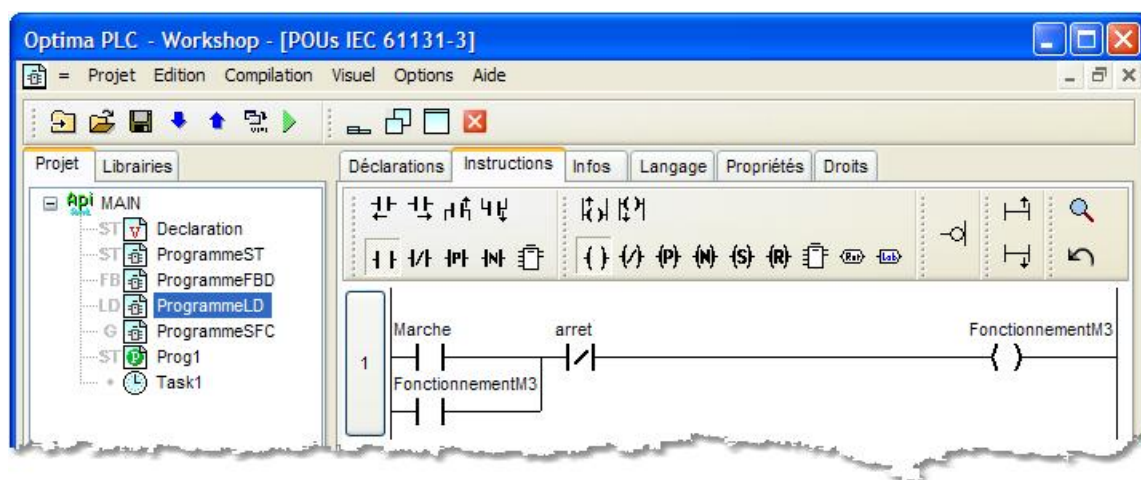
## 8.2.5 ProgrammeLD

Le programme suivant est un programme en langage Ladder (shémas contact).

Faites un clic droit sur MAIN, choisissez "Créer un bloc de fonction", appelez-le ProgrammeLD, choisissez l'option langage LD, puis validez.

Il faut maintenant programmer l'équation booléenne du [Fonctionnement](#) <sup>393</sup>

Tapez les instructions suivantes comme indiquées sur la copie d'écran.



## 8.2.6 ProgrammeSFC

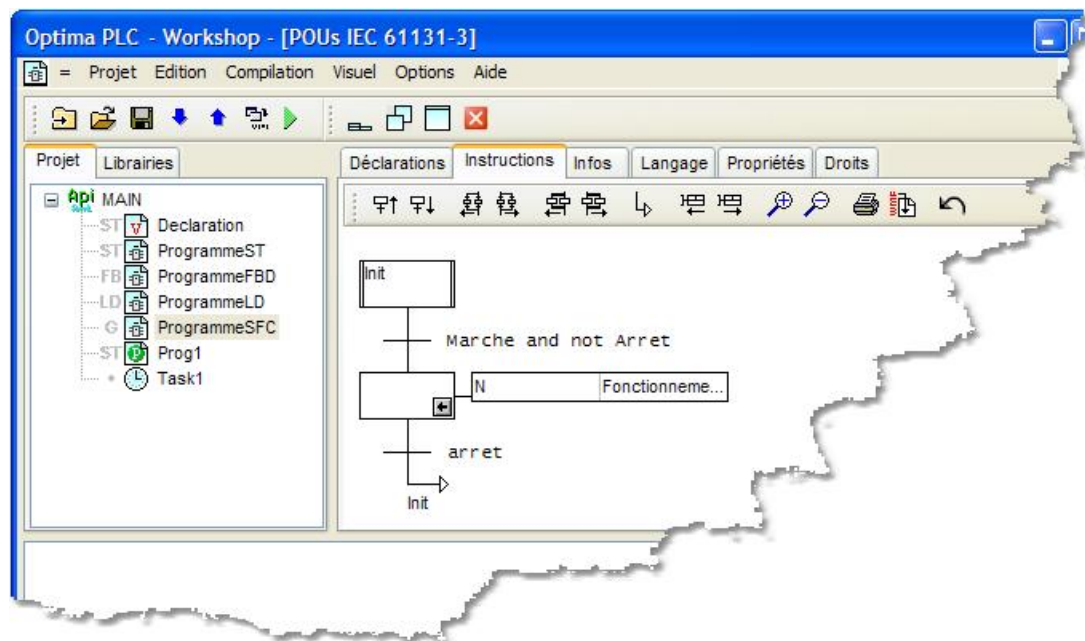
Le programme suivant est un programme en diagramme SFC (GRAFCET).

Faites un clic droit sur MAIN, choisissez "Créer un bloc de fonction", appelez-le ProgrammeSFC, choisissez

l'option langage SFC, puis validez.

Il faut maintenant programmer l'équation booléenne du [Fonctionnement](#) <sup>393</sup>

Tapez les instructions suivantes comme indiquées sur la copie d'écran.



## 8.3 Exemple 2 : Porte automatique

### 8.3.1 Fonctionnement

Description du mode de fonctionnement :

L'utilisateur se présente devant la porte, il a possibilité d'ouvrir la porte de deux manières différentes :

- Soit, il possède la clef
- Soit, il entre le code d'entrée sur l'emplacement de saisie prévu à cette effet

L'ouverture de la porte s'effectue jusqu'à ce qu'elle atteigne le capteur de fin de course d'ouverture. Elle se referme au bout d'une temporisation T1. La personne dispose d'un bouton poussoir à l'intérieur de l'enceinte pour ouvrir la porte à tout moment.

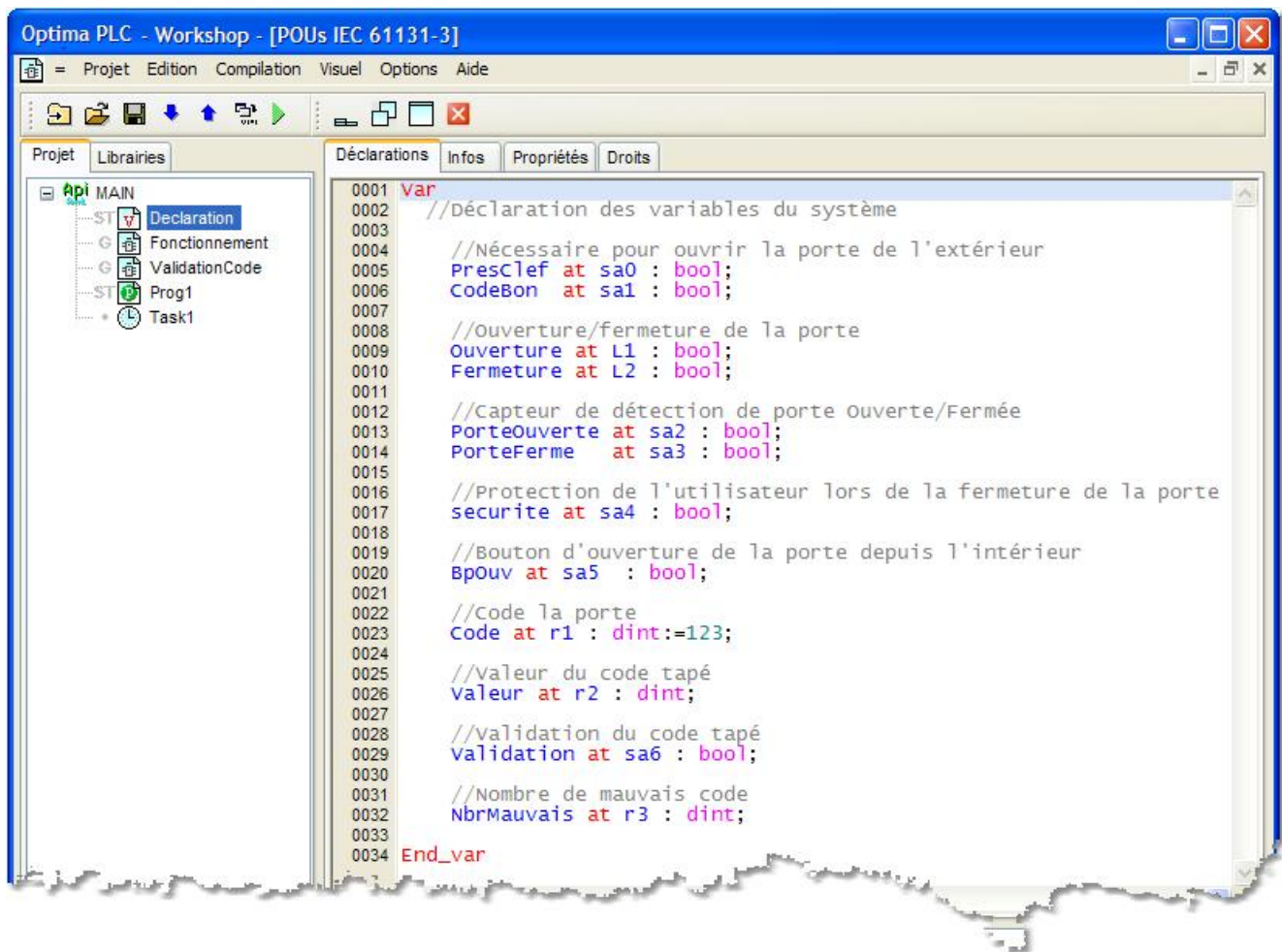
Lorsque la personne sort de l'enceinte, la porte se referme au bout d'une temporisation T2.

Par mesure de sécurité, un capteur de pression est installé au niveau de la porte. Ce capteur permet d'ouvrir la porte si quelque chose vient bloquer la porte lors de la fermeture.

L'utilisateur a droit à trois mauvais codes pour ouvrir la porte. Au bout de ces trois mauvais codes, la saisie ne sera possible qu'après une temporisation T3.

### 8.3.2 Déclarations

Déclaration des variables du système.

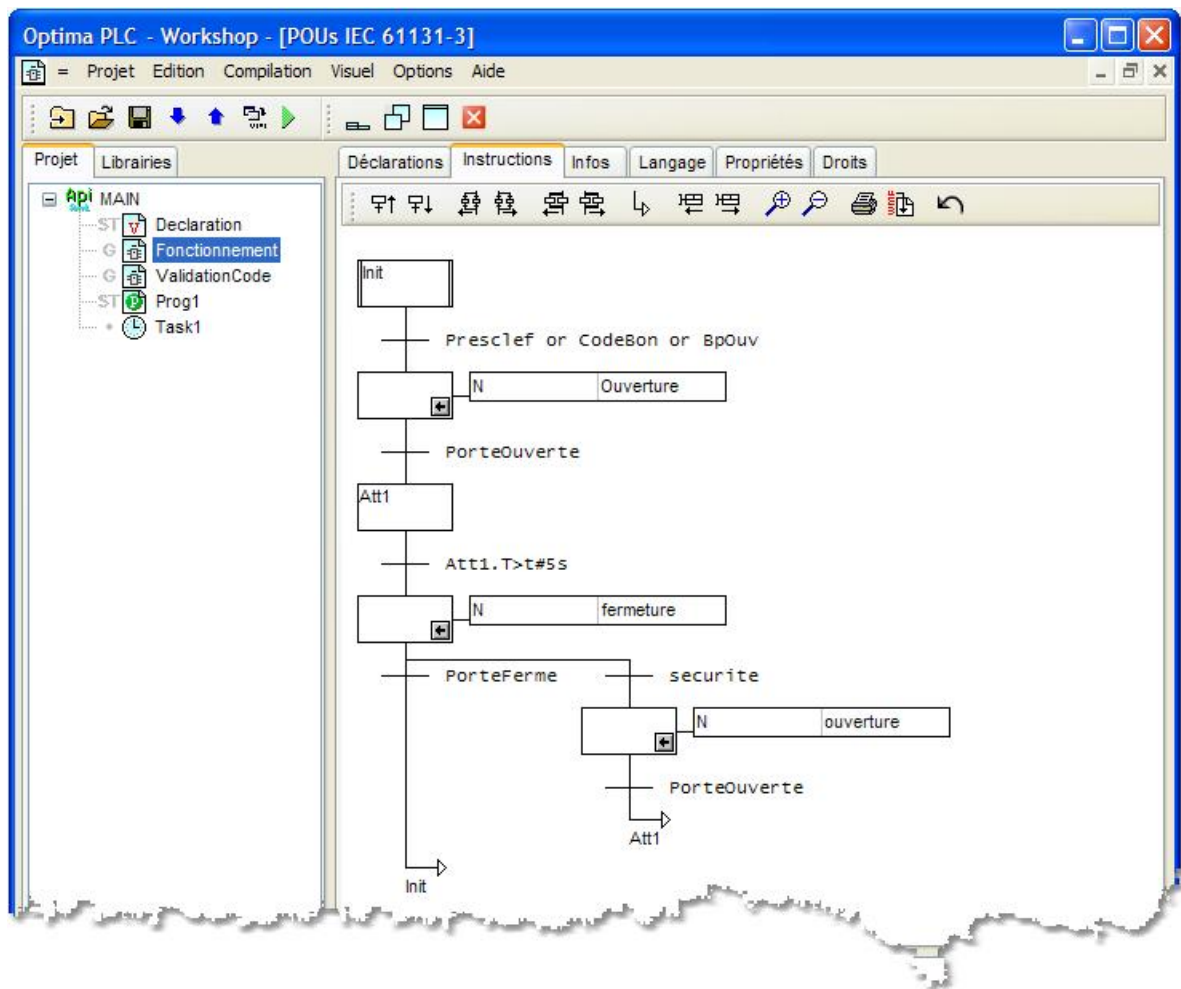


### 8.3.3 Fonctionnement

Voici le programme de fonctionnement de la porte, c'est un programme d'ouverture et de fermeture basique de porte automatique.

A ceci près que lors de la fermeture, si quelqu'un ou quelque chose bloque la fermeture de la porte, un capteur de sécurité permet la réouverture de la porte.





### 8.3.4 Validation du code

Il nous faut maintenant créer un POU qui permet une gestion du code de sécurité de la porte.

- Créer un Bloc de fonction que vous appellerez ValidationCode et choisissez le langage SFC.
- Vous disposez tout d'abord d'une étape initiale, c'est ici que nous allons initialiser les variables de ce programme (par exemple le code de la porte, ici "123")

Faire un clic droit sur l'étape et choisir "Ajouter une action", une fenêtre s'ouvre "Paramètres d'action". Cocher la case "Edition de l'action in situ en langage ST" pour permettre l'écriture de l'initialisation des variables. Valider.

- Cliquer sur la case d'action pour éditer l'action en langage structuré
- Taper la séquence suivante :

```
Code      := 123;    // initialisation du code
CodeBon   := False;  // CodeBon à 0
```

- A partir de cette étape initiale, le programme attend la validation du code tapé, dès lors il va comparer la valeur entrée "Valeur" avec la valeur réelle du code "Code".

Il est donc nécessaire de créer un "ou" suivant le cas où le code est bon ou mauvais.

- Sélectionnez l'icône "branche alternative après". Le "ou" est maintenant créé, nous allons considérer la branche de gauche comme la validation d'un code BON et inversement pour celle de droite.
- Faire un clic droit sur la transition de gauche "Editer la transition in situ (ST)" et l'éditer :

```
Validation and (Valeur = UsersCode)
```



- Faire un clic droit sur la transition de droite "Editier la transition in situ (ST)" et l'éditer :

Validation and (Valeur <> UsersCode)

"=" en ST signifie l'égalité  
 "<>" en ST signifie la différence

- Dans un premier temps, nous allons nous intéresser à la branche "code bon"

Sélectionner l'étape suivante, faire un clic droit sur celle ci et choisir "Ajouter une action", cocher la case "Edition de l'action in situ", valider. Cette étape permet de transmettre l'information code bon au programme fonctionnement. Elle réinitialise également le nombre de mauvais codes.

La transition suivante sera validée si CodeBon:=True, Cette transition est nécessaire pour s'assurer que l'information a bien été transmis au programme de la porte.

- Enfin, nous allons nous intéresser à la branche "code mauvais"

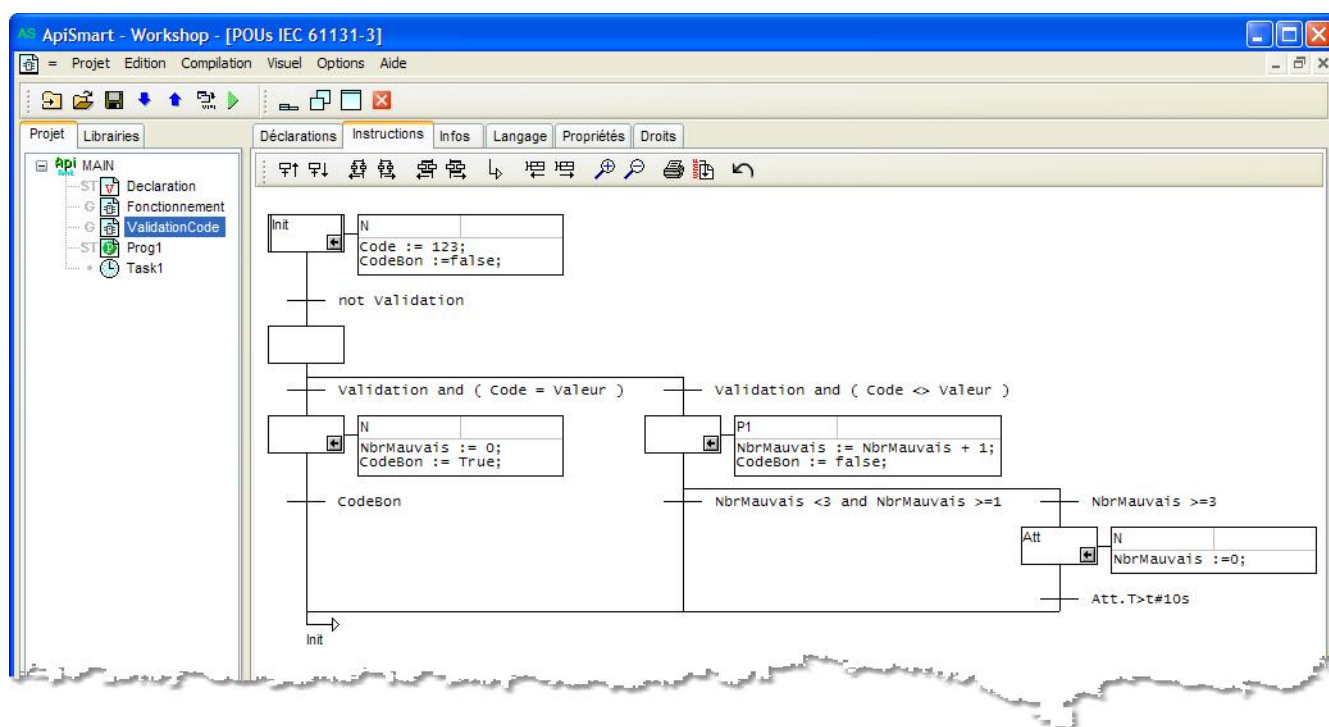
Dès lors que la validation a été effectuée et que le code tapé n'est pas valide avec le code de la porte, il faut incrémenter le nombre de mauvais codes effectués. Pour cela créer une étape qui permettra l'incrémement et qui renverra l'information CodeBon := False.

Il faut choisir le qualificateur P1 pour que l'action s'exécute une seule fois dès lors que l'étape est active.

Il y a maintenant deux cas possibles une fois l'étape active : soit le nombre de mauvais code est supérieur ou égal à trois, soit il est inférieur à trois.

Il faut donc créer un ou.

Dans le cas ou le nombre de mauvais codes est supérieur ou égal à trois, il est nécessaire de faire attendre l'utilisateur.



## 8.4 Exemples fournis avec Optima PLC

Plusieurs autres exemples sont fournis et installés en même temps qu'Optima PLC. Ils sont accessibles dans le répertoire "Exemples" situé sous le répertoire d'installation. Pour voir ces exemples, il est aussi possible de cliquer sur le bouton "Exemples" dans la fenêtre de sélection de projet.

<b>Curves</b>	Démonstration d'enregistrement de valeurs et d'affichage de courbes.
<b>Elevator</b>	Exemple simple de simulation d'ascenseur.
<b>Feux de croisement</b>	Exemple simple de cycle de signalisation d'un carrefour.
<b>Malaxeur</b>	Cycle de gestion d'un malaxeur.
<b>ModbusExchanges</b>	Démonstration de la librairie Modbus.
<b>RegulPID</b>	Démonstration des outils de régulation PID (compatibles avec les outils correspondant dans Apigraf)
<b>Tapis</b>	Exemple simple de gestion d'un graphe de pousseur de carton.
<b>Tests</b>	Ce répertoire contient un certain nombre de projets de test divers mettant chacun en oeuvre une ou plusieurs fonctions particulières d'Optima PLC.

## 9 Drivers Optima PLC

Les drivers constituent le lien entre Optima PLC et les matériels d'entrées/sorties.

Ce chapitre décrit les drivers disponibles en standard.

### 9.1 Advantech

#### 9.1.1 Carte Advantech PCL-728

*Version 1.0*



Cette interface permet l'accès aux canaux analogiques de la carte PCL-728.

2 canaux analogiques en sortie sont disponibles (1 et 2).

La configuration par défaut de la carte donne une tension de sortie comprise entre 0 et +5V.

##### 9.1.1.1 Paramètres

Pour configurer le driver, le sélectionner dans la liste des drivers puis cliquer sur le bouton "Paramètres".

Il y a trois paramètres mais seul le paramètre "Adresse de base IO" est obligatoire, les autres ont des valeurs par défaut :

- Adresse de base IO :

L'adresse IO est définie sur la carte par les DIP switches (la valeur proposée par défaut correspond à la configuration usine).

- Mode de chaque canal (configuré sur la carte avec JP1 et JP5 pour le canal 1, JP2 et JP9 pour le canal 2) :

1 pour unipolaire ou 4 - 20 ma,  
2 pour bipolaire.

### 9.1.1.2 Utilisation

#### Dans l'atelier de développement

Les entrées/sorties doivent être déclarées dans la partie "Variables globales". Les 2 sorties analogiques sont de type LREAL.

#### Exemple

```
PCL728Channel1    at ANO0    : LREAL;  
PCL728Channel2    at ANO1    : LREAL;
```

Pour les utiliser dans le programme, il est possible d'utiliser la variable (ex : ANO0) ou son mnémonique (ex : PCL728Channel1).

En unipolaire, les valeurs des canaux vont de 0 à 4095, en bipolaire de -2048 à 2047.

Exemple (canal 1 en unipolaire et canal 2 en bipolaire) :

```
PCL728Channel1:= 2000;  
  
PCL728Channel2:= -400;
```

#### Dans le runtime Optima PLC

Les entrées/sorties sont visualisables dans le menu "Automate -> Variables -> Sorties analogiques ANO".

## 9.1.2 Carte Advantech PCI-1710

Version 1.0



Ce driver permet l'utilisation de la carte multifonctions Advantech PCI-1710.

La carte dispose de 16 entrées analogiques 12 bits à gain programmable, 2 sorties analogiques 12 bits, 16 entrées et 16 sorties digitales. La répartition est faite de la manière suivante :

	Première voie	Dernière voie
<b>Entrées digitales</b>	0	15
<b>Sorties digitales</b>	16	31
<b>Entrées analogiques</b>	0	15
<b>Sorties analogiques</b>	16	17

Le driver fourni par le constructeur pour Windows peut au préalable être installé dans le but de tester le bon fonctionnement de la carte avec les utilitaires fournis.

Cependant, l'installation du driver Advantech n'est pas obligatoire pour utiliser la carte avec Optima PLC. La présence de celle-ci ainsi que son adresse de base sont automatiquement déterminées par Optima PLC, qui effectue ensuite des accès directs au matériel afin de permettre l'obtention de performances temps-réel.

La carte est également fournie sous trois autres versions :

Modèle	Différence par rapport à la PCI-1710
<b>PCI-1710L</b>	Pas de sortie analogique.
<b>PCI-1710HG</b>	Valeurs de gain plus élevées (HG = High Gain).
<b>PCI-1710HGL</b>	Valeurs de gain plus élevées et pas de sortie analogique.

### 9.1.2.1 Paramètres

Pour configurer le driver, le sélectionner dans la liste des drivers puis cliquer sur le bouton "Paramètres".

- Numéro de la carte

Ce paramètre permet à Optima PLC d'identifier la carte parmi plusieurs éventuellement présentes. 4 switches (SW1) de la carte permettent de définir un numéro d'identification de 0 à 15 (0 par défaut). Ce numéro doit être indiqué ici.

- Fréquence par canal

Nombre d'acquisitions souhaitées par seconde et par canal. Attention, cette fréquence, en particulier si elle est élevée, peut ne pas être obtenue de manière exacte. Le journal de bord mentionne au démarrage de l'application une approximation de la fréquence si celle-ci a été nécessaire.

La lecture de la FIFO de la carte consomme un temps CPU relativement élevé, une fréquence globale d'acquisition trop haute risque donc de pénaliser les fonctions d'avant plan de l'application (interface utilisateur), voire même de les bloquer. Une fréquence globale de 50000 acquisitions par seconde, toutes voies confondues, semble une limite raisonnable. Il est possible de moduler la fréquence des acquisitions avec les fonctions fournies par la librairie PCI1710, ce qui permet de ne pas pénaliser le reste du système en permanence si une fréquence élevée n'est utile qu'à des moments précis.

- Nombre de valeurs dans tableaux

Taille des tableaux de mesures attachés à chaque voie, permettant d'effectuer des acquisitions en tâche de fond (voir librairie PCI1710).

- Echelle de mesure canaux x à y

Echelle et type de mesure pour le groupe de 2 canaux spécifié.

- Tension de référence sorties analogiques

Pour chaque sortie analogique, la tension de référence peut être définie : 5V interne, 10V interne, ou tension externe. Attention : dans les deux premiers cas, la valeur à placer dans la sortie analogique est comprise respectivement entre 0 et 5.0 et entre 0 et 10.0. Dans le cas de la tension de référence externe, la valeur doit être comprise entre 0 et 4095.

### 9.1.2.2 Utilisation

#### Dans l'atelier de développement

Les entrées/sorties doivent être déclarées dans une partie "Variables globales". Les entrées analogiques sont de type LREAL.

#### Exemple

```
Channel1    at ANI0    : LREAL; // première entrée analogique
Channel2    at ANI1    : LREAL; // deuxième entrée analogique

ChanOut1    at ANO16   : LREAL; // première sortie analogique
```

Pour les utiliser dans le programme, il est possible d'utiliser la variable (ex : ANI0) ou son mnémonique (ex : Channel1).

Les valeurs sont renvoyées en Volts, en tenant compte de l'échelle.

#### Dans le runtime Optima PLC

Les entrées/sorties sont visualisables dans le menu "Automate -> Variables -> Entrées analogiques ANI" et "Sorties analogiques ANO".

#### Utilisation des tableaux de mesures

Les fonctions d'acquisitions dans des tableaux sont disponibles en utilisant la librairie PCI1710. Les données sont stockées dans ces tableaux à la fréquence d'acquisition demandée.

Les tableaux sont nommés de la manière suivante :

PCI1710\_<Carte ID>\_<No de canal>

Ainsi, le canal 10 de la carte 0 sera stocké dans le tableau PCI1710\_0\_10.

Les valeurs stockées sont entières, comprises entre 0 et 4095 si le canal est unipolaire, entre -2048 et 2047 si la voie est bipolaire. Pour chaque tableau, le coefficient d'affichage est défini en fonction de la gamme d'acquisition du canal, ce qui permet un affichage en V.

De plus, le "temps" est stocké dans un tableau :

PCI1710\_<Carte ID>\_Time

Les valeurs stockées sont les index d'acquisitions (0, 1, 2, etc.) et le coefficient est défini de manière à permettre un affichage en secondes.

### 9.1.2.3 Valeurs de statut

Statut	Description
-1	Driver non initialisé
-2	Driver stoppé en phase d'arrêt de l'application
0	Fonctionnement nominal
1	Mauvais numéro de carte (doit être compris entre 0 et 15)
2	Pas de carte PCI-1710 détectée avec le numéro de carte paramétré
3	Fréquence par canal incorrecte (doit être entre 1 et 50000 Hz)
4	Gamme de valeurs incorrecte pour un groupe de canaux

### 9.1.3 Carte Advantech PCI-1711

Version 1.0



Ce driver permet l'utilisation de la carte multifonctions Advantech PCI-1711.

La carte dispose de 16 entrées analogiques 12 bits à gain programmable, 2 sorties analogiques 12 bits, 16 entrées et 16 sorties digitales. La répartition est faite de la manière suivante :

	Première voie	Dernière voie
<b>Entrées digitales</b>	0	15
<b>Sorties digitales</b>	16	31
<b>Entrées analogiques</b>	0	15
<b>Sorties analogiques</b>	16	17

Le driver fourni par le constructeur pour Windows peut au préalable être installé dans le but de tester le bon fonctionnement de la carte avec les utilitaires fournis.

Cependant, l'installation du driver Advantech n'est pas obligatoire pour utiliser la carte avec Optima PLC. La présence de celle-ci ainsi que son adresse de base sont automatiquement déterminées par Optima PLC, qui effectue ensuite des accès directs au matériel afin de permettre l'obtention de performances temps-réel.

La carte est également fournie sous une autre version, PCI-1711L, ne disposant pas de sortie analogique.

#### 9.1.3.1 Paramètres

Pour configurer le driver, le sélectionner dans la liste des drivers puis cliquer sur le bouton "Paramètres".

- Numéro de la carte

Ce paramètre permet à Optima PLC d'identifier la carte parmi plusieurs éventuellement présentes. 4 switches (SW1) de la carte permettent de définir un numéro d'identification de 0 à 15 (0 par défaut). Ce numéro doit être indiqué ici.

- Fréquence par canal

Nombre d'acquisitions souhaitées par seconde et par canal. Attention, cette fréquence, en particulier si elle est élevée, peut ne pas être obtenue de manière exacte. Le journal de bord mentionne au démarrage de l'application une approximation de la fréquence si celle-ci a été nécessaire.

La lecture de la FIFO de la carte consomme un temps CPU relativement élevé, une fréquence globale d'acquisition trop haute risque donc de pénaliser les fonctions d'avant plan de l'application (interface utilisateur), voire même de les bloquer. Une fréquence globale de 50000 acquisitions par seconde, toutes voies confondues, semble une limite raisonnable. Il est possible de moduler la fréquence des acquisitions avec les fonctions fournies par la librairie PCI1711, ce qui permet de ne pas pénaliser le reste du système en permanence si une fréquence élevée n'est utile qu'à des moments précis.

- Nombre de valeurs dans tableaux

Taille des tableaux de mesures attachés à chaque voie, permettant d'effectuer des acquisitions en tâche de fond (voir librairie PCI1711).

- Echelle de mesure canaux x à y

Echelle et type de mesure pour le groupe de 2 canaux spécifié.

- Tension de référence sorties analogiques

Pour chaque sortie analogique, la tension de référence peut être définie : 5V interne, 10V interne, ou tension externe. Attention : dans les deux premiers cas, la valeur à placer dans la sortie analogique est comprise respectivement entre 0 et 5.0 et entre 0 et 10.0. Dans le cas de la tension de référence externe, la valeur doit être comprise entre 0 et 4095.

### 9.1.3.2 Utilisation

#### Dans l'atelier de développement

Les entrées/sorties doivent être déclarées dans une partie "Variables globales". Les entrées analogiques sont de type LREAL.

#### Exemple

```
Channel1      at ANI0      : LREAL; // première entrée analogique
Channel2      at ANI1      : LREAL; // deuxième entrée analogique

ChanOut1      at ANO16     : LREAL; // première sortie analogique
```

Pour les utiliser dans le programme, il est possible d'utiliser la variable (ex : ANI0) ou son mnémonique (ex : Channel1).

Les valeurs sont renvoyées en Volts, en tenant compte de l'échelle.

#### Dans le runtime Optima PLC

Les entrées/sorties sont visualisables dans le menu "Automate -> Variables -> Entrées analogiques ANI" et "Sorties analogiques ANO".

#### Utilisation des tableaux de mesures

Les fonctions d'acquisitions dans des tableaux sont disponibles en utilisant la librairie PCI1711. Les données sont stockées dans ces tableaux à la fréquence d'acquisition demandée.

Les tableaux sont nommés de la manière suivante :

PCI1711\_<Carte ID>\_<No de canal>

Ainsi, le canal 10 de la carte 0 sera stocké dans le tableau PCI1711\_0\_10.

Les valeurs stockées sont entières, comprises entre 0 et 4095 si le canal est unipolaire, entre -2048 et 2047 si la voie est bipolaire. Pour chaque tableau, le coefficient d'affichage est défini en fonction de la gamme d'acquisition du canal, ce qui permet un affichage en V.

De plus, le "temps" est stocké dans un tableau :

PCI1711\_<Carte ID>\_Time

Les valeurs stockées sont les index d'acquisitions (0, 1, 2, etc.) et le coefficient est défini de manière à permettre un affichage en secondes.

### 9.1.3.3 Valeurs de statut

Statut	Description
-1	Driver non initialisé
-2	Driver arrêté en phase d'arrêt de l'application
0	Fonctionnement nominal

<b>1</b>	Mauvais numéro de carte (doit être compris entre 0 et 15)
<b>2</b>	Pas de carte PCI-1711 détectée avec le numéro de carte paramétré
<b>3</b>	Fréquence par canal incorrecte (doit être entre 1 et 50000 Hz)
<b>4</b>	Gamme de valeurs incorrecte pour un groupe de canaux

### 9.1.4 Carte Advantech PCI-1715

Version 1.0



Ce driver permet l'utilisation de la carte Advantech PCI-1715.

La carte dispose de 32 entrées analogiques 12 bits à gain programmable.

Le driver fourni par le constructeur pour Windows peut au préalable être installé dans le but de tester le bon fonctionnement de la carte avec les utilitaires fournis.

Cependant, l'installation du driver Advantech n'est pas obligatoire pour utiliser la carte avec Optima PLC. La présence de celle-ci ainsi que son adresse de base sont automatiquement déterminées par Optima PLC, qui effectue ensuite des accès directs au matériel afin de permettre l'obtention de performances temps-réel.

#### 9.1.4.1 Paramètres

Pour configurer le driver, le sélectionner dans la liste des drivers puis cliquer sur le bouton "Paramètres".

- Numéro de la carte

Ce paramètre permet à Optima PLC d'identifier la carte parmi plusieurs éventuellement présentes. 4 switches (SW1) de la carte permettent de définir un numéro d'identification de 0 à 15 (0 par défaut). Ce numéro doit être indiqué ici.

- Fréquence par canal

Nombre d'acquisitions souhaitées par seconde et par canal. Attention, cette fréquence, en particulier si elle est élevée, peut ne pas être obtenue de manière exacte. Le journal de bord mentionne au démarrage de l'application une approximation de la fréquence si celle-ci a été nécessaire.

La lecture de la FIFO de la carte consomme un temps CPU relativement élevé, une fréquence globale d'acquisition trop haute risque donc de pénaliser les fonctions d'avant plan de l'application (interface utilisateur), voire même de les bloquer. Une fréquence globale de 200000 acquisitions par seconde, toutes voies confondues, semble une limite raisonnable. Il est possible de moduler la fréquence des acquisitions avec les fonctions fournies par la librairie PCI1715, ce qui permet de ne pas pénaliser le reste du système en permanence si une fréquence élevée n'est utile qu'à des moments précis.

- Nombre de valeurs dans tableaux

Taille des tableaux de mesures attachés à chaque voie, permettant d'effectuer des acquisitions en tâche de fond (voir librairie PCI1715).



- Echelle de mesure canaux x à y

Echelle et type de mesure pour le groupe de 2 canaux spécifié.

### 9.1.4.2 Utilisation

#### Dans l'atelier de développement

Les entrées/sorties doivent être déclarées dans une partie "Variables globales". Les entrées analogiques sont de type LREAL.

#### Exemple

```
Channel1      at ANI0      : LREAL; // première entrée analogique
Channel2      at ANI1      : LREAL; // deuxième entrée analogique

ChanOut1      at ANO16     : LREAL; // première sortie analogique
```

Pour les utiliser dans le programme, il est possible d'utiliser la variable (ex : ANI0) ou son mnémonique (ex : Channel1).

Les valeurs sont renvoyées en Volts, en tenant compte de l'échelle.

#### Dans le runtime Optima PLC

Les entrées/sorties sont visualisables dans le menu "Automate -> Variables -> Entrées analogiques ANI" et "Sorties analogiques ANO".

#### Utilisation des tableaux de mesures

Les fonctions d'acquisitions dans des tableaux sont disponibles en utilisant la librairie PCI1715. Les données sont stockées dans ces tableaux à la fréquence d'acquisition demandée.

Les tableaux sont nommés de la manière suivante :

PCI1715\_<Carte ID>\_<No de canal>

Ainsi, le canal 10 de la carte 0 sera stocké dans le tableau PCI1715\_0\_10.

Les valeurs stockées sont entières, comprises entre 0 et 4095 si le canal est unipolaire, entre -2048 et 2047 si la voie est bipolaire. Pour chaque tableau, le coefficient d'affichage est défini en fonction de la gamme d'acquisition du canal, ce qui permet un affichage en V.

De plus, le "temps" est stocké dans un tableau :

PCI1715\_<Carte ID>\_Time

Les valeurs stockées sont les index d'acquisitions (0, 1, 2, etc.) et le coefficient est défini de manière à permettre un affichage en secondes.

### 9.1.4.3 Valeurs de statut

Statut	Description
<b>-1</b>	Driver non initialisé
<b>-2</b>	Driver stoppé en phase d'arrêt de l'application
<b>0</b>	Fonctionnement nominal
<b>1</b>	Mauvais numéro de carte (doit être compris entre 0 et 15)
<b>2</b>	Pas de carte PCI-1715 détectée avec le numéro de carte paramétré
<b>3</b>	Fréquence par canal incorrecte (doit être entre 1 et

	100000 Hz)
4	Gamme de valeurs incorrecte pour un groupe de canaux

### 9.1.5 Carte Advantech PCI-1720

Version 1.0



Ce driver permet l'utilisation de la carte Advantech PCI-1720.

La carte dispose de 4 sorties analogiques 12 bits avec étendue configurable.

Le driver fourni par le constructeur pour Windows peut au préalable être installé dans le but de tester le bon fonctionnement de la carte avec les utilitaires fournis.

Cependant, l'installation du driver Advantech n'est pas obligatoire pour utiliser la carte avec Optima PLC. La présence de celle-ci ainsi que son adresse de base sont automatiquement déterminées par Optima PLC, qui effectue ensuite des accès directs au matériel afin de permettre l'obtention de performances temps-réel.

Les voies analogiques sont mises à jour en sortie de manière synchrone. Les valeurs sont à donner en Volts dans les canaux de sortie de l'application.

#### 9.1.5.1 Paramètres

Pour configurer le driver, le sélectionner dans la liste des drivers puis cliquer sur le bouton "Paramètres".

- Numéro de la carte

Ce paramètre permet à Optima PLC d'identifier la carte parmi plusieurs éventuellement présentes. 4 switchs (S1) de la carte permettent de définir un numéro d'identification de 0 à 15 (0 par défaut). Ce numéro doit être indiqué ici.

- Plage de tension du canal 0
- Plage de tension du canal 1
- Plage de tension du canal 2
- Plage de tension du canal 3

Pour chaque canal, une plage de tension peut être choisie parmi :

- 0V à 5V
- 0V à 10V
- -5V à 5V
- -10V à 10V

#### 9.1.5.2 Valeurs de statut

Statut	Description
-1	Driver non initialisé
-2	Driver stoppé en phase d'arrêt de l'application
0	Fonctionnement nominal
1	Mauvais numéro de carte (doit être compris entre 0 et 15)
2	Pas de carte PCI-1720 détectée avec le numéro de carte paramétré

## 9.1.6 Carte Advantech PCI-1723

Version 1.0



Ce driver permet l'utilisation de la carte Advantech PCI-1723.

La carte dispose de 8 sorties analogiques 16 bits, et de 16 canaux digitaux d'entrée ou de sortie.

Le driver fourni par le constructeur pour Windows peut au préalable être installé dans le but de tester le bon fonctionnement de la carte avec les utilitaires fournis.

Cependant, l'installation du driver Advantech n'est pas obligatoire pour utiliser la carte avec Optima PLC. La présence de celle-ci ainsi que son adresse de base sont automatiquement déterminées par Optima PLC, qui effectue ensuite des accès directs au matériel afin de permettre l'obtention de performances temps-réel.

Les voies analogiques sont mises à jour en sortie de manière synchrone. Les valeurs sont à donner en Volts dans les canaux de sortie de l'application.

Les 16 entrées/sorties digitales peuvent être programmées en entrée ou en sortie par groupe de 8.

### 9.1.6.1 Paramètres

Pour configurer le driver, le sélectionner dans la liste des drivers puis cliquer sur le bouton "Paramètres".

- Numéro de la carte

Ce paramètre permet à Optima PLC d'identifier la carte parmi plusieurs éventuellement présentes. 4 switchs (S1) de la carte permettent de définir un numéro d'identification de 0 à 15 (0 par défaut). Ce numéro doit être indiqué ici.

- Direction des entrées/sorties digitales 0 à 7

Les 8 premières E/S digitales peuvent être programmées soit en entrée soit en sortie.

- Direction des entrées/sorties digitales 8 à 15

Les 8 dernières E/S digitales peuvent être programmées soit en entrée soit en sortie.

### 9.1.6.2 Valeurs de statut

Statut	Description
-1	Driver non initialisé
-2	Driver stoppé en phase d'arrêt de l'application
0	Fonctionnement nominal
1	Mauvais numéro de carte (doit être compris entre 0 et 15)
2	Pas de carte PCI-1723 détectée avec le numéro de carte paramétré

### 9.1.7 Carte Advantech PCI-1733

Version 1.0



Ce driver donne accès aux 32 entrées binaires de la carte Advantech PCI-1733.

Le driver fourni par le constructeur pour Windows peut au préalable être installé dans le but de tester le bon fonctionnement de la carte avec les utilitaires fournis.

Cependant, l'installation du driver Advantech n'est pas obligatoire pour utiliser la carte avec Optima PLC. La présence de celle-ci ainsi que son adresse de base sont automatiquement déterminées par Optima PLC, qui effectue ensuite des accès directs au matériel afin de permettre l'obtention de performances temps-réel.

#### 9.1.7.1 Paramètres

- Numéro de la carte

Ce paramètre permet à Optima PLC d'identifier la carte parmi plusieurs éventuellement présentes. 4 switches (SW1) de la carte permettent de définir un numéro d'identification de 0 à 15 (0 par défaut). Ce numéro doit être indiqué ici.

#### 9.1.7.2 Valeurs de statut

Statut	Description
-1	Driver non initialisé
-2	Driver stoppé en phase d'arrêt de l'application
0	Fonctionnement nominal
1	Mauvais numéro de carte (doit être compris entre 0 et 15)
2	Carte non détectée avec le numéro indiqué.

### 9.1.8 Carte Advantech PCI-1734

Version 1.0



Ce driver donne accès aux 32 sorties binaires de la carte Advantech PCI-1734.

Le driver fourni par le constructeur pour Windows peut au préalable être installé dans le but de tester le bon fonctionnement de la carte avec les utilitaires fournis.

Cependant, l'installation du driver Advantech n'est pas obligatoire pour utiliser la carte avec Optima PLC. La présence de celle-ci ainsi que son adresse de base sont automatiquement déterminées par Optima PLC, qui effectue ensuite des accès directs au matériel afin de permettre l'obtention de performances temps-réel.

### 9.1.8.1 Paramètres

- Numéro de la carte

Ce paramètre permet à Optima PLC d'identifier la carte parmi plusieurs éventuellement présentes. 4 switches (SW1) de la carte permettent de définir un numéro d'identification de 0 à 15 (0 par défaut). Ce numéro doit être indiqué ici.

### 9.1.8.2 Valeurs de statut

Statut	Description
-1	Driver non initialisé
-2	Driver stoppé en phase d'arrêt de l'application
0	Fonctionnement nominal
1	Mauvais numéro de carte (doit être compris entre 0 et 15)
2	Carte non détectée avec le numéro indiqué.

### 9.1.9 Carte Advantech PCI-1747U

Version 1.0



Ce driver donne accès aux entrées analogiques de la carte Advantech PCI-1747U.

La carte dispose de 64 entrées analogiques 16 bits à gain programmable.

Le driver fourni par le constructeur pour Windows peut au préalable être installé dans le but de tester le bon fonctionnement de la carte avec les utilitaires fournis.

Cependant, l'installation du driver Advantech n'est pas obligatoire pour utiliser la carte avec Optima PLC. La présence de celle-ci ainsi que son adresse de base sont automatiquement déterminées par Optima PLC, qui effectue ensuite des accès directs au matériel afin de permettre l'obtention de performances temps-réel.

Le driver peut être utilisé conjointement à la librairie PCI1747U, qui permet un stockage des valeurs acquises dans des tableaux.

### 9.1.9.1 Paramètres

Pour configurer le driver, le sélectionner dans la liste des drivers puis cliquer sur le bouton "Paramètres".

- Numéro de la carte

Ce paramètre permet à Optima PLC d'identifier la carte parmi plusieurs éventuellement présentes. 4 switches (SW1) de la carte permettent de définir un numéro d'identification de 0 à 15 (0 par défaut). Ce numéro doit être indiqué ici.

- Fréquence par canal

Nombre d'acquisitions souhaitées par seconde et par canal. Attention, cette fréquence, en particulier si elle est élevée, peut ne pas être obtenue de manière exacte. Le journal de bord mentionne au démarrage de l'application une approximation de la fréquence si celle-ci a été nécessaire.

La lecture de la FIFO de la carte consomme un temps CPU relativement élevé, une fréquence globale d'acquisition trop haute risque donc de pénaliser les fonctions d'avant plan de l'application (interface utilisateur), voire même de les bloquer. Une fréquence globale de 100000 acquisitions par seconde, toutes voies confondues, semble une limite raisonnable. Il est possible de moduler la fréquence des acquisitions avec les fonctions fournies par la librairie PCI1747U, ce qui permet de ne pas pénaliser le reste du système en permanence si une fréquence élevée n'est utile qu'à des moments précis.

- Nombre de valeurs dans tableaux

Taille des tableaux de mesures attachés à chaque voie, permettant d'effectuer des acquisitions en tâche de fond (voir librairie PCI1747U).

- Echelle de mesure canaux x à y

Echelle et type de mesure pour le groupe de 4 canaux spécifié.

### 9.1.9.2 Utilisation

#### Dans l'atelier de développement

Les entrées/sorties doivent être déclarées dans une partie "Variables globales". Les entrées analogiques sont de type LREAL.

#### Exemple

```
Channel1    at ANI0    : LREAL;
Channel2    at ANI1    : LREAL;
```

Pour les utiliser dans le programme, il est possible d'utiliser la variable (ex : ANI0) ou son mnémonique (ex : Channel1).

Les valeurs sont renvoyées en Volts, en tenant compte de l'échelle.

#### Dans le runtime Optima PLC

Les entrées/sorties sont visualisables dans le menu "Automate -> Variables -> Entrées analogiques ANI".

#### Utilisation des tableaux de mesures

Les fonctions d'acquisitions dans des tableaux sont disponibles en utilisant la librairie PCI1747U. Les données sont stockées dans ces tableaux à la fréquence d'acquisition demandée.

Les tableaux sont nommés de la manière suivante :

PCI1747U\_<Carte ID>\_<No de canal>

Ainsi, le canal 10 de la carte 0 sera stocké dans le tableau PCI1747U\_0\_10.

Les valeurs stockées sont entières, comprises entre 0 et 65535 si le canal est unipolaire, entre -32768 et 32767 si la voie est bipolaire. Pour chaque tableau, le coefficient d'affichage est défini en fonction de la gamme d'acquisition du canal, ce qui permet un affichage en V.

De plus, le "temps" est stocké dans un tableau :

PCI1747U\_<Carte ID>\_Time

Les valeurs stockées sont les index d'acquisitions (0, 1, 2, etc.) et le coefficient est défini de manière à permettre un affichage en secondes.

### 9.1.9.3 Valeurs de statut

Statut	Description
-1	Driver non initialisé
-2	Driver stoppé en phase d'arrêt de l'application
0	Fonctionnement nominal
1	Mauvais numéro de carte (doit être compris entre 0 et 15)
2	Pas de carte PCI-1747U détectée avec le numéro de carte paramétré
3	Fréquence par canal incorrecte (doit être entre 1 et 50000 Hz)
4	Gamme de valeurs incorrecte pour un groupe de canaux

### 9.1.10 Carte Advantech PCI-1751

Version 1.2



Ce driver donne accès aux entrées/sorties binaires de la carte Advantech PCI-1751.

Les groupes de canaux peuvent être utilisés en entrée ou en sortie. Les modes programmés sur la carte peuvent être définis par des straps sur celle-ci, ou bien par programmation, en accord avec la définition des canaux dans l'application.

Le driver fourni par le constructeur pour Windows peut au préalable être installé dans le but de tester le bon fonctionnement de la carte avec les utilitaires fournis.

Cependant, l'installation du driver Advantech n'est pas obligatoire pour utiliser la carte avec Optima PLC. La présence de celle-ci ainsi que son adresse de base sont automatiquement déterminées par Optima PLC, qui effectue ensuite des accès directs au matériel afin de permettre l'obtention de performances temps-réel.

#### 9.1.10.1 Paramètres

- Numéro de la carte

Le système peut contenir plusieurs cartes PCI 1751. Ce paramètre permet de préciser laquelle est adressée ici. Le numéro de la première carte est 1. C'est la valeur par défaut.

- Masque d'inversion IO 0 à 7

- Masque d'inversion IO 8 à 15
- Masque d'inversion IO 16 à 23
- Masque d'inversion IO 24 à 31
- Masque d'inversion IO 32 à 39

Ces paramètres permettent de définir une inversion pour chaque IO par rapport à l'état rapporté par la carte. Il s'agit d'un masque dont les bits à 1 indiquent une inversion de l'emplacement correspondant.

Par exemple, la valeur de masque \$21 pour le masque d'inversion des IO 0 à 7, indique que les IO 0 et 5 du driver doivent être inversés.

Les valeurs par défaut sont \$FF, car l'état originel de chaque entrée est à 1 en l'absence de signal.

### 9.1.10.2 Valeurs de statut

Statut	Description
-1	Driver non initialisé
-2	Driver stoppé en phase d'arrêt de l'application
0	Fonctionnement nominal
1	Mauvais numéro de carte (doit être compris entre 1 et 16)
2	Carte non détectée

### 9.1.11 Carte Advantech PCI-1762

Version 1.0



Ce driver donne accès aux 16 entrées isolées et 16 sorties relais de la carte Advantech PCI-1762. 16 entrées supplémentaires de contrôle de l'état des relais en sortie sont également disponibles.

Les canaux se répartissent ainsi :

Premier canal	Dernier canal	Utilisation
0	15	Entrées isolée de la carte
16	31	Etat de contrôle des 16 sorties relais
32	47	Sorties relais de la carte

Le driver fourni par le constructeur pour Windows peut au préalable être installé dans le but de tester le bon fonctionnement de la carte avec les utilitaires fournis.

Cependant, l'installation du driver Advantech n'est pas obligatoire pour utiliser la carte avec Optima PLC. La présence de celle-ci ainsi que son adresse de base sont automatiquement déterminées par Optima PLC, qui effectue ensuite des accès directs au matériel afin de permettre l'obtention de performances temps-réel.



### 9.1.11.1 Paramètres

- Numéro de la carte

Ce paramètre permet à Optima PLC d'identifier la carte parmi plusieurs éventuellement présentes. 4 switches (SW1) de la carte permettent de définir un numéro d'identification de 0 à 15 (0 par défaut). Ce numéro doit être indiqué ici.

### 9.1.11.2 Valeurs de statut

Statut	Description
-1	Driver non initialisé
-2	Driver stoppé en phase d'arrêt de l'application
0	Fonctionnement nominal
1	Mauvais numéro de carte (doit être compris entre 0 et 15)
2	Carte non détectée avec le numéro indiqué

## 9.1.12 Carte Advantech PCI-1780U

Version 1.0



Cette carte est principalement utilisée pour ses 8 compteurs 16 bits. Cependant ce driver permet également de tirer parti des 8 entrées et 8 sorties TOR qu'elle comporte.

Le driver fourni par le constructeur pour Windows peut au préalable être installé dans le but de tester le bon fonctionnement de la carte avec les utilitaires fournis.

Cependant, l'installation du driver Advantech n'est pas obligatoire pour utiliser la carte avec Optima PLC. La présence de celle-ci ainsi que son adresse de base sont automatiquement déterminées par Optima PLC, qui effectue ensuite des accès directs au matériel afin de permettre l'obtention de performances temps-réel.

Le driver peut être utilisé conjointement à la librairie PCI1780U, qui permet la programmation des compteurs.

Les 8 premières voies de la carte (0 à 7) sont en entrée, les 8 suivantes (8 à 15) sont en sortie.

### 9.1.12.1 Paramètres

Pour configurer le driver, le sélectionner dans la liste des drivers puis cliquer sur le bouton "Paramètres".

- Numéro de la carte

Ce paramètre permet à Optima PLC d'identifier la carte parmi plusieurs éventuellement présentes. 4 switches (SW1) de la carte permettent de définir un numéro d'identification de 0 à 15 (0 par défaut). Ce numéro doit être indiqué ici.

### 9.1.12.2 Valeurs de statut

Statut	Description
-1	Driver non initialisé
-2	Driver stoppé en phase d'arrêt de l'application
0	Fonctionnement nominal

<b>1</b>	Mauvais numéro de carte (doit être compris entre 0 et 15)
<b>2</b>	Pas de carte PCI-1780U détectée avec le numéro de carte paramétré

### 9.1.13 Modules Adam

#### 9.1.13.1 Adam 4017, 4018

Version 1.1



Ces modules procurent 8 entrées analogiques 16 bits sur bus RS485.

La communication avec un module est effectuée via un port de communication sériel RS 485, qui doit être configuré dans la configuration des ports sériels d'Optima PLC.

La configuration d'un module (vitesse de communication, adresse, etc.), ainsi que son éventuelle calibration, peut être effectuée avec le logiciel fourni par Advantech. Le fonctionnement du driver Optima PLC suppose que cette configuration est déjà effectuée et que l'accès au module fonctionne correctement.

##### 9.1.13.1.1 Paramètres

Pour configurer le driver, le sélectionner dans la liste des drivers puis cliquer sur le bouton "Paramètres".

- Numéro du port sériel

Indiquer ici le numéro de port COM utilisé pour la communication avec le module. Ce port COM doit par ailleurs être configuré et activé dans Optima PLC.

- Adresse du module

Ce paramètre correspond à l'adresse donnée au module Adam avec le logiciel Advantech.

- Vitesse de communication

Indiquer ici la vitesse de communication donnée lors de la configuration du module. Cette vitesse doit également correspondre à celle donnée au port COM.

- Affichage de message d'erreur

Si la valeur "oui" est sélectionnée, un défaut de communication avec le module (hormis lors de l'initialisation de l'application, où un autre type de message est systématiquement affiché) est signalé à l'écran.

##### 9.1.13.1.2 Valeurs de statut

Statut	Description
<b>-1</b>	Driver non initialisé
<b>-2</b>	Driver stoppé en phase d'arrêt de l'application
<b>0</b>	Fonctionnement nominal

<b>1, 241</b>	Défaut de communication (pas de réponse)
<b>2, 242</b>	Défaut de communication
<b>3, 243</b>	Port COM non activé
<b>4, 244</b>	Adresse de module incorrecte
<b>5, 245</b>	Un seul module attendu sur le port COM (obsolète)
<b>6, 246</b>	Adresse de module dupliquée sur le port COM
<b>7, 247</b>	Incohérence de vitesse utilisée sur le port COM
<b>255</b>	Module non utilisé

### 9.1.13.2 Adam 4050, 4053, 4055

Version 1.1



Ces modules procurent des entrées/sorties digitales sur bus RS485.

	Entrées	Sorties
<b>Adam 4050</b>	7 (voies 0 à 6)	8 (voies 8 à 15)
<b>Adam 4053</b>	16 (voies 0 à 15)	0
<b>Adam 4055</b>	8 (voies 0 à 7)	8 (voies 8 à 15)

La communication avec un module est effectuée via un port de communication sériel RS 485, qui doit être configuré dans la configuration des ports sériels d'Optima PLC.

La configuration d'un module (vitesse de communication, adresse, etc.), ainsi que son éventuelle calibration, peut être effectuée avec le logiciel fourni par Advantech. Le fonctionnement du driver Optima PLC suppose que cette configuration est déjà effectuée et que l'accès au module fonctionne correctement.

#### 9.1.13.2.1 Paramètres

Pour configurer le driver, le sélectionner dans la liste des drivers puis cliquer sur le bouton "Paramètres".

- Numéro du port sériel

Indiquer ici le numéro de port COM utilisé pour la communication avec le module. Ce port COM doit par ailleurs être configuré et activé dans Optima PLC.

- Adresse du module

Ce paramètre correspond à l'adresse donnée au module Adam avec le logiciel Advantech.

- Echelle de mesure

Ce paramètre définit la pleine échelle et le type des signaux mesurés.

- Vitesse de communication

Indiquer ici la vitesse de communication donnée lors de la configuration du module. Cette vitesse doit également correspondre à celle donnée au port COM.

- Affichage de message d'erreur

Si la valeur "oui" est sélectionnée, un défaut de communication avec le module (hormis lors de l'initialisation de l'application, où un autre type de message est affiché) est signalé à l'écran.

- Mode compatibilité

Ce paramètre permet de simplifier le portage d'applications Apigraf vers Optima PLC. En mode normal, la valeur des canaux disponible dans l'application est exactement la valeur en virgule flottante renvoyée par le module. En mode de compatibilité avec Apigraf, la valeur est multipliée par un facteur dépendant de l'échelle choisie afin qu'il n'y ait pas de partie fractionnaire (les canaux analogiques étant renvoyés dans des valeurs entières dans Apigraf).

#### 9.1.13.2 Valeurs de statut

Statut	Description
-1	Driver non initialisé
-2	Driver stoppé en phase d'arrêt de l'application
0	Fonctionnement nominal
1, 241	Défaut de communication (pas de réponse)
2, 242	Défaut de communication
3, 243	Port COM non activé
4, 244	Adresse de module incorrecte
5, 245	Un seul module attendu sur le port COM (obsolète)
6, 246	Adresse de module dupliquée sur le port COM
7, 247	Incohérence de vitesse utilisée sur le port COM
255	Module non utilisé

## 9.2 Hilscher

### 9.2.1 Carte Hilscher CIF générique

Version 1.0



Ce driver permet la gestion par Optima PLC d'une carte Hilscher CIF générique.

Le principe d'utilisation est le suivant : dans un premier temps, la carte est configurée et testée avec SyCon (utilitaire de configuration Hilscher). La configuration ainsi obtenue est reportée puis utilisée dans Optima PLC en mode IO (c'est à dire en lecture et écriture directe dans la DPM (Dual Port Memory) de la carte).

#### 9.2.1.1 Installation

Les étapes suivantes doivent être suivies pour obtenir un fonctionnement correct de la carte Hilscher avec Optima PLC.

- Installation du driver Hilscher
- Installation du logiciel Hilscher de configuration SyCon
- Définition d'une configuration à l'aide de SyCon et téléchargement sur la carte. Le mode de handshake utilisé doit être "uncontrolled" avec cette version du driver.
- Test de la configuration avec SyCon (lecture des entrées, écriture des sorties)

Important : Optima PLC n'est pas en mesure d'exploiter la carte si le fonctionnement n'est pas déjà correct avec SyCon.

Les [étapes du paramétrage](#)<sup>419</sup> sont détaillées dans les paragraphes suivants.

### 9.2.1.2 Paramètres

Paramètres à entrer sous Optima PLC :

- Numéro de la carte

Jusqu'à quatre cartes Hilscher peuvent être installées. Préciser dans ce paramètre le numéro de la carte actuelle (1 à 4).

- Activation du Watch Dog

Si le watch dog est activé, le logiciel embarqué sur la carte coupe automatiquement les sorties en cas de défaillance du PC. Préciser Oui ou Non.

- Redémarrage automatique après défaut

Selon le paramétrage effectué dans SyCon, la carte coupe la communication lors d'un incident sur le réseau. Si ce paramètre est positionné sur Oui, Optima PLC provoque automatiquement un reset de la carte et donc un redémarrage de la communication si l'incident est réparé.

Si le paramètre est positionné sur Non, la communication n'est pas redémarrée automatiquement. Un reset de la carte doit être effectué. Ce reset peut être obtenu par différents moyens :

- Redémarrage de l'application (un reset est effectué au démarrage).
- Reset effectué dans SyCon (il est possible de lancer ce logiciel pendant le fonctionnement de l'application Optima PLC).
- Utiliser la librairie Hilscher\_CIF et la fonction CIF\_Reset ou CIF\_Reinit (appeler l'une de ces fonctions sur une condition déclenchée depuis une page de supervision par exemple).

### 9.2.1.3 Etapes du paramétrage

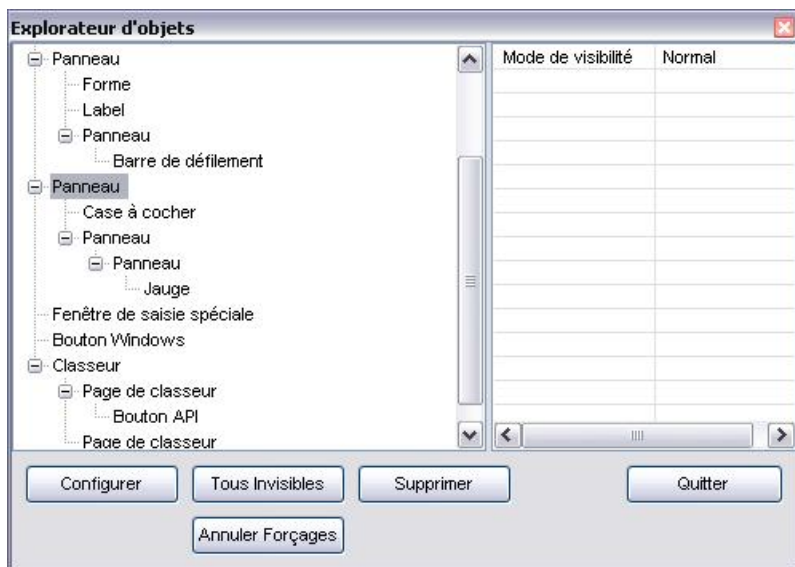
Voici un exemple montrant comment paramétrer rapidement un réseau de terrain avec une carte Hilscher et Optima PLC. Le réseau utilisé ici est Interbus, mais le processus est très proche pour les autres réseaux (Profibus, DeviceNet, etc...).

En premier lieu, utiliser le CD fourni par Hilscher pour installer le "Device driver" et l'utilitaire "SyCon". Le serveur OPC n'est pas requis par Optima PLC. Installer ensuite physiquement la carte dans le PC.

Dans le "menu démarrer" de Windows, appeler l'utilitaire de configuration du driver pour vérifier la détection de la carte.



La fenêtre suivante liste les cartes détectées.



## Configuration de SyCon

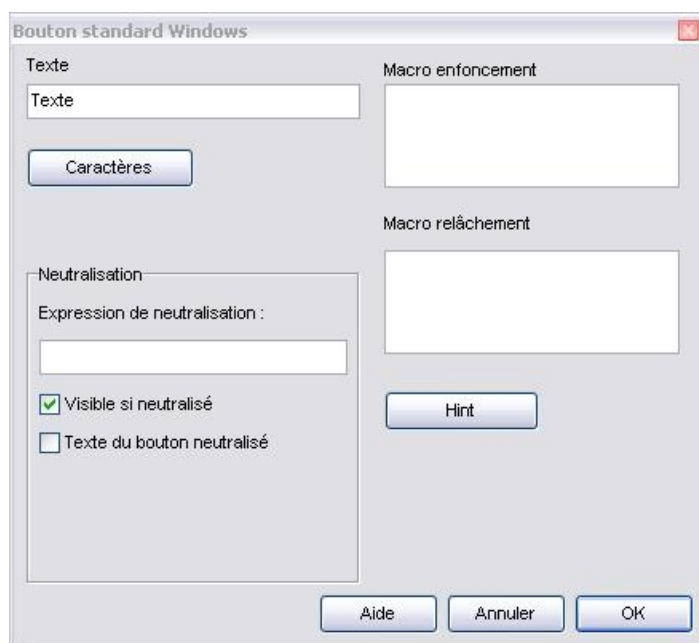
Dans le "menu démarrer", appeler le logiciel de configuration SyCon.



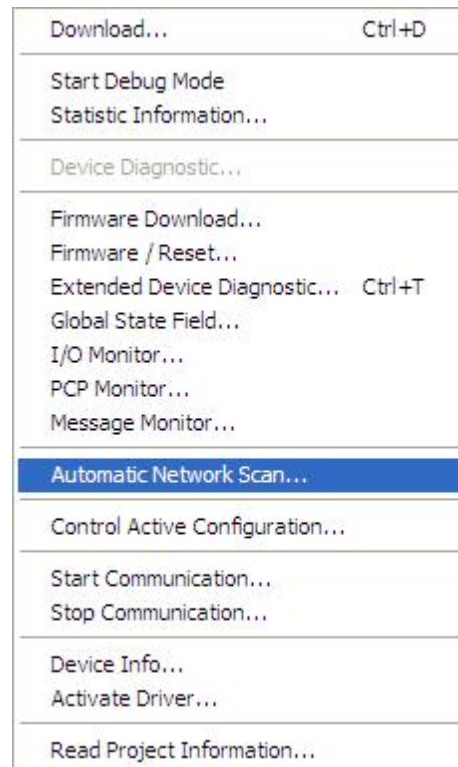
La fenêtre de base du logiciel s'affiche. Le projet est initialement vide. Cliquer sur l'icône suivante pour insérer un "maître" sur le réseau :



Choisir dans la liste le type de carte à utiliser. Dans l'exemple il s'agit d'une carte CIF50-IBM dédiée au protocole Interbus.



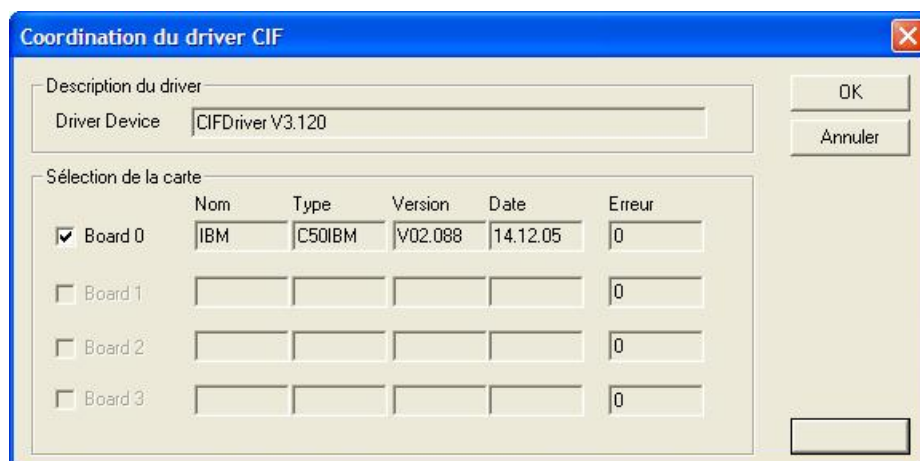
Dans le menu "Online", demander une scrutation automatique du réseau, afin de détecter la configuration actuelle.



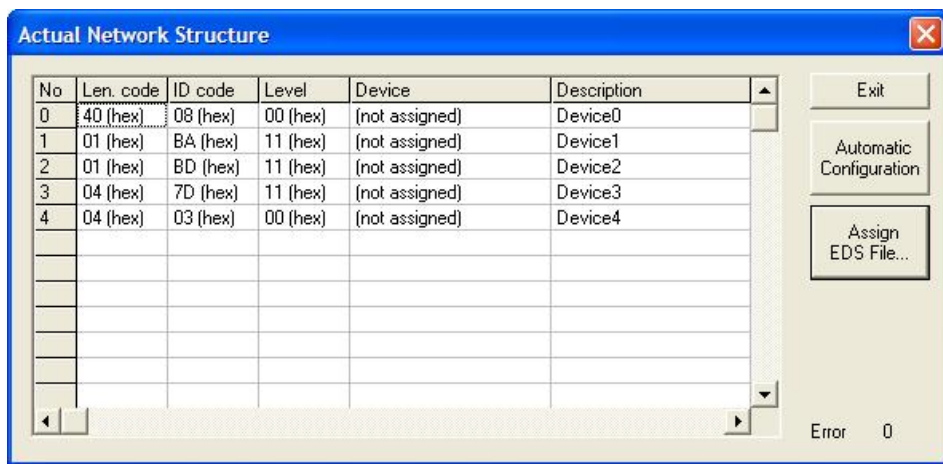
La scrutation doit être effectuée avec la carte CIF...



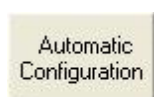
...sélectionnée :



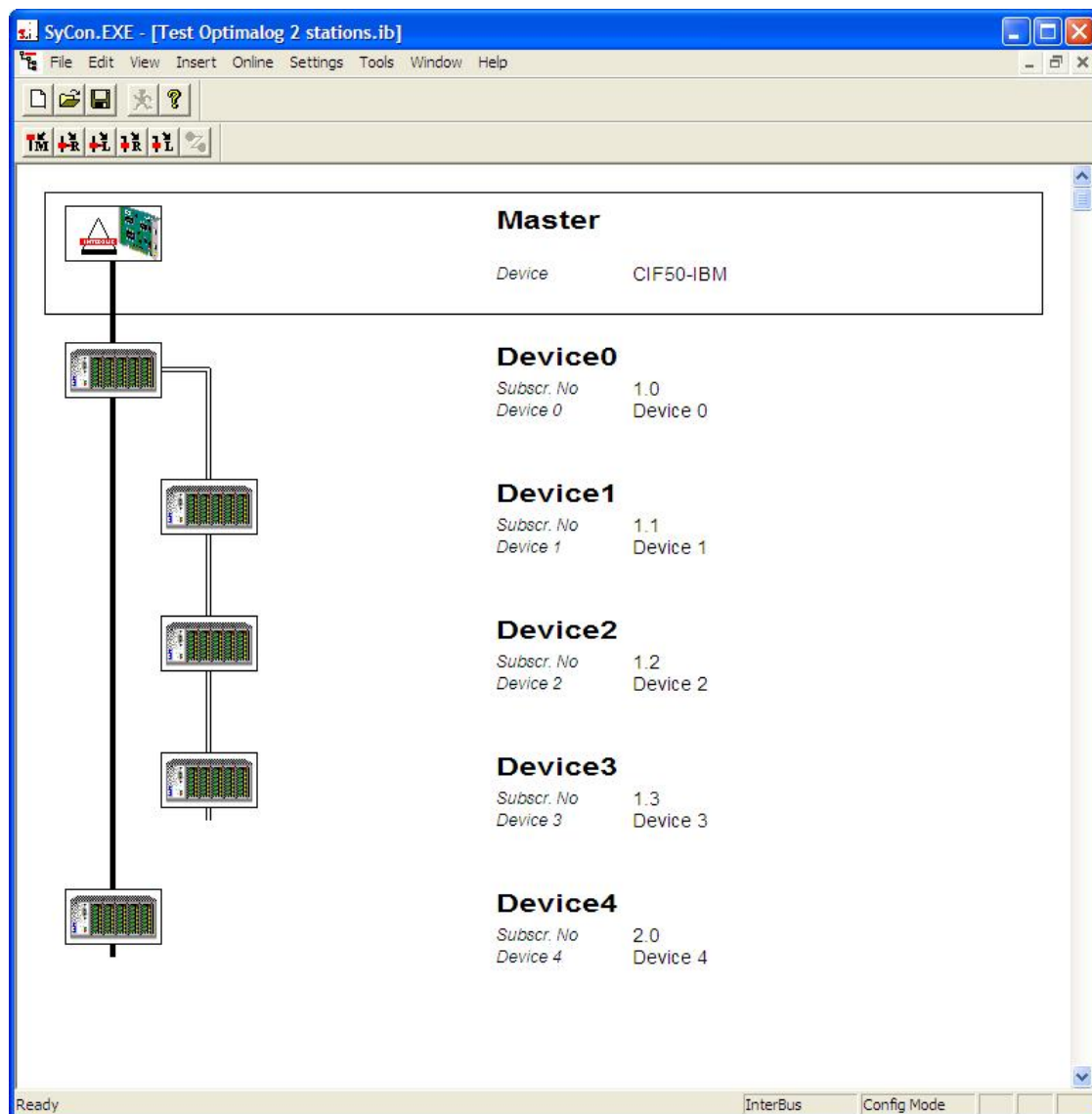
Voici les équipements Interbus détectés sur notre réseau :



En cliquant sur ce bouton...

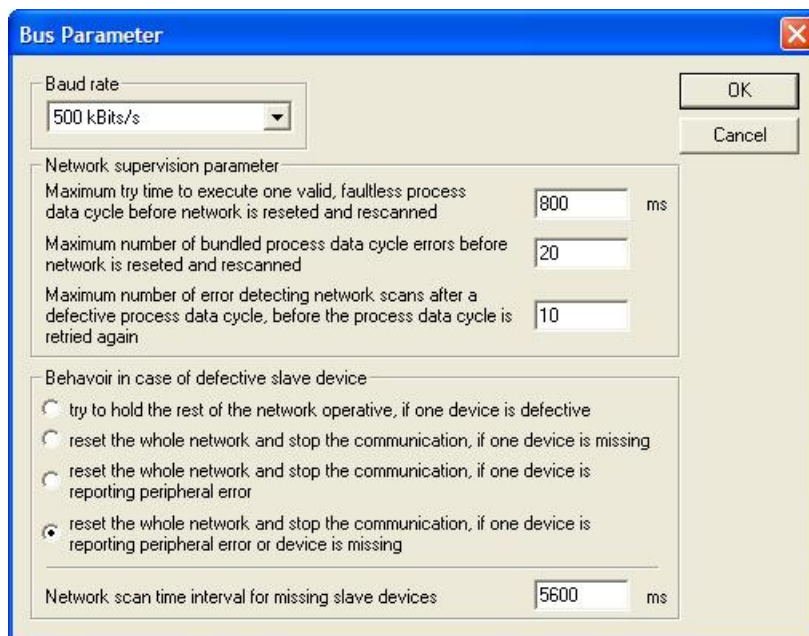


...SyCon prend automatiquement la configuration détectée comme étant la configuration du projet. Cette technique est très rapide mais il est également possible de configurer manuellement les équipements du réseau si nécessaire.

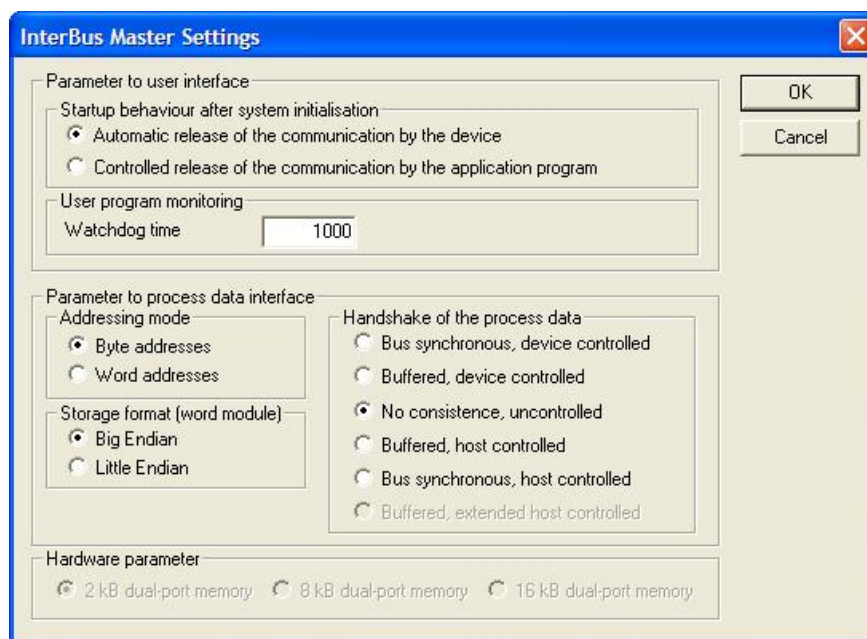


Dans le menu "Settings", paramétrer la communication et le maître. La configuration du bus de terrain dépend de celui-ci.





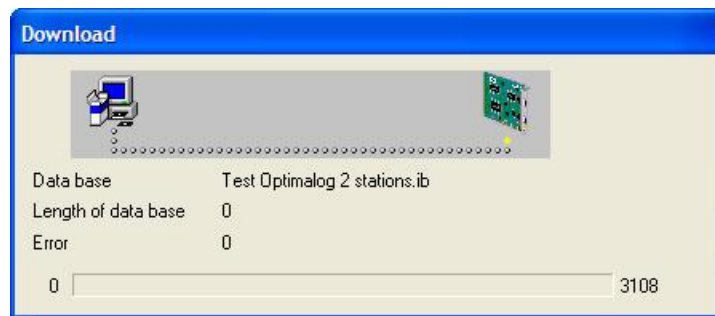
Configuration du maître (carte Hilscher). Le mode "no consistence, uncontrolled" est impératif avec la version actuelle du driver. Cela signifie qu'il n'y a pas de synchronisation dans la mise à jour des informations en entrée ou en sortie (voir documentation Hilscher).



Lorsque la configuration est terminée, celle-ci doit être téléchargée sur la carte (menu "Online").



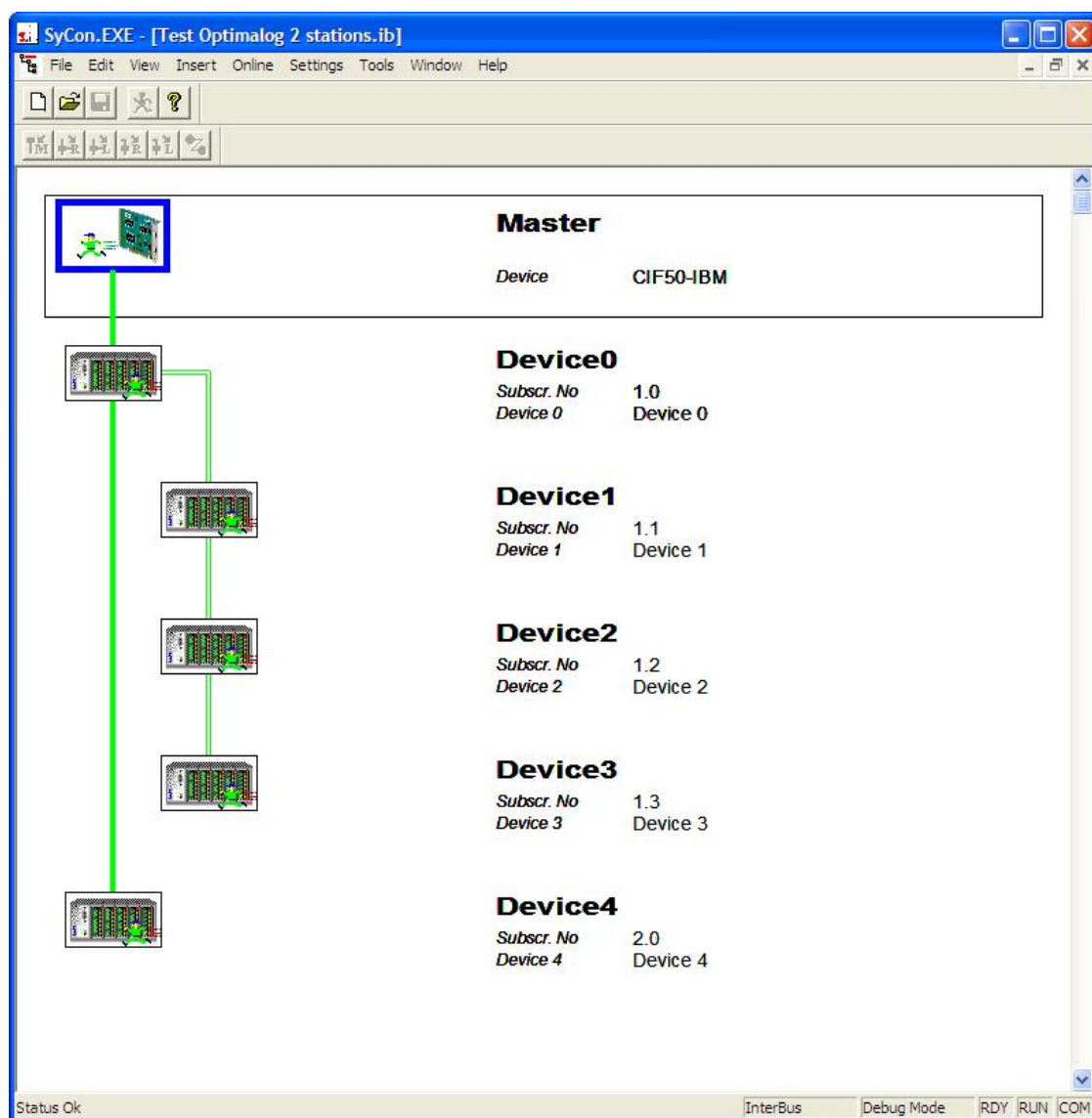
Le code de configuration est alors envoyé à la carte, qui le stocke en mémoire flash. Cette configuration est alors permanente.



Une fois le téléchargement terminé, si le paramètre "Automatic release of the communication by the device" a été sélectionné, la carte commence immédiatement à dialoguer. Sinon, le dialogue peut être démarré manuellement (menu "Online").



L'état du réseau peut aussi être testé en "Debug Mode" dans le menu "Online", en voici le résultat :



Lorsque la communication est démarrée (hors "Debug Mode"), des informations de diagnostic sont disponibles et permettent de contrôler le fonctionnement des équipements et de déceler l'origine d'une panne. Ci-dessous, l'affichage "Global state fields" indique l'état général de la carte et des équipements gérés (paramétrés, détectés et en défaut).

**Global State Field**

Online master main state: OPERATE

Collective status bits: I2ER I1ER NRDY EVE PRHL NEXC ACLR CTRL

Collective online error location and corresponding error:  
 Error at device address: 0  
 Corresponding error event: No actual error

Statistic bus information:  
 Number of defective process data cycles: 0 dec  
 Number of network reinitializations: 0 dec

Device specific status bits

Parameterized Devices    Activated Devices    Devices with Diagnostic

0	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	32	33	34	35	36	37	38	39	40	41
42	43	44	45	46	47	48	49	50	51	52	53	54	55
56	57	58	59	60	61	62	63	64	65	66	67	68	69
70	71	72	73	74	75	76	77	78	79	80	81	82	83
84	85	86	87	88	89	90	91	92	93	94	95	96	97
98	99	100	101	102	103	104	105	106	107	108	109	110	111
112	113	114	115	116	117	118	119	120	121	122	123	124	125
126													

Error: 0

L'option "IO Monitor" permet de tester l'état des entrées et des sorties. Les zones d'entrées et de sorties sont visualisées sous forme de bytes, les sorties peuvent être forcées.

**Moniteur d'E/S**

Données d'entrée

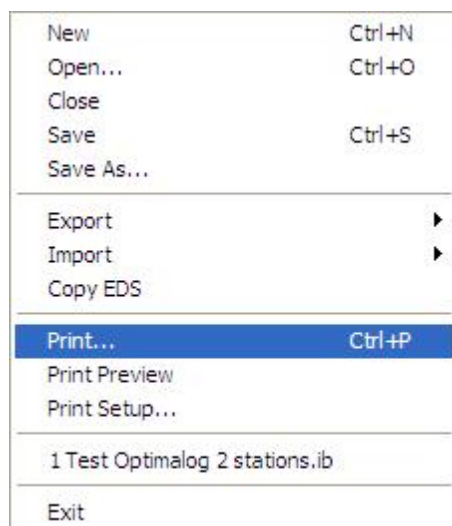
déc	0	1	2	3	4	5	6	7	8	9
0	255	128	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0
3	0	0								
4										
5										
6										
7										

Données de sortie

déc	0	1	2	3	4	5	6	7	8	9
0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0
3	0	0								
4										
5										
6										
7										

Erreur: 0

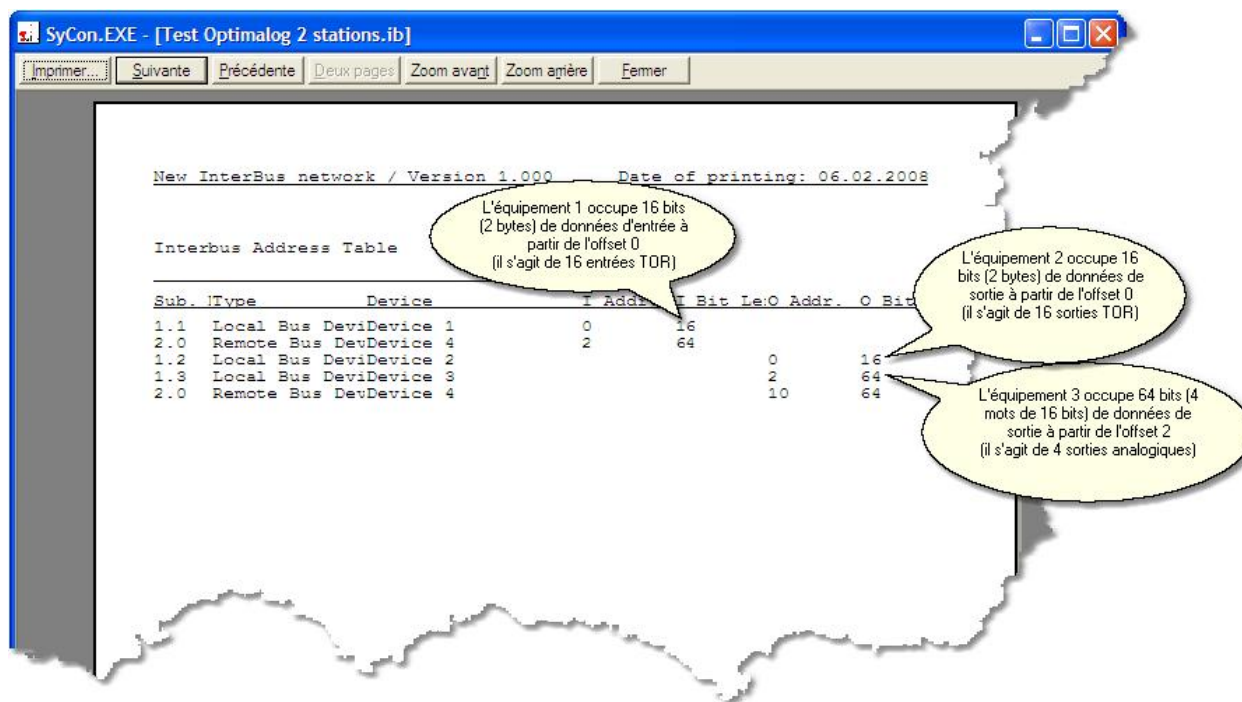
Lorsque le fonctionnement est satisfaisant à ce stade, il est possible de passer au paramétrage d'Optima PLC. Dans ce but, il est nécessaire d'obtenir la synthèse de la répartition des données de chaque équipement du réseau dans les zones d'entrée et de sortie de la carte. Pour cela, le plus simple est d'imprimer le rapport du projet :



La fenêtre suivante permet de sélectionner les informations à imprimer dans le rapport. Pour la configuration avec Optima PLC, c'est la partie "Address table" qui est intéressante ici :

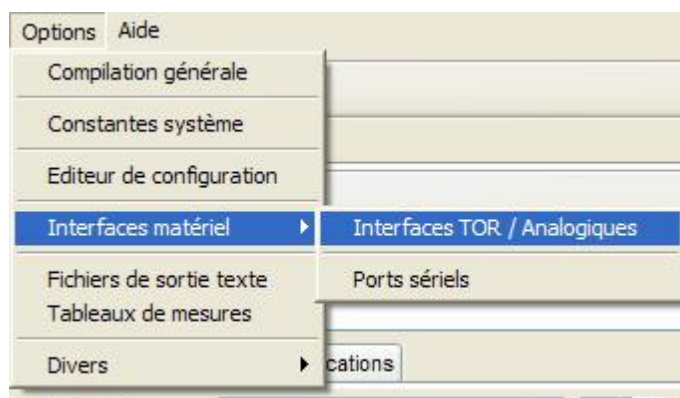


Voici la répartition des données obtenues pour la configuration de notre réseau Interbus. Pour chaque équipement, la taille des données en entrée ou en sortie, ainsi que leur position dans les tables de la carte, est indiquée. L'équipement 0 n'apparaît pas ici, car il ne comporte aucune entrée/sortie (il s'agit ici d'une "tête de station").

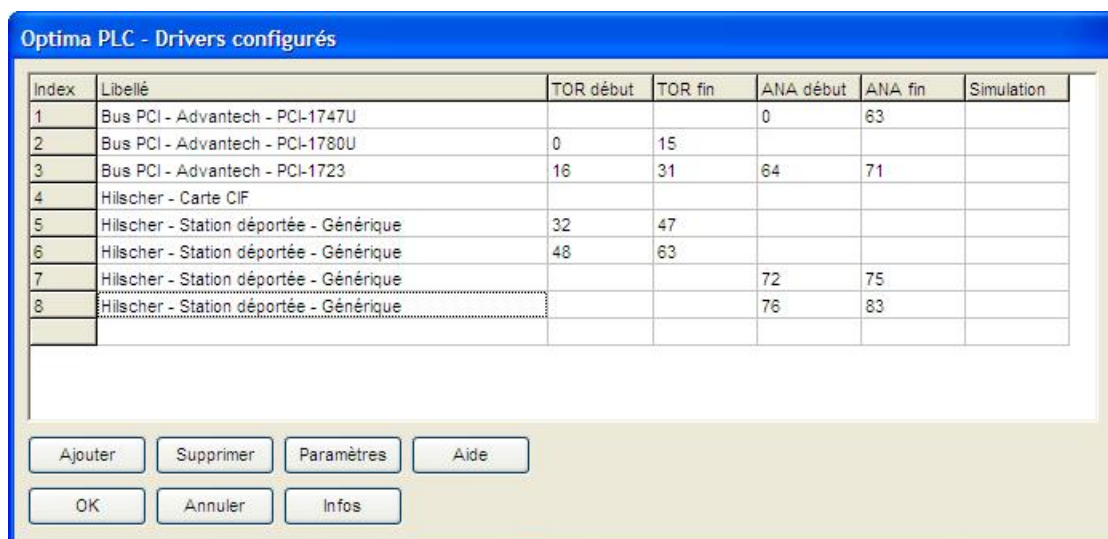


## Configuration d'Optima PLC

Dans l'atelier, appeler la configuration des interfaces.



Dans la liste des interfaces configurées, il faut définir une interface pour la carte CIF (sans entrée/sortie directement disponible), et une interface pour chaque équipement du réseau (ici les modules 1 à 4).



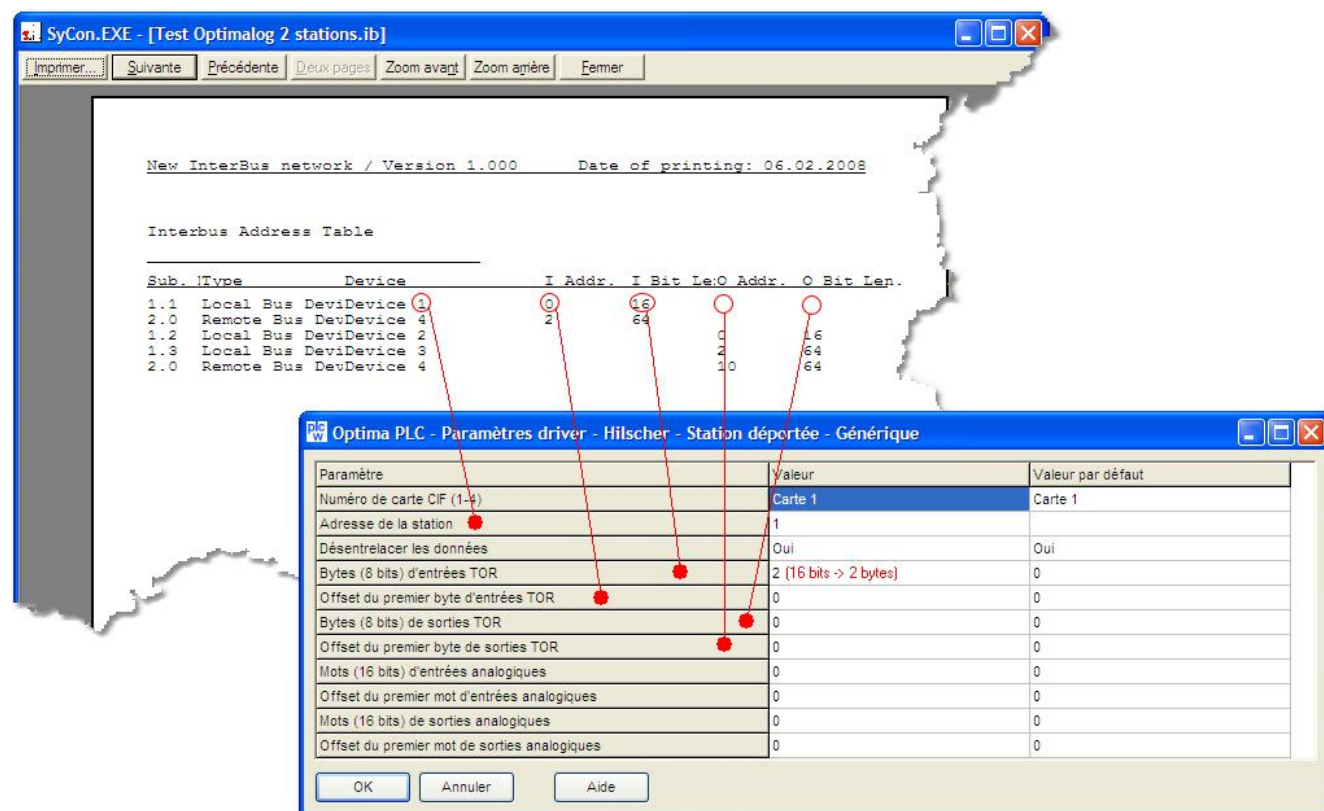
Configuration de la carte CIF :





Configuration des équipements.

Pour cette étape, il est nécessaire de reprendre le rapport de configuration généré par SyCon, en particulier la table des adresses. La transposition des données de chaque équipement est alors immédiate et systématique, comme ici pour le premier :



Procéder de la même manière pour chaque équipement de la liste :

Configuration de l'équipement 1 :



Configuration de l'équipement 2 :

Optima PLC - Paramètres driver - Hilscher - Station déportée - Générique

Paramètre	Valeur	Valeur par défaut
Numéro de carte CIF (1-4)	Carte 1	Carte 1
Adresse de la station	2	
Désentrelacer les données	Oui	Oui
Bytes (8 bits) d'entrées TOR	0	0
Offset du premier byte d'entrées TOR	0	0
Bytes (8 bits) de sorties TOR	2	0
Offset du premier byte de sorties TOR	0	0
Mots (16 bits) d'entrées analogiques	0	0
Offset du premier mot d'entrées analogiques	0	0
Mots (16 bits) de sorties analogiques	0	0
Offset du premier mot de sorties analogiques	0	0

OK Annuler Aide

Configuration de l'équipement 3 :

Optima PLC - Paramètres driver - Hilscher - Station déportée - Générique

Paramètre	Valeur	Valeur par défaut
Numéro de carte CIF (1-4)	Carte 1	Carte 1
Adresse de la station	3	
Désentrelacer les données	Oui	Oui
Bytes (8 bits) d'entrées TOR	0	0
Offset du premier byte d'entrées TOR	0	0
Bytes (8 bits) de sorties TOR	0	0
Offset du premier byte de sorties TOR	0	0
Mots (16 bits) d'entrées analogiques	0	0
Offset du premier mot d'entrées analogiques	0	0
Mots (16 bits) de sorties analogiques	4	0
Offset du premier mot de sorties analogiques	2	0

OK Annuler Aide

Configuration de l'équipement 4 :

Optima PLC - Paramètres driver - Hilscher - Station déportée - Générique

Paramètre	Valeur	Valeur par défaut
Numéro de carte CIF (1-4)	Carte 1	Carte 1
Adresse de la station	4	
Désentrelacer les données	Oui	Oui
Bytes (8 bits) d'entrées TOR	0	0
Offset du premier byte d'entrées TOR	0	0
Bytes (8 bits) de sorties TOR	0	0
Offset du premier byte de sorties TOR	0	0
Mots (16 bits) d'entrées analogiques	4	0
Offset du premier mot d'entrées analogiques	2	0
Mots (16 bits) de sorties analogiques	4	0
Offset du premier mot de sorties analogiques	10	0

OK Annuler Aide

#### 9.2.1.4 Valeurs de statut

Statut	Description
-1	Driver non initialisé
-2	Driver stoppé en phase d'arrêt de l'application



<b>0</b>	Fonctionnement nominal
<b>1</b>	Mauvais numéro de carte (doit être compris entre 1 et 4)
<b>2</b>	Carte déjà utilisée
<b>3</b>	Clé de registre Windows incorrecte concernant cette carte (driver non correctement installé)
<b>4</b>	Adresse incorrecte dans le registre Windows
<b>5</b>	Taille de carte incorrecte dans le registre Windows
<b>6</b>	Défaut interne de définition de données dans le driver Optima PLC
<b>7</b>	Carte non correctement identifiée
<b>8</b>	Allocation d'une zone d'accès à la carte impossible
<b>9</b>	Mode de Handshake incorrect (doit être Uncontrolled).
<b>10</b>	Carte non prête (Information Ready à 0).
<b>11</b>	Carte non en run (Information Run à 0).
<b>12</b>	Carte non en communication (Information COM à 0).

### 9.2.1.5 Stations

#### 9.2.1.5.1 Hilscher - Station déportées générique

Version 1.0

Ce driver est conçu pour une utilisation liée à celui de la [carte Hilscher CIF générique](#)<sup>[418]</sup>. Il permet de modéliser l'accès aux données d'entrée et de sortie d'une station via les zones de données correspondantes de la carte Hilscher.

##### 9.2.1.5.1.1 Paramètres

- Numéro de la carte CIF

Numéro de la carte à laquelle la station est rattachée

- Adresse de la station

Numéro d'adresse de la station sur le réseau

- Désentrelacer les données

Cette option permet de pallier le problème lié à l'ordre des bytes de données. Si l'ordre est : poids fort suivi de poids faible ("big endian"), les données doivent être désentrelacées. Le mode normal du PC est : poids faible suivi de poids fort ("little endian").

Les paramètres suivants doivent être entrés en fonction de la répartition des données de la station dans les zones mémoires de la carte Hilscher. Cette répartition peut être obtenue dans le logiciel de configuration SyCon, une fois le réseau correctement configuré.

- Bytes d'entrées TOR

Nombre de bytes de données d'entrées TOR (accessibles bits à bits dans Optima PLC).

- Offset du premier byte d'entrées TOR

Adresse du premier byte d'entrées TOR dans la zone d'entrée de la carte Hilscher.

- Bytes de sorties TOR

Nombre de bytes de données de sorties TOR (accessibles bits à bits dans Optima PLC).

- Offset du premier byte de sorties TOR

Adresse du premier byte de sorties TOR dans la zone d'entrée de la carte Hilscher.

- Mots d'entrées analogiques

Nombre de mots de 16 bits de données d'entrées (accessibles sous forme numérique dans Optima PLC).

- Offset du premier mot d'entrées analogiques

Adresse du premier mot d'entrées analogiques dans la zone d'entrée de la carte Hilscher.

- Mots de sorties analogiques

Nombre de mots de 16 bits de données de sorties (accessibles sous forme numérique dans Optima PLC).

- Offset du premier mot de sorties analogiques

Adresse du premier mot de sorties analogiques dans la zone d'entrée de la carte Hilscher.

#### 9.2.1.5.1.2 Valeurs de statut

Statut	Description
-1	Driver non initialisé
-2	Driver stoppé en phase d'arrêt de l'application
0	Fonctionnement nominal
1	Mauvais numéro de carte (doit être compris entre 1 et 4)
2	Carte non définie
3	Incohérence de définition des zones d'entrée ou de sortie
100	Station non définie dans le projet
101	Station définie dans le projet mais non détectée
102	Station en défaut non communiqué par la carte

#### 9.2.1.5.2 Hilscher - Station déportées 16l 16O

Version 1.0

Ce driver est identique à la station générique, mais se limite à la gestion de 16 bits d'entrées TOR et 16 bits de sorties TOR.

#### 9.2.2 Carte Hilscher CIF CAN Layer 2

Version 1.0



Ce driver permet la gestion par Optima PLC d'une carte Hilscher CIF CanOpen utilisée en mode transparent (Layer 2). Dans ce mode, l'application se charge de la gestion directe des messages CAN reçus et envoyés, qui ne sont donc pas gérés par la carte elle-même comme c'est le cas avec le protocole CanOpen ou DeviceNet.

L'installation et l'utilisation de la carte se font de manière similaire au cas de la [carte Hilscher générique](#)<sup>418</sup>. Les valeurs de statut sont également identiques.

L'envoi et la réception des messages CAN sont effectués via la librairie Hilscher\_CIF et les fonctions dédiées au CAN Layer 2.

#### 9.2.3 Carte Hilscher CIF 50-En Modbus/TCP mode IO (serveur)

Ce driver permet d'utiliser la carte Hilscher CIF en mode serveur, c'est à dire que la carte constitue un "esclave" Modbus/TCP répondant aux sollicitations de maîtres Modbus/TCP du réseau.

Le driver comporte 2048 entrées TOR et 2048 sorties TOR. Ces entrées/sorties correspondent aux variables que les maîtres Modbus/TCP peuvent venir lire et écrire de l'extérieur.

De même 256 canaux 16 bits en entrées et 256 canaux 16 bits en sorties sont disponibles, aux mêmes emplacements que les entrées/sorties TOR.

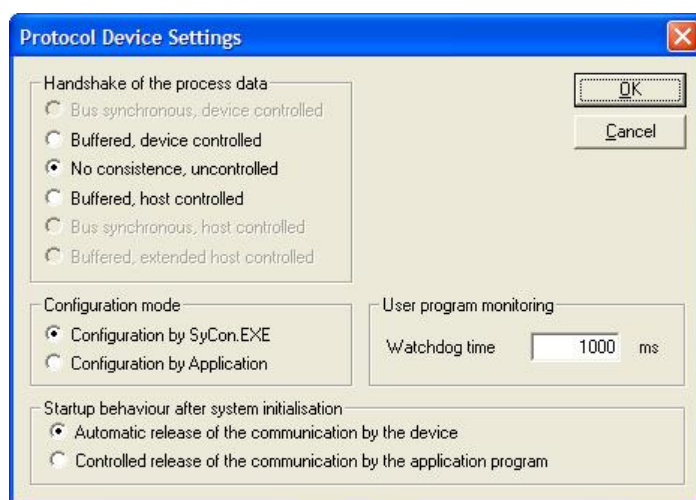
Répartition des indices de variables :

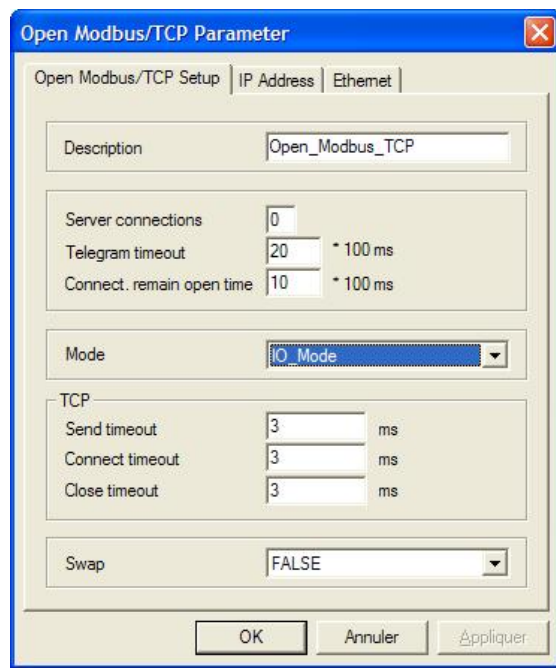
<b>Entrées TOR</b>	indices 0 à 2047
<b>Sorties TOR</b>	indices 2048 à 4095
<b>Entrées analogiques</b>	indices 0 à 255
<b>Sorties analogiques</b>	indice 256 à 511

### 9.2.3.1 Installation

Les étapes suivantes doivent être suivies pour obtenir un fonctionnement correct de la carte Hilscher avec Optima PLC.

- Installation du driver Hilscher
- Installation du logiciel Hilscher de configuration SyCon
- Définition d'une configuration à l'aide de SyCon et téléchargement sur la carte. Le mode de handshake utilisé doit être "uncontrolled" avec cette version du driver. Le mode fonctionnement Modbus/TCP doit être "IO mode" et non "Message mode".





- Test de la configuration avec SyCon (lecture des entrées, écriture des sorties)

Important : Optima PLC n'est pas en mesure d'exploiter la carte si le fonctionnement n'est pas déjà correct avec SyCon.

### 9.2.3.2 Paramètres

Paramètre à entrer sous Optima PLC :

- Numéro de la carte

Jusqu'à quatre cartes Hilscher peuvent être installées. Préciser dans ce paramètre le numéro de la carte actuelle (1 à 4).

### 9.2.3.3 Valeurs de statut

Statut	Description
-1	Driver non initialisé
-2	Driver stoppé en phase d'arrêt de l'application
0	Fonctionnement nominal
1	Mauvais numéro de carte CIF (doit être entre 1 et 4)
2	Carte déjà définie
3	Driver Hilscher non installé
4	Adresse de carte incorrecte dans le registre
5	Taille mémoire de la carte incorrecte
6	Version incorrecte du driver Optima PLC
7	Identificateur de carte incorrect
8	Impossible d'allouer la zone mémoire de la carte
9	Mode de Handshake mal configuré sur la carte (le mode doit être Uncontrolled, reconfigurer avec SyCon)
10	Type de carte incorrect
11	Modèle de carte incorrect

<b>12</b>	Mode de gestion Modbus incorrect (doit être configuré avec SyCon à IO mode)
<b>20</b>	Carte non prête
<b>21</b>	Carte non en mode Run
<b>22</b>	Carte non en état de communiquer

### 9.2.4 Carte Hilscher CIF 50-En Modbus/TCP mode message (client)

Ce driver permet d'utiliser la carte Hilscher CIF en mode client, c'est à dire que la carte est un maître Modbus/TCP.

Ce driver ne met à disposition aucune entrée/sortie. Son mode de fonctionnement est le suivant : lorsque l'option est proposée, la carte Hilscher est utilisée par les fonctions Modbus en lieu et place de l'interface Ethernet standard du PC.

Cette option est proposée par exemple dans le paramétrage de l'interface Modbus/TCP générique. Pour utiliser la carte Hilscher avec ce driver à la place de l'interface Ethernet standard, prenez soin de définir le driver Hilscher avant le driver Modbus/TCP générique dans la liste des drivers, et dans ce dernier de préciser le numéro de la carte Hilscher à utiliser, de 1 à 4.

Vous pouvez également utiliser en programmation les fonctions habituelles de la bibliothèque Modbus, en relation avec un canal Modbus obtenu par la fonction ModbusGetHilscherChannel.

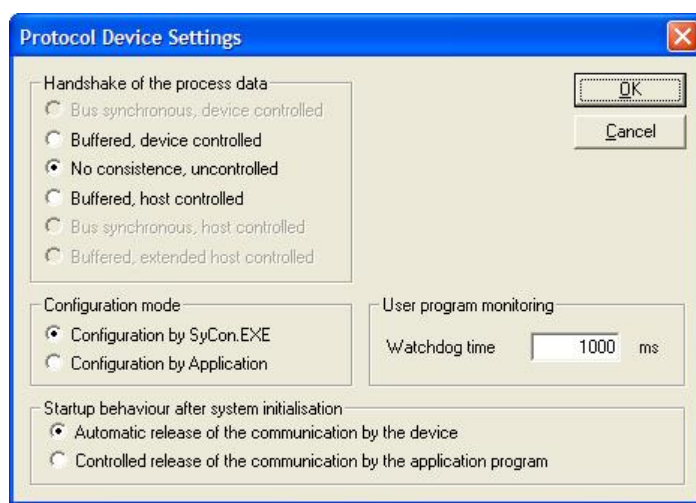
Attention, la carte Hilscher ne permet pas d'utiliser toutes les fonctions habituellement fournies par Optima PLC. Les fonctions Modbus suivantes uniquement sont disponibles : 1 à 7, 15 et 16.

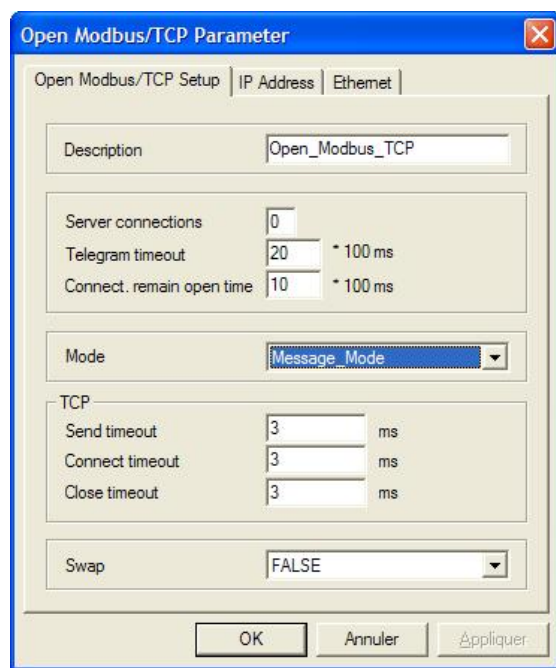
Dans ce mode de fonctionnement, la carte ne peut pas être utilisée en tant qu'esclave Modbus (serveur).

#### 9.2.4.1 Installation

Les étapes suivantes doivent être suivies pour obtenir un fonctionnement correct de la carte Hilscher avec Optima PLC.

- Installation du driver Hilscher
- Installation du logiciel Hilscher de configuration SyCon
- Définition d'une configuration à l'aide de SyCon et téléchargement sur la carte. Le mode de handshake utilisé doit être "uncontrolled" avec cette version du driver. Le mode fonctionnement Modbus/TCP doit être "Message mode" et non "IO mode".





- Test de la configuration avec SyCon (émission/réception de messages modbus).

Important : Optima PLC n'est pas en mesure d'exploiter la carte si le fonctionnement n'est pas déjà correct avec SyCon.

### 9.2.4.2 Paramètres

Paramètre à entrer sous Optima PLC :

- Numéro de la carte

Jusqu'à quatre cartes Hilscher peuvent être installées. Préciser dans ce paramètre le numéro de la carte actuelle (1 à 4).

### 9.2.4.3 Valeurs de statut

Statut	Description
-1	Driver non initialisé
-2	Driver stoppé en phase d'arrêt de l'application
0	Fonctionnement nominal
1	Mauvais numéro de carte CIF (doit être entre 1 et 4)
2	Carte déjà définie
3	Driver Hilscher non installé
4	Adresse de carte incorrecte dans le registre
5	Taille mémoire de la carte incorrecte
6	Version incorrecte du driver Optima PLC
7	Identificateur de carte incorrect
8	Impossible d'allouer la zone mémoire de la carte
9	Mode de Handshake mal configuré sur la carte (le mode doit être Uncontrolled, reconfigurer avec SyCon)
10	Type de carte incorrect
11	Modèle de carte incorrect

12	Mode de gestion Modbus incorrect (doit être configuré avec SyCon à Message mode)
20	Carte non prête
21	Carte non en mode Run
22	Carte non en état de communiquer

## 9.3 Heidenhain

### 9.3.1 Carte IK220

Version 1.0



Ce driver donne accès aux voies de comptage de la carte Heidenhain IK220.

Ce driver ne permet pas un accès en temps réel aux données de la carte, qui est pilotée via le driver fourni par le constructeur. De plus, cette structure particulière explique que le driver peut gérer jusqu'à 16 voies : en effet, jusqu'à 8 cartes IK220 possédant chacune 2 voies peuvent être utilisées simultanément. **Ce driver n'est donc à définir qu'une fois** dans la configuration Optima PLC pour utiliser de une à huit cartes.

#### 9.3.1.1 Installation

Installer en premier lieu le driver fourni par le constructeur, puis la ou les cartes dans le PC.

#### 9.3.1.2 Paramètres

Les paramètres à définir sont les suivants (communs à toutes les voies gérées) :

- Période des acquisitions

Ce paramètre correspond au temps séparant deux acquisitions de positions. Les valeurs possibles doivent être choisies dans la liste déroulante proposée.

- Nombre de valeurs dans tableau

Nombre maximum de valeurs pouvant être stockées dans les tableaux d'acquisition associés à chaque voie.

#### 9.3.1.3 Utilisation

##### Dans l'atelier de développement

Les entrées/sorties doivent être déclarées soit dans une partie "Variables globales". Les entrées analogiques sont de type LREAL.

##### Exemple

```
Channel1    at ANI0    : LREAL; // première entrée analogique
Channel2    at ANI1    : LREAL; // deuxième entrée analogique
```

Pour les utiliser dans le programme, il est possible d'utiliser la variable (ex : ANI0) ou son mnémonique (ex : Channel1).

Les valeurs sont renvoyées dans l'unité liée au matériel, soit par exemple le millième de mm.

## Dans le runtime Optima PLC

Les entrées/sorties sont visualisables dans le menu "Automate -> Variables -> Entrées analogiques ANI" et "Sorties analogiques ANO".

### Utilisation des tableaux de mesures

Les fonctions d'acquisitions dans des tableaux sont disponibles en utilisant la librairie IK220. Les données sont stockées dans ces tableaux à la fréquence d'acquisition demandée.

Les tableaux sont nommés de la manière suivante :

IK220\_<No de canal>

Ainsi, le canal 10 sera stocké dans le tableau IK220\_10.

De plus, le "temps" est stocké dans un tableau :

IK220\_Time

Les valeurs stockées sont les index d'acquisitions (0, 1, 2, etc.) et le coefficient est défini de manière à permettre un affichage en secondes.

#### 9.3.1.4 valeurs de statut

Statut	Description
-1	Driver non initialisé
-2	Driver stoppé en phase d'arrêt de l'application
0	Fonctionnement nominal
1	Redéfinition du driver IK220 (doit être utilisé une seule fois pour un maximum de 8 cartes)
2	Fichier IK220.dll non trouvé
3	Echec de l'initialisation du driver IK220

## 9.4 ICP DAS

### 9.4.1 Carte ICP DAS PCI-P16R16

Version 1.0



Ce driver donne accès aux 16 entrées isolées et 16 sorties relais de la carte ICP DAS PCI-P16R16.

Les canaux se répartissent ainsi :

Premier canal	Dernier canal	Utilisation
0	15	Entrées isolée de la carte
16	31	Sorties relais de la carte



### 9.4.1.1 Paramètres

- Numéro de la carte

Le système peut contenir plusieurs cartes PCI-P16R16. Ce paramètre permet de préciser laquelle est adressée ici. Le numéro de la première carte est 1. C'est la valeur par défaut.

### 9.4.1.2 Valeurs de statut (COPY)

Statut	Description
-1	Driver non initialisé
-2	Driver stoppé en phase d'arrêt de l'application
0	Fonctionnement nominal
1	Mauvais numéro de carte (doit être compris entre 1 et 16)
2	Carte non détectée

## 9.5 Phoenix Contact

### 9.5.1 Module Modbus/TCP type FL IL 24 BK-PAC

Ce driver permet de piloter des entrées/sorties PhoenixContact en utilisant une tête de station du type FL IL 24 BK-PAC.

La communication entre l'automate Optima PLC et le matériel Phoenix Contact est effectuée sur un bus Ethernet selon le protocole Modbus/TCP.

Ce driver autorise une configuration maximale de 256 DI, 256 DO, 32 AI et 32 AO.

Pour que l'automate puisse accéder aux entrées/sorties, la configuration matérielle devra être au préalable enregistrée dans la tête de station (avec l'utilitaire "Factory Line IO Configurator" de Phoenix Contact).

La répartition des entrées/sorties est la suivante :

	Optima PLC	Phoenix Contact
<b>Entrées</b>	I0 à I255	DI0 à DI255
<b>Sorties</b>	O256 à O511	DO0 à DO255
<b>Canaux d'entrées</b>	ANI0 à ANI31	AI0 à AI31
<b>Canaux de sorties</b>	ANO32 à ANO63	AO0 à AO31

### 9.5.1.1 Paramètres

Pour configurer le driver, le sélectionner dans la liste des drivers puis cliquer sur le bouton "Paramètres".

Il y a cinq paramètres mais seul le paramètre "Adresse IP" est obligatoire, les autres ont des valeurs par défaut :

- Adresse IP :

L'adresse IP doit être celle de la tête de station Phoenix Contact.

Elle est composée de 4 octets séparés par des points.

Cette adresse doit appartenir au sous-réseau de classe C du PC automate (Optima PLC).

#### Exemple

adresse de la tête de station : 192.189.207.60

adresse du PC avec Optima PLC : 192.189.207.193

Si aucune adresse IP n'est saisie, une erreur s'affiche au démarrage de l'application.

- Période de scrutation mini (en ms) :

Période à laquelle la lecture/écriture des entrées/sorties doit s'effectuer.

- Délai du chien de garde (en ms) :

Délai de déclenchement du chien de garde dans la station Phoenix Contact.  
Si ce délai a la valeur 0 alors le chien de garde est désactivé.

- Délai du monitoring (en ms) :

Délai de déclenchement du monitoring dans la station Phoenix Contact.  
Si ce délai a la valeur 0 alors le monitoring est désactivé.

- Affichage des erreurs Oui(1)/Non(0) :

Si ce paramètre vaut 1 alors des messages d'erreur apparaissent en cas d'erreur de communication.

### 9.5.1.2 Utilisation

#### Dans l'atelier de développement

Les entrées/sorties doivent être déclarées soit dans la partie "Variables globales" soit dans la partie "Déclaration" d'un bloc. Les entrées/sorties sont de type bool pour les digitales et de type LReal pour les analogiques.

#### Exemple

```
DItest      at I4      : bool;
DOtest      at O260    : bool:= false;
AItest      at ANI30   : LReal;
AOtest      at ANO32   : LReal:= 0.0;
```

Pour les utiliser dans le programme, il est possible d'utiliser la variable (ex : O260)

#### Exemple

```
DOtest := I4;
```

Le status de communication peut être testé :

Statut	Signification
<b>0</b>	pas d'erreur
<b>2</b>	Erreur trame, la réponse de l'esclave est un message d'erreur (trame envoyée par l'automate incorrecte).
<b>3</b>	Erreur timeout trame, la requête s'est terminée car le délai entre deux caractères reçus trop important).
<b>4</b>	Erreur timeout réponse, la requête s'est terminée car le délai sans réponse était trop important.
<b>7</b>	Erreur CRC, le contrôle de la trame reçue indique une erreur de transmission
<b>8</b>	Erreur timeout connexion, la connexion réseau n'a pas été établie ou a été interrompue

Un status d'erreur indique qu'une erreur est survenue mais ne stoppe pas la communication.

Par exemple, en cas d'erreur de connexion due à une station non alimentée, il suffit de mettre sous tension la station pour communiquer, le status repasse alors à 0 dès qu'un échange correct est réalisé.

Ce status peut être utilisé dans le programme grâce à la fonction IODriverStatus (librairie IODrivers).

#### Exemple

```
var
DOtest      at O260 : bool:= false;
```

```

    StopCycle      : bool:= false;
    TestStatus     : dint:= 0;
    DriverID       : int:= 0;
end_var;

.....

TestStatus:= IODriverStatus(DriverID);
if not StopCycle
    then DOTest:= true;
end_if;

if TestStatus > 0
    then StopCycle:= true;
        DOTest:= false;
end_if;

```

## Dans le runtime Optima PLC

Les entrées/sorties sont visualisables dans le menu "Automate - Variables".

## 9.6 Port I/O PC

### 9.6.1 Port I/O PC - 8 entrées TOR

*Version 1.0*

Cette interface permet l'utilisation de 8 entrées TOR accessibles par un port I/O du PC. Ceci autorise l'accès aux entrées de nombreuses cartes dont les interfaces ne sont pas disponibles ici.

Aucune initialisation de matériel n'est effectuée ici, l'interface se cantonne à la lecture du port. Les initialisations éventuellement nécessaires peuvent être faites avant le démarrage de l'application.

#### 9.6.1.1 Paramètres

Les paramètres à préciser :

- Adresse de base IO :

L'adresse du port contenant les 8 bits des entrées (consulter la documentation de la carte pour connaître cette adresse).

Frapper l'adresse précédée de 16# pour définir une adresse en hexadécimal.

- Masque d'inversion IO 0 à 7 :

Ce paramètre précise si l'état du port doit être inversé après lecture (0 : pas d'inversion, 1 : les 8 entrées sont inversées).

Frapper la valeur du masque précédée de 16# pour définir une valeur en hexadécimal.

#### 9.6.1.2 Utilisation

##### Dans l'atelier de développement

Les entrées doivent être déclarées soit dans la partie "Variables globales" soit dans la partie "Déclaration" d'un bloc. Les entrées sont de type bool.

##### Exemple

```

DITest      at I0      : bool;
SATest      at SA10    : bool;

```

Pour les utiliser dans le programme, il est possible d'utiliser la variable (ex : I0)

##### Exemple

```
SATest := I0;
```

## Dans le runtime Optima PLC

Les entrées sont visualisables dans le menu "Automate - Variables".

Un diagnostic est disponible dans le menu "Automate - Drives - Infos".

### 9.6.2 Port I/O PC - 8 sorties TOR

*Version 1.0*

Cette interface permet l'utilisation de 8 sorties TOR accessibles par un port I/O du PC. Ceci autorise l'accès aux sorties de nombreuses cartes dont les interfaces ne sont pas disponibles ici.

Aucune initialisation de matériel n'est effectuée ici, l'interface se cantonne à l'écriture du port. Les initialisations éventuellement nécessaires peuvent être faites avant le démarrage de l'application.

#### 9.6.2.1 Paramètres

Les paramètres à préciser sont :

- Adresse de base IO :

L'adresse du port contenant les 8 bits des sorties (consulter la documentation de la carte pour connaître cette adresse).

Frapper l'adresse précédée de 16# pour définir une adresse en hexadécimal.

- Masque d'inversion IO 0 à 7 :

Ce paramètre précise si l'état du port doit être inversé avant écriture (0 : pas d'inversion, 1 : les 8 sorties sont inversées).

Frapper la valeur du masque précédée de 16# pour définir une valeur en hexadécimal.

- Ecriture sur le port après chaque changement de la valeur de sortie Oui(1)/Non(0) :

Ce paramètre permet d'écrire la valeur après chaque changement ou lors de chaque scrutation.

#### 9.6.2.2 Utilisation

##### Dans l'atelier de développement

Les sorties doivent être déclarées soit dans la partie "Variables globales" soit dans la partie "Déclaration" d'un bloc. Les sorties sont de type bool.

##### Exemple

```
DOTest      at O2      : bool;
SATest      at SA10    : bool;
```

Pour les utiliser dans le programme, il est possible d'utiliser la variable (ex : O2)

##### Exemple

```
O2 := SATest;
```

## Dans le runtime Optima PLC

Les sorties sont visualisables dans le menu "Automate - Variables".

Un diagnostic est disponible dans le menu "Automate - Drives - Infos".

## 9.7 Simulation

### 9.7.1 Driver de simulation standard

Version 1.0

Ce driver permet de simuler des entrées/sorties TOR ou analogiques dans une application Optima PLC.

2048 E/S TOR peuvent être simulées, chaque voie étant accessible soit en entrée, soit en sortie. 256 canaux analogiques peuvent être également simulés.

Le principe est le suivant : au lieu que les états des E/S soient obtenus depuis un matériel, ils sont ici lus ou écrits dans des tables qui simulent ce matériel. Le fait de modifier ces tables, pour les entrées, a pour conséquence de modifier l'image mémoire de ces entrées pour Optima PLC. Pour les sorties, les tables permettent d'obtenir l'état électrique normalement commandé par l'application.

#### 9.7.1.1 Paramètre

Le driver ne comporte qu'un seul paramètre : le suffixe de nom d'accès à la variable de simulation (voir plus loin). Ce paramètre n'est utile que dans le cas où plusieurs drivers de simulation sont utilisés, afin de les différencier. Dans le cas contraire il peut être laissé vide (cas par défaut).

#### 9.7.1.2 Utilisation dans la supervision

L'accès par la " supervision " recouvre la supervision standard d'Optima PLC, ainsi que l'accès via le serveur OPC.

Les données de simulation peuvent être utilisées dans la supervision de plusieurs manières.

#### Accès absolu

L'accès se fait via la variable " Sim " suivie du type de variable attendu.

Si un suffixe a été défini dans le paramètre du driver, celui-ci est ajouté à la fin de " Sim ". Par exemple, si le suffixe est " P1 ", la variable à utiliser est " SimP1 ".

<b>Entrées TOR</b>	Sim.I0 à Sim.I2047
<b>Sorties TOR</b>	Sim.O0 à Sim.O2047

Les valeurs des index peuvent être variables :

Sim.I[R1]

Dans ce cas, la variable de simulation d'entrée dont l'index est contenu dans R1 est utilisée.

#### Attention

Pour que la voie I, O, ou An corresponde à la même voie dans la numérotation globale d'Optima PLC, il faut que le driver soit placé en premier dans la liste des drivers. En effet, la voie Sim.I10 correspond à la 10ème voie dans le driver. Si la première E/S gérée par le driver est la numéro 64, par exemple, Sim.I10 correspond en fait à la voie 74 dans la numérotation globale d'Optima PLC.

#### Accès par mnémonique

Si des E/S sont déclarées dans le programme, ils peuvent également être utilisés via leur mnémonique.

Exemple

```
E_BoutonDepartCycle at I10 : bool ; // bouton départ cycle
S_VoyantCycleEnCours at O23 : bool ; // voyant indiquant le cycle en cours
```

Ces deux E/S peuvent être simulées en utilisant les écritures suivantes :

Sim\_E\_BoutonDepartCycle

### 9.7.1.3 Accès par zone de mémoire partagée

Une application tierce peut utiliser les données de simulation en accédant au " fichier mappé en mémoire " nommé " OptimaPLCSim " (plus le suffixe éventuellement précisé dans le paramètre).

La structure de la zone mémoire ainsi accessible est la suivante :

<b>Entrées TOR</b>	2048 bytes	Chaque byte vaut 0 ou 1
<b>Sorties TOR</b>	2048 bytes	Chaque byte vaut 0 ou 1
<b>Canaux analogiques</b>	256 valeurs réelles	Chaque canal est codé sur 80 bits (format IEEE standard 754)

La taille totale de la zone partagée est donc de 6656 bytes.

Note : l'accès à cette zone se fait normalement avec l'API Win32, au moyen des fonctions CreateFileMapping, OpenFileMapping et MapViewOfFile.

## 9.7.2 Driver de simulation Solid Dynamics Motion

*Version 2.0*

Ce driver permet de simuler des entrées/sorties TOR ou analogiques dans une application Optima PLC. Il est plus particulièrement dédié à la communication avec le logiciel Motion de la société Solid Dynamics.

2048 E/S TOR peuvent être simulées, chaque voie étant accessible soit en entrée, soit en sortie. 256 canaux analogiques peuvent être également simulés.

Le principe est le suivant : au lieu que les états des E/S soient obtenus depuis un matériel, ils sont ici lus ou écrits dans des tables qui simulent ce matériel. Le fait de modifier ces tables, pour les entrées, a pour conséquence de modifier l'image mémoire de ces entrées pour Optima PLC. Pour les sorties, les tables permettent d'obtenir l'état électrique normalement commandé par l'application.

### 9.7.2.1 Paramètres

Le driver comporte deux paramètres : ce sont les noms des deux fichiers d'échange de données mappés en mémoire, utilisés pour la communication avec les autres applications, en particulier Motion.

Le premier paramètre correspond au fichier d'entrée d'Optima PLC, le second au fichier de sortie.

### 9.7.2.2 Utilisation dans la supervision

L'accès par la " supervision " recouvre la supervision standard d'Optima PLC, ainsi que l'accès via le serveur OPC.

L'accès aux données de simulation se fait via les variables " SimMotionOptimaPLCInput " et " SimMotionOptimaPLCOutput ", suivies du type de variable attendu.

Si un nom de fichier mappé différent du nom par défaut a été utilisé, le nom de la variable est constitué du préfixe " Sim " suivi du nom du fichier.

<b>Entrées TOR</b>	SimMotionOptimaPLCInput.D0 à SimMotionOptimaPLC.D2047
<b>Sorties TOR</b>	SimMotionOptimaPLCOutput.D0 à SimMotionOptimaPLCOutput.D2047
<b>Entrées analogiques</b>	SimMotionOptimaPLCInput.A0 à SimMotionOptimaPLCInput.A255
<b>Sorties analogiques</b>	SimMotionOptimaPLCOutput.A0 à SimMotionOptimaPLCOutput.A255

Les valeurs des index peuvent être variables :

SimMotionOptimaPLCInput.D[R1]

Dans ce cas, la variable de simulation d'entrée dont l'index est contenu dans R1 est utilisée.

### Attention

Pour que la voie D ou A corresponde à la même voie dans la numérotation globale d'Optima PLC, il faut que le driver soit placé en premier dans la liste des drivers. En effet, la voie SimMotionOptimaPLCInput.D10 correspond à la 10ème voie dans le driver. Si la première E/S gérée par le driver est la numéro 64, par exemple, D10 correspond en fait à la voie 74 dans la numérotation globale d'Optima PLC.

### 9.7.2.3 Accès par zone de mémoire partagée

Une application comme Motion peut utiliser les données de simulation en accédant aux " fichiers mappés en mémoire " nommés par défaut " MotionOptimaPLCInput " et " MotionOptimaPLCOutput ".

La structure des zones mémoire ainsi accessibles est la suivante :

Taille du segment	4 bytes	Taille du segment sur 32 bits
Canaux TOR	2048 doubles	Chaque valeur vaut 0 ou 1
Canaux analogiques	256 doubles	Valeurs réelles

Chaque valeur " double " est codée sur 64 bits au format de réel IEEE standard 754.

La taille totale de chaque zone partagée est donc normalement de 18436 bytes.

Note : l'accès à ces zones se fait normalement avec l'API Win32, au moyen des fonctions CreateFileMapping, OpenFileMapping et MapViewOfFile.

Note 2 : dans cette version, aucun mécanisme de synchronisation n'est prévu, la consistance des données n'est donc pas assurée.

## 9.8 Wago

### 9.8.1 Module Modbus/TCP type 750-341, 342, 352, 841, 842

Version 1.1

Ce driver permet de piloter des entrées/sorties WAGO en utilisant une tête de station du type 750-XXX.

La communication entre l'automate Optima PLC et le matériel WAGO est effectuée sur un bus Ethernet selon le protocole Modbus UDP (idem Modbus TCP, mais basé sur UDP).

Ce driver autorise une configuration maximale de 256 DI, 256 DO

Pour que l'automate puisse accéder aux entrées/sorties, la configuration réseau (adresse IP) devra être au préalable enregistrée dans la tête de station (avec l'utilitaire "BootP server" de WAGO).

En cas de problème de fonctionnement, l'onglet "Diagnostics" des informations du module, accessible dans le runtime, permet d'évaluer la situation. Si nécessaires, ces informations peuvent être copiées (via le presse-papier) dans un email envoyé à Optimalog pour une aide au diagnostic.

#### 9.8.1.1 Paramètres

Pour configurer le driver, le sélectionner dans la liste des drivers puis cliquer sur le bouton "Paramètres".

Il y a cinq paramètres mais seul le paramètre "Adresse IP" est obligatoire, les autres ont des valeurs par défaut :

- Adresse IP :

L'adresse IP doit être celle de la tête de station WAGO.  
Elle est composée de 4 octets séparés par des points.

Cette adresse doit appartenir au sous-réseau de classe C du PC automate (Optima PLC).

**exemple**

adresse de la tête de station : 192.189.207.60  
adresse du PC avec Optima PLC : 192.189.207.193

Si aucune adresse IP n'est saisie, une erreur s'affiche au démarrage de l'application.

- Mode de communication

Indiquer si la communication est faite en UDP (télégrammes, mode sans connexion) ou en TCP (mode avec connexion et gestion de flux). Le mode UDP est plus souple et permet une meilleure gestion des timeouts.

- Timeout de réémission

Temps en microsecondes au bout duquel une commande est renvoyée si aucune réponse n'est reçue.

- Timeout de reset

Temps en microsecondes au bout duquel la connexion est réinitialisée si aucune réponse n'est reçue.

- Période de scrutation mini (en ms) :

Période à laquelle la lecture/écriture des entrées/sorties doit s'effectuer. Pour une scrutation à la vitesse maximale, mettre 0.

- Délai du chien de garde (en ms) :

Délai de déclenchement du chien de garde dans la station Wago. La valeur entrée est multipliée par 100 ms. Si ce délai a la valeur 0 alors le chien de garde est désactivé.

- Verrouillage de l'adresse IP (oui : 1/non : 0) :

Indique si l'adresse IP spécifiée doit être verrouillée dans la station WAGO.

- Affichage des erreurs (oui : 1/non : 0) :

Si ce paramètre vaut 1 alors des messages d'erreur apparaissent en cas d'erreur de communication.

- Pilotage des sorties :

Deux modes sont possibles.

"Automatique est le mode par défaut. Les sorties sont actualisées dès le démarrage de l'application, avec un état initial à 0.

"Sur commande soft" : dans ce mode, lors du démarrage de l'application, seules les entrées sont actualisées. Les sorties ne sont pas envoyées au module Wago. Le démarrage et l'arrêt de cette actualisation sont commandés par l'appel de fonctions de la librairie IODrivers :

IODriverWagoStartUpdateOutputs démarre l'actualisation des sorties.

IODriverWagoStopUpdateOutputs arrête l'actualisation des sorties.

Ce dernier mode permet un arrêt puis un redémarrage de l'application sans affecter l'état des sorties du module Wago, ce qui peut être indispensable dans certaines applications, comme par exemple la GTB. Attention, pour cela le chien de garde doit être désactivé (voir paramètre plus haut).

### 9.8.1.2 Utilisation

#### Dans l'atelier de développement

Les entrées/sorties doivent être déclarées soit dans la partie "Variables globales" soit dans la partie "Déclaration"



d'un bloc. Les entrées/sorties sont de type bool pour les digitales et de type lreal pour les analogiques.

### Exemple

```
DITest    at I4      : bool;
DOTest    at O260    : bool:= false;
AITest    at ANI30   : lreal;
AOTest    at ANO32   : lreal:= 0.0;
```

Pour les utiliser dans le programme, il est possible d'utiliser la variable (ex : O260).

### Exemple

```
DOTest:= I4;
```

## Dans le runtime Optima PLC

Les entrées/sorties sont visualisables dans le menu "Automate - Variables".

### 9.8.1.3 Valeurs de statut

Statut	Description
<b>-1</b>	Driver non initialisé
<b>-2</b>	Driver stoppé en phase d'arrêt de l'application
<b>0</b>	Fonctionnement nominal
<b>2</b>	Erreur de trame modbus
<b>3</b>	Timeout au cours d'une trame modbus
<b>4</b>	Timeout de réception de réponse à une requête modbus
<b>7</b>	Erreur de CRC dans une réponse modbus
<b>8</b>	Pas de connexion avec le module
<b>51</b>	Erreur interne : initialisation modbus impossible
<b>52</b>	Adresse IP incorrecte ou non spécifiée
<b>60</b>	Erreur de configuration : plus de variables d'entrées ou de sortie sont utilisées que n'en comporte le module Wago.
<b>100</b>	Transmission en cours de la configuration au module Wago

## 9.9 Réseau Ethernet standard

### 9.9.1 Module Modbus/TCP générique

Ce driver permet d'utiliser de manière standard les entrées/sorties d'un équipement répondant au protocole Modbus/TCP ou Modbus/UDP. Le PC joue le rôle de client.

La répartition des entrées et sorties se fait de la manière suivante :

<b>Entrées TOR</b>	indices 0 à 1023
<b>Sorties TOR</b>	indices 1024 à 2047
<b>Entrées analogiques</b>	indices 0 à 63
<b>Sorties analogiques</b>	indice 64 à 127

#### 9.9.1.1 Paramètres

- Adresse IP

Adresse IP permettant d'atteindre l'équipement distant.

- Numéro de port de communication

Numéro du port TCP ou UDP utilisé pour la communication (normalement 502).

- Mode TCP ou UDP

Sur Ethernet, le protocole Modbus peut être basé sur TCP (liaison avec établissement d'une connexion) ou sur UDP (échange de datagrammes sans notion de connexion).

Lorsque cela est possible, préférer le mode UDP, qui est plus léger et permet à Optima PLC une plus grande maîtrise des timings. Attention, ce mode UDP n'est pas autorisé avec une carte Hilscher.

- Numéro de carte Hilscher

Si la communication est gérée par l'une des 4 cartes Hilscher CIF présente dans le PC (il doit s'agir d'une carte CIF 50-EN), préciser son numéro ici, sinon laisser 0.

Le driver correspondant à la carte Hilscher CIF 50-EN en mode master doit être précédemment configuré dans la liste des drivers.

Attention, ce paramètre n'est pas pris en compte si le mode UDP est précédemment sélectionné.

- Période de scrutation

Ce paramètre n'est pas géré actuellement. Laisser la valeur 0.

- Affichage des erreurs

Si la valeur Oui ou 1 est précisée, un défaut de communication sera signalé à l'affichage. Sinon, un tel défaut est signalé uniquement dans le journal de bord.

- Fonction Modbus de lecture des entrées TOR

Ce paramètre permet d'ajuster la fonction utilisée aux possibilités de l'équipement. Les fonctions FC1 à FC4 sont autorisées. L'utilisation des fonctions FC3 ou FC4 est optimal.

- Nombre d'entrées TOR disponibles sur l'équipement

Indiquer ici le nombre d'entrées TOR disponibles sur l'équipement. ce nombre (128 maximum) indique combien d'entrées seront lues à chaque cycle par le driver.

- Adresse de la première entrée TOR

Adresse de la première entrée à lire (dans l'espace des mots 16 bits).

- Fonction Modbus d'écriture des sorties TOR

Ce paramètre permet d'ajuster la fonction utilisée aux possibilités de l'équipement. Les fonctions FC15 et FC16 sont autorisées. L'utilisation de la fonction FC16 est optimal.

- Nombre de sorties TOR disponibles sur l'équipement

Indiquer ici le nombre de sorties TOR disponibles sur l'équipement. ce nombre (128 maximum) indique combien de sorties seront écrites à chaque cycle par le driver.

- Adresse de la première sortie TOR

Adresse de la première sortie TOR à écrire (dans l'espace des mots 16 bits).

- Fonction Modbus de lecture des entrées analogiques

Ce paramètre permet d'ajuster la fonction utilisée aux possibilités de l'équipement. Les fonctions FC3 et FC4 sont autorisées.

- Nombre d'entrées analogiques disponibles sur l'équipement

Indiquer ici le nombre d'entrées analogiques (ou de compteurs) disponibles sur l'équipement. Ce nombre (64 maximum) indique combien d'entrées seront lues à chaque cycle par le driver.

- Adresse de la première entrée analogique

Adresse de la première entrée analogique à lire (dans l'espace des mots 16 bits).

- Fonction Modbus d'écriture des sorties analogiques

Seule la fonction FC16 est autorisée ici.

- Nombre de sorties analogiques disponibles sur l'équipement

Indiquer ici le nombre de sorties analogiques (ou de compteurs) disponibles sur l'équipement. Ce nombre (64 maximum) indique combien de sorties seront écrites à chaque cycle par le driver.

- Adresse de la première sortie analogique

Adresse de la première sortie analogique à écrire (dans l'espace des mots 16 bits).

### Précisions concernant les nombres d'entrées et de sorties.

Le driver met en place des trames de lecture et écriture en fonction des paramètres mentionnant les nombres et adresses des entrées et des sorties. Il y a donc au maximum 4 trames définies (entrées TOR, sorties TOR, entrées analogiques, sorties analogiques).

Le temps de cycle total étant fonction du nombre de trames, il est préférable de laisser à 0 un nombre de variables si celles-ci ne sont pas utilisées dans l'application. Par exemple, si l'application n'utilise pas de sorties analogiques bien qu'il y en ait de disponibles, laisser à 0 le nombre de sorties analogiques évitera la création d'une trame inutile et réduira le temps de scrutation des variables utiles.

#### 9.9.1.2 Valeurs de statut

Statut	Description
<b>-1</b>	Driver non initialisé
<b>-2</b>	Driver stoppé en phase d'arrêt de l'application
<b>0</b>	Fonctionnement nominal
<b>2</b>	Erreur de trame modbus
<b>3</b>	Timeout au cours d'une trame modbus
<b>4</b>	Timeout de réception de réponse à une requête modbus
<b>7</b>	Erreur de CRC dans une réponse modbus
<b>8</b>	Pas de connexion avec le module
<b>51</b>	Erreur interne : initialisation modbus impossible
<b>52</b>	Adresse IP incorrecte ou non spécifiée
<b>60</b>	Erreur de configuration : plus de variables d'entrées ou de sortie sont utilisées que n'en comporte le module, d'après les paramètres entrés.

## 10 Questions fréquentes

Les questions les plus fréquemment posées sont repertoriées ici avec leur réponse.

## 10.1 Questions générales

### 10.1.1 Pourquoi utiliser le PC comme automate ?

Avec un PC, vous disposez d'une interface opérateur conviviale permettant de suivre le process en cours, très simple à développer.

Il vous donne également des capacités de stockage et d'archivage requises lorsqu'une traçabilité est nécessaire.

Le PC est une architecture ouverte et standard. Vous n'êtes pas contraint d'utiliser du matériel propriétaire et pouvez mettre en concurrence les fournisseurs de PC, d'autant plus que l'offre du marché est très large.

Enfin, l'utilisation d'une même plate-forme pour le développement et l'exécution de l'application procure un confort certain pour la mise au point, la mise en service, la maintenance.

### 10.1.2 Optima PLC permet-il de programmer un automate du marché ?

Non, Optima PLC est prévu et optimisé pour fonctionner uniquement sur une base PC.

De futurs développements sont cependant prévus pour permettre la génération de code portable (C et C++), et s'exécutant sur d'autres OS que Windows.

### 10.1.3 Optima PLC est-il un outil de simulation ?

Optima PLC peut être utilisé à des fins de simulation et de formation ; en particulier, le fait qu'il fonctionne sur PC facilite énormément son intégration au sein d'un ensemble d'autres logiciels.

Mais Optima PLC est avant tout un vrai système d'automate sur PC, destiné à un usage industriel.

### 10.1.4 Questions concernant le temps réel

L'aspect temps réel d'Optima PLC est un point important du produit, et génère de nombreuses questions.

#### 10.1.4.1 Optima PLC est-il un système temps-réel ?

Oui, le runtime Optima PLC utilise pour son exécution un noyau temps-réel conçu et développé spécifiquement par OptimaLog, pour obtenir sous Windows une régularité et un déterminisme optimisé et adapté à l'automatisme.

Les performances obtenues permettent de travailler avec une période minimale de 1 ms, et un jitter maximum de l'ordre de 20 µs.

Contrairement à certains produits concurrents qui se prétendent à tort temps-réel, Optima PLC se base donc bien sur une couche logicielle spécifique qui prend le pas sur le noyau de Windows, et non sur le mécanisme multi-tâche de Windows lui-même.

Cependant, comme avec tous les systèmes temps-réel s'exécutant conjointement à Windows, il ne rend pas Windows temps-réel. Il n'est par exemple pas possible d'appeler des fonctions de l'API Windows depuis les tâches temps-réel d'Optima PLC.

#### 10.1.4.2 Optima PLC peut-il fonctionner sans le noyau temps-réel ?

Oui, le noyau temps-réel est débrayable. Son utilisation est l'une des options de compilation.

Le noyau temps-réel est d'ailleurs désactivé par défaut.

#### 10.1.4.3 Pourquoi le noyau temps-réel est-il désactivé par défaut ?

Le fonctionnement du noyau temps-réel nécessite des ressources (mémoire en particulier) assez importantes. Celles-ci ne sont pas toujours disponibles sur certains PC :

- PC ancien
- PC utilisé pour développer, mais également pour d'autres usages
- PC avec peu de ressource
- PC utilisé pour évaluer Optima PLC, mais non dédié au fonctionnement de l'application

Pour éviter tout problème, le noyau temps-réel est désactivé par défaut à partir de la version 2.5.

#### 10.1.4.4 Optima PLC nécessite-t-il du matériel supplémentaire pour fonctionner en temps réel ?

Non, Optima PLC utilise uniquement les ressources d'un PC standard.

#### 10.1.4.5 Je pensais qu'il était possible de faire du temps-réel avec Windows, je ne vois pas ce qu'Optima PLC apporte de plus ?

Windows n'est absolument pas conçu au départ pour faire du "vrai" temps réel. Si les performances obtenues peuvent s'avérer excellentes, la structure même de Windows ne permet pas de garantir la constance de celles-ci avec une application classique.

Le fait que Windows soit un OS multitâche préemptif, condition effectivement nécessaire pour faire du temps réel, n'est pas suffisant. D'autres conditions sont requises, concernant la gestion des tâches et leur commutation, la structure des drivers, la gestion de la mémoire paginée, etc...

La confusion est souvent entretenue par certains acteurs du marché qui tentent de faire accréditer l'idée que leur produit garantit des performances temps réel. Elle l'est également par certains éléments de Windows lui-même : par exemple, le niveau de priorité le plus élevé qui puisse être attribué à une tâche s'appelle "TpTimeCritical", alors que, comme pour toutes les tâches exécutées au niveau utilisateur sous Windows, il ne saurait être question de véritable temps réel.

Cette impossibilité de travailler en temps réel sous Windows explique l'apparition d'un marché parallèle de systèmes véritablement temps réel, basés sur des solutions uniquement logicielles ou bien intégrant du matériel supplémentaire, et capables de s'intégrer plus ou moins à Windows.

#### 10.1.4.6 Pourquoi ne pas utiliser plutôt un système temps-réel du marché ?

Si vous souhaitez réaliser des applications d'automatisme "classiques", Optima PLC offre certains avantages par rapport aux systèmes temps-réel proposés par ailleurs :

- Optima PLC est un outil dédié aux automaticiens, il ne requiert pas la compétence pointue d'ingénieur en développement spécialisé dans les systèmes temps-réels. Une formation courte ou une autoformation suffisent pour sa prise en main.
- Optima PLC est un système léger, simple et facile à installer et à utiliser.
- Optima PLC s'avère bien plus économique à l'achat et à l'utilisation qu'un système temps-réel, certes puissant, mais surdimensionné et trop complexe pour des applications d'automatisme, bien qu'il soit possible de concevoir avec cet outil des applications habituellement cantonnées aux systèmes temps réels complexes.

#### 10.1.4.7 Optima PLC permet-il de faire du temps réel avec tous les PC ?

En règle générale, il n'y a pas de problème avec les PC standards **monoprocesseurs**. Cependant, certains PC comportent des fonctions installées qui peuvent perturber les fonctions temps-réel d'Optima PLC. D'autres part certains PC multiprocesseurs ne permettent pas à l'heure actuelle de faire fonctionner correctement le noyau temps réel, bien que ce problème reste marginal.

Les problèmes viennent généralement d'un ou plusieurs drivers de matériel installés sous Windows, sans lien avec Optima PLC, et "masquant" (désactivant) parfois les interruptions du système pendant un temps trop important. On observe alors une augmentation du "jitter" (c'est à dire de la variation) du cadencement des tâches de l'automate.

Ce problème est commun à tous les systèmes temps-réels fonctionnant conjointement à Windows actuellement sur le marché. Il nécessite d'effectuer quelques tests pour mesurer le jitter maximum observé. Dans la plupart des cas, celui-ci reste cependant inférieur à 20 µs.

D'autre part, le noyau temps-réel d'Optima PLC nécessite des ressources importantes en mémoire, ce qui peut constituer parfois un problème sur certains PC affectés à d'autres usages, et faisant fonctionner beaucoup d'autres applications.

Optima PLC permet de visualiser les performances obtenues, en particulier les variations sur la période d'exécution des tâches de l'automate. En cas de perturbation trop importante, un message d'avertissement est affiché pendant l'exécution du runtime.

#### 10.1.4.8 Optima PLC exploite-t-il les ressources des PC multiprocesseurs ?

Le mode multiprocesseur est effectivement exploité, mais différemment selon que le noyau temps réel est activé ou non.

Si le noyau temps réel est activé : toutes les tâches temps réel sont exécutées sur un même processeur. Les tâches gérées au niveau Windows (priorités 8 et 9, affichages) sont exécutées indifféremment sur chaque processeur.

Si le noyau temps réel est désactivé : toutes les tâches sont gérées au niveau Windows, et peuvent s'exécuter indifféremment sur chaque processeur.

#### **10.1.4.9 Optima PLC permet-il de faire du temps-réel avec tous les matériels interfacés ?**

Non, les performances ne sont pas les mêmes avec tous les matériels.

Pour créer l'interface avec un matériel, deux solutions sont possibles dans le cadre d'Optima PLC :

##### **Utilisation de drivers constructeurs**

Ces drivers sont relativement simples à interfacier avec Optima PLC, mais leur emploi impose l'utilisation de l'API de Windows, avec laquelle il n'est pas possible de faire du vrai temps réel : il est impossible pour Optima PLC de garantir un temps de réaction donné car le séquençement est directement géré par Windows.

Les drivers actuellement basés sur une connexion réseau TCP/IP avec Ethernet utilisent la couche réseau de Windows et ne procurent pas de véritables performances temps-réel.

Les performances atteintes par ce type de driver sont cependant très optimisées dans Optima PLC. Le temps de cycle (temps séparant deux actualisations des entrées/sorties) typique est d'environ 10 millisecondes, avec des variations maximales de quelques millisecondes.

##### **Utilisation de drivers spécifiques à Optima PLC**

Les drivers développés par Optimalog permettent généralement d'accéder directement au matériel et de l'utiliser avec les performances maximales possibles. C'est généralement le cas des cartes ou des équipements dont les spécifications d'accès direct au matériel sont documentés par les constructeurs, comme par exemple Hilscher.

Les performances de ces drivers peuvent atteindre le maximum autorisé par Optima PLC en terme de temps de cycle, c'est à dire normalement la milliseconde. Ces performances sont cependant également dépendantes du matériel employé.

#### **10.1.4.10 Quelles performances obtient-on sans le noyau temps-réel ?**

La période de base est généralement de la milliseconde, mais peut varier jusqu'à 3 millisecondes : la régularité est donc beaucoup moins bonne qu'avec le noyau temps-réel. Cependant, la période obtenue est tout à fait suffisante pour faire fonctionner de nombreuses applications.

#### **10.1.5 Est-il possible de passer automatiquement d'un langage de la norme à un autre ?**

Non, et cette possibilité n'a jamais été envisagée par les concepteurs de la norme contrairement à un mythe tenace.

Une certaine portabilité serait envisageable si une très petite part des possibilités de programmation était utilisée dans chaque langage, mais au pris de nombreuses difficultés. Ce n'est pas à l'heure actuelle une priorité dans Optima PLC.

#### **10.1.6 Est-il possible de modifier une application sans l'arrêter ?**

Oui, Optima PLC permet de faire des modifications importantes dans une application et de les mettre en ligne avec un temps d'interruption inférieur à la milliseconde.

Pour plus d'informations, voir Modifications "en ligne".

#### **10.1.7 J'ai déjà travaillé avec Apigraf, puis-je porter facilement un projet Apigraf sous Optima PLC ?**

Si les deux outils sont basés sur les mêmes principes généraux de fonctionnement, quelques différences font que le portage d'Apigraf vers Optima PLC n'est pas une opération immédiate. Celle-ci reste toutefois très simple.

Le langage Apigraf de base (combinatoire) existe également dans Optima PLC, cependant certains ajustements doivent être effectués manuellement (voir [Différences avec le langage Apigraf origine](#)<sup>[385]</sup>).

- Les déclarations, en particulier les définitions de mnémoniques, doivent être redéfinies conformément à la norme IEC 61131-3.
- Les modules logiciels existants dans Apigraf n'ont pas tous leur équivalent dans Optima PLC.
- Certaines différences mineures, dans les types de variables par exemple, doivent être prises en compte pour le portage.
- La partie IHM (interface homme/machine) doit être refaite dans Optima PLC.

### 10.1.8 Où puis-je télécharger la dernière version d'Optima PLC ?

Vous pouvez actualiser Optima PLC à l'emplacement suivant :

#### Site web

<http://www.optimalog.com/downloadoplc.html>

#### Site FTP OptimaLog

<ftp://optimaplc@optimalog.serveftp.net>

### 10.1.9 Optima PLC respecte-t-il la norme IEC 61131-3 ?

Optima PLC est l'un des seuls logiciels de sa catégorie à respecter l'ensemble des définitions de la norme et ses préconisations.

## 10.2 Questions techniques

### 10.2.1 Matériel

#### 10.2.1.1 Quelle configuration est nécessaire pour faire fonctionner Optima PLC ?

La configuration suivante est conseillée pour faire fonctionner le runtime Optima PLC.

PC équipé d'un Pentium IV, 2 GHz minimum,  
Windows XP Service Pack 2 ou Windows Vista  
Une prise USB minimum (clé de protection)  
512 Mo de RAM minimum  
Système de fichiers NTFS conseillé

L'atelier de développement peut quant à lui fonctionner sans problème sur une plateforme Windows XP SP2 ou Vista standard.

#### 10.2.1.2 Que peut-il arriver si je n'ai pas assez de mémoire sur le PC ?

Pour obtenir des performances temps-réel lors de l'exécution du runtime, Optima PLC doit verrouiller une grosse quantité de mémoire physique. Si cette mémoire physique est insuffisante, cela peut perturber les performances temps-réel et dans certains cas provoquer un crash de Windows.

Lors du démarrage, le runtime Optima PLC indique dans le fichier journal (Events.log) si la quantité de RAM est insuffisante pour faire fonctionner l'application.

### 10.2.2 Comment faire...

#### 10.2.2.1 Comment obtenir des informations sur la licence d'utilisation d'Optima PLC ?

Dans l'atelier ou dans le runtime Optima PLC, le premier menu comporte une option "Licence" qui provoque l'affichage d'une page d'information sur la licence d'utilisation en cours.

#### 10.2.2.2 Comment limiter le risque que l'utilisateur corrompe les données ou le système sous Windows ?

Lorsqu'un PC est mis en production avec une application fonctionnant sous Windows, le risque est grand que certains utilisateurs aillent modifier des données ou des paramètres de fonctionnement du système de manière inconsidérée.

Pour éviter ce risque, il suffit que le système fonctionne dans une session "non administrateur", disposant de droits limités. Cependant l'application ne peut s'exécuter dans ce type de session, car l'accès aux ressources du PC lui est alors interdit.

Pour pallier ce problème, il suffit de paramétrer l'application pour qu'elle s'exécute dans le contexte d'un utilisateur disposant des droits d'administrateur. Pour faire ce paramétrage, voir [Session Windows pour exécution](#)<sup>[32]</sup> et [Paramétrage des sessions Windows](#)<sup>[33]</sup>.

### 10.2.2.3 Comment empêcher l'utilisateur de modifier les paramètres de fonctionnement dans le runtime ?

Vous devez pour cela activer le mode "[gestion des utilisateurs](#)"<sup>[329]</sup> dans le runtime.

### 10.2.2.4 Comment utiliser des temporisations dans Optima PLC ?

Les différentes possibilités sont décrites dans le paragraphe [Techniques de gestion du temps](#)<sup>[387]</sup>.

### 10.2.2.5 Comment faire afficher des messages d'avertissement pour l'utilisateur ?

Voir la bibliothèque de fonctions "Messages", qui contient des fonctions permettant d'afficher des messages d'information, d'erreur, des demandes de confirmation... Voir également la documentation de programmation avancée.

### 10.2.2.6 Comment forcer un diagramme SFC à son étape initiale ?

Voir le bloc ResetSFC.

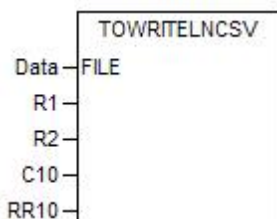
A noter que cette possibilité n'est pas envisagée dans la norme.

### 10.2.2.7 Comment écrire des données dans un fichier pour les relire avec un tableur ?

Cette opération est très simple.

- Déclarer le fichier dans le menu Options / Fichiers de sortie texte.
- Charger la librairie TextOut dans les librairies standards (onglet "Librairies").
- Dans le programme, utiliser les fonctions TOWriteLnCSV (pour générer un fichier avec données séparées par des virgules) ou TOWriteLnTab (pour générer un fichier avec données séparées par des tabulations).

Dans l'exemple ci-dessous, on écrit dans le fichier Data (le nom réel sur disque est défini dans Options / Fichiers de sortie texte) les valeurs des variables numériques entières R1 et R2, de la chaîne de caractères C10 (la valeur sera encadrée de caractères "), et de la variable réelle RR10. Les valeurs sont ici séparées par des virgules (format CSV), un format TSV (tabulations) est également disponible avec TOWriteLnTab.



### 10.2.2.8 Comment saisir des valeurs de paramètres sur mon écran tactile ?

Voir l'option "[clavier virtuel automatique](#)"<sup>[334]</sup> du runtime, qui permet de faire apparaître automatiquement un clavier sur l'écran lorsqu'une saisie est nécessaire.

### 10.2.2.9 Comment faire démarrer mon application automatiquement au démarrage du PC ?

Voir l'option "[démarrage automatique avec le PC](#)"<sup>[334]</sup> du runtime.

### 10.2.2.10 Comment faire communiquer mon application Optima PLC avec une autre application ?

Le problème de la communication avec un autre programme (communication inter-processus) peut être abordé de plusieurs manières différentes.



### Communication via un socket TCP/IP

C'est la méthode généralement retenue pour la communication entre les applications Optimalog. Dans Optima PLC, la librairie TCP fournit toutes les fonctions nécessaires.

- Avantages  
Cette méthode fonctionne même si les applications ne sont pas sur le même PC.  
Elle permet d'envoyer des volumes de données importants.
- Inconvénients  
Elle est un peu plus complexe à programmer, car il faut tenir compte de l'établissement de la communication.  
Elle ne permet pas une communication depuis les tâches temps-réel.

### Communication via le registre Windows

Cette méthode consiste à utiliser les fonctions de la librairie WinRegistry pour lire et écrire des données dans le registre Windows.

- Avantages  
Méthode très simple à mettre en oeuvre.
- Inconvénients  
Les programmes doivent absolument fonctionner sur le même PC.  
Ne permet pas d'échanger de gros volumes de données.  
Elle ne permet pas une communication depuis les tâches temps-réel.

### Communication par zone de données partagée

Cette méthode est basée sur la possibilité de partager une même zone de donnée mémoire du PC entre plusieurs applications Windows. Elle fait appel à la librairie SharedMemory.

- Avantages  
Facilité d'utilisation dans Optima PLC  
Zone de donnée partagée pouvant être assez grande (éviter toutefois de dépasser le Mo).  
Communication instantanée  
Une fois l'initialisation de la zone effectuée, l'accès aux données peut se faire depuis les tâches temps-réel.
- Inconvénients  
Les programmes doivent absolument fonctionner sur le même PC.  
Les applications ne disposent pas toujours de la possibilité d'utiliser cette technique.  
Attention à la consistance des données.

#### 10.2.2.11 Comment faire afficher simplement des messages de défaut ?

Pour obtenir un affichage de défauts de manière très simple, utiliser les flags d'alarmes prédéfinies A :

- Définir un mnémonique et un commentaire pour chaque variable A utilisée, décrivant chacune un défaut.
- Dans le programme, mettre à 1 ou à 0 la variable de défaut
- Configurer l'affichage des alarmes dans le runtime.

Autre solution pour un traitement plus élaboré des alarmes : utiliser le logiciel *Optim'Alarm*, spécialement conçu dans ce but et parfaitement intégré à Optima PLC.

#### 10.2.2.12 Comment configurer la session utilisateur pour exécuter une application Optima PLC

Pour s'exécuter, une application Optima PLC a besoin des droits d'administration maximum sur le PC. En effet, l'application requiert l'accès à certaines ressources du PC qui nécessitent ces droits (gestion de certains drivers et du noyau temps réel).

L'application doit donc s'exécuter dans une session utilisateur disposant des droits d'administrateur et des privilèges maximum. En outre, à partir de la version Vista de Windows, une autorisation explicite d'élévation des droits est demandée en standard, ce qui peut poser problème pour exécuter l'application automatiquement au démarrage du PC.

Configurer correctement le démarrage automatique tout en conservant un niveau de sécurité correct n'est donc pas simple, car plusieurs situations peuvent se rencontrer.

[Windows XP](#) 

[Windows Vista et versions suivantes](#) 

### Configuration conseillée

Nous vous conseillons de démarrer votre PC dans une session aux droits limités, de conserver l'UAC active (à partir de Vista), et d'utiliser le compte Administrateur pour l'exécution de l'application Optima PLC (afin d'éviter le problème de l'élévation à partir de Vista).

Ceci peut être configuré dans l'atelier logiciel, menu Options, Divers, Session Windows pour exécution. Entrez à cet endroit le login souhaité (Administrateur) et le mot de passe correspondant (il est ensuite conservé sous forme cryptée). Ces informations sont utilisées lorsqu'un démarrage automatique est configuré depuis le runtime.

Le paramétrage des comptes des utilisateurs peut être également effectué dans l'atelier logiciel, menu Options, Divers, Paramétrage des sessions Windows. Il est possible avec cette fonction d'activer le compte Administrateur et de lui affecter un mot de passe.

Attention : les informations de compte utilisateur définies ici sont également utilisées pour le lancement automatique d'autres application liées à Optima PLC et requérant des droits d'administrateur : Optima Update, Optima Lock, Optima Tracks, Optim'Alarm...

Voir aussi l'option "[démarrage automatique avec le PC](#)"  du runtime.

#### 10.2.2.12.1 Configuration des sessions sous Windows XP

La situation sous XP est relativement simple, puisque deux situations seulement peuvent se rencontrer.

##### La session ouverte dispose des droits d'administration

Dans ce cas, l'application Optima PLC peut être lancée directement sans difficulté. Attention cependant : bien que cette situation soit très courante sous Windows XP, elle n'est pas recommandée pour des raisons de sécurité.

##### La session ouverte dispose de droits limités

Ceci est la situation de travail recommandée.

L'application Optima PLC doit être lancée avec les droits d'un autre compte utilisateur, disposant des droits d'administration et d'un mot de passe.

#### 10.2.2.12.2 Configuration des sessions sous Windows Vista jusqu'à Windows 10

A partir de Vista, lorsqu'un programme nécessitant des droits d'administration étendus est lancé, le comportement du système peut varier :

- Si la session actuelle dispose des droits d'administration, une confirmation d'élévation des privilèges est demandée, sauf si ce mécanisme (UAC, User Account Control) a été désactivé. En outre, le compte Administrateur est particulier puisqu'il ne nécessite pas d'élévation.
- Si la session actuelle dispose de droits limités, une fenêtre s'affiche, proposant d'utiliser un compte administrateur et d'entrer son mot de passe.

Différents scénarios peuvent donc se présenter pour la configuration du démarrage automatique d'une application Optima PLC.

##### Démarrage depuis une session ouverte disposant des droits d'administration

Cette situation est déconseillée : il est recommandé de travailler depuis une session utilisateur aux droits limités pour une meilleure sécurité.

Si l'UAC est désactivé, l'exécution ne pose pas de problème.

Si l'UAC n'est pas désactivée, le lancement de l'application nécessite une confirmation d'élévation et ne peut donc se faire de manière automatique. Il y a deux solutions :

- lancement de l'application dans cette session depuis une tâche planifiée du système (à configurer manuellement).
- lancement de l'application avec le compte Administrateur (donc sans élévation).

### Démarrage depuis une session ouverte disposant de droits limités

Ceci est la situation de travail recommandée.

L'application Optima PLC doit être lancée avec les droits d'un autre compte utilisateur disposant des droits d'administration et d'un mot de passe.

Cependant, cette exécution nécessitant normalement une élévation, deux solutions sont possibles :

- utilisation du compte Administrateur (exception à la nécessité d'élévation).
- utilisation d'un autre compte disposant des droits d'administration : dans ce cas il est nécessaire de désactiver l'UAC afin de permettre l'exécution sans demande de confirmation d'élévation.

## 10.2.3 Résolution de problèmes

### 10.2.3.1 Je définis des mnémoniques sur des variables standards mais ils n'apparaissent pas

Pour être correctement prises en compte dans les écrans de visualisation, les définitions de variables standards avec mnémoniques (registres, entrées/sorties, etc...) doivent être effectuées dans un bloc de variables globales.

De plus, avec une version antérieure à la v2.9.1, au moins l'une des déclarations faites dans ce bloc doit être utilisée dans le reste du projet, sinon le bloc tout entier n'est pas intégré à l'application compilée. A partir de la v2.9.1, les déclarations de variables globales sont systématiquement intégrées au projet.

### 10.2.3.2 J'ai une erreur de compilation sur le nom d'une fonction décrite dans la documentation

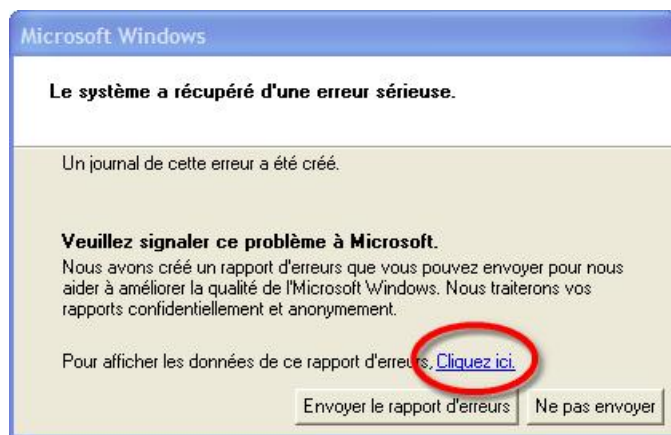
Vous avez trouvé dans la documentation une fonction (ou un bloc de fonction) que vous avez utilisé dans votre programme. Mais lors de la compilation, un message vous indique que la fonction est inconnue.

La fonction est probablement dans une librairie qui n'est pas actuellement chargée. Seuls les POU's définis dans une librairie actuellement chargée sont réellement disponibles. Cliquez sur l'onglet "Librairies" de l'arbre des POU's, et vérifiez la présence de cette librairie. Si besoin, cliquez sur "Autres librairies disponibles" et chargez la librairie contenant la fonction souhaitée.

### 10.2.3.3 Un écran bleu s'affiche ou le PC redémarre pendant l'exécution du runtime.

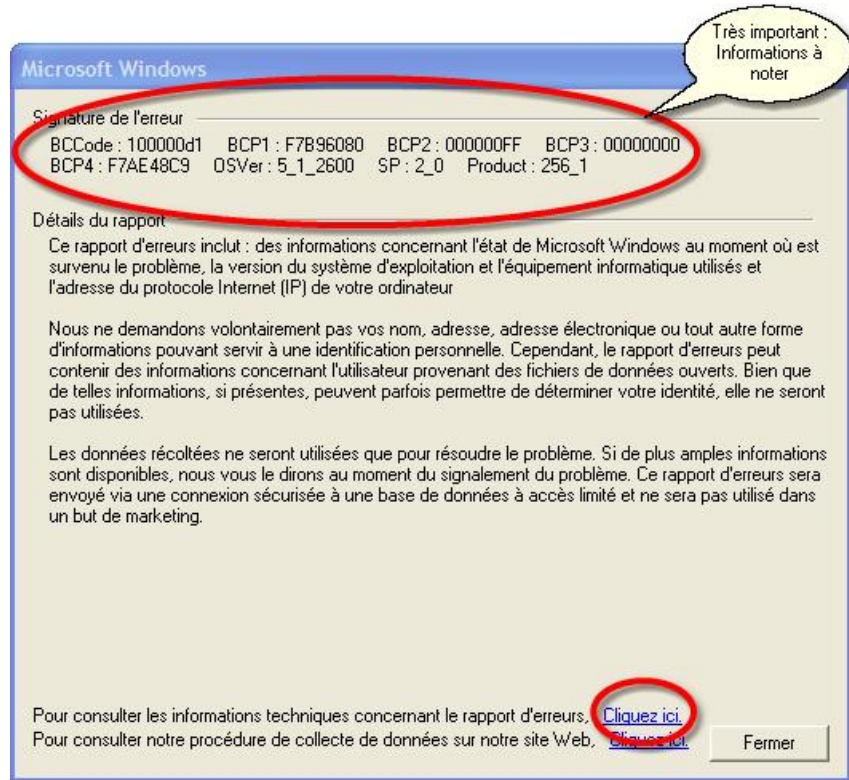
Si un écran bleu est affiché, notez les informations principales en haut de l'écran.

Dans tous les cas, lors du démarrage suivant du PC, un message d'information est affiché, demandant confirmation avant d'envoyer le rapport d'erreur à Microsoft.

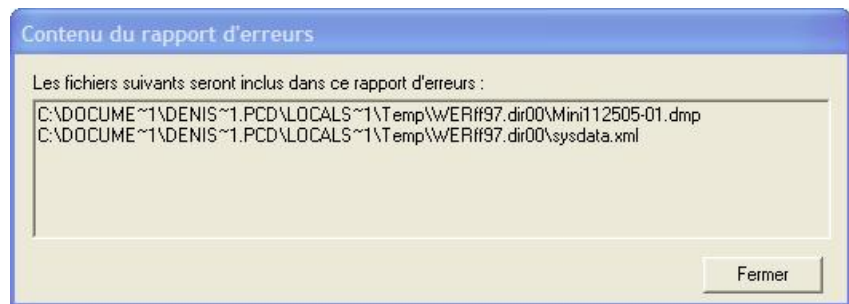


N'envoyez pas le rapport d'erreur, par contre sauvegardez les fichiers mentionnés dans ce rapport d'erreur, et faites les parvenir, ainsi que votre projet, à OptimaLog pour analyse et correction du problème.

**Attention : la sauvegarde des fichiers mentionnés ci-après doit être faite AVANT de cliquer sur "Ne pas envoyer".**



Fichiers à récupérer et à faire parvenir à OptimaLog (avant de refermer la première fenêtre) :



#### 10.2.3.4 J'ai un problème avec Optima PLC, qui dois-je contacter ?

Pour du support supplémentaire, contactez OptimaLog par email à l'adresse suivante :

[optimaplc@optimalog.com](mailto:optimaplc@optimalog.com)

en précisant vos coordonnées, la version précise d'Optima PLC, la nature de votre problème, et en joignant le maximum de fichiers permettant d'effectuer un diagnostic, en particulier les fichiers de votre application.

Optimalog s'engage à une totale confidentialité, aucun fichier transmis ne sera utilisé en dehors du diagnostic.

### 10.2.4 Divers

#### 10.2.4.1 Comment puis-je réduire le temps de compilation de mon application ?

Le temps de compilation peut prendre, dans les cas extrêmes, quelques dizaines de secondes. Il ne dépasse généralement pas la dizaine de secondes.

Outre la vitesse du PC utilisé, le facteur qui influe le plus sur le temps de compilation est le nombre de bibliothèques chargées et donc de POU's à compiler. Lors de la création d'un nouveau projet, Optima PLC place par défaut dans l'onglet Bibliothèques toutes les bibliothèques standards. Cependant, si vous n'utilisez pas certaines d'entre elles, les fermer permet de diminuer notablement le temps de compilation.

## 11 Annexes

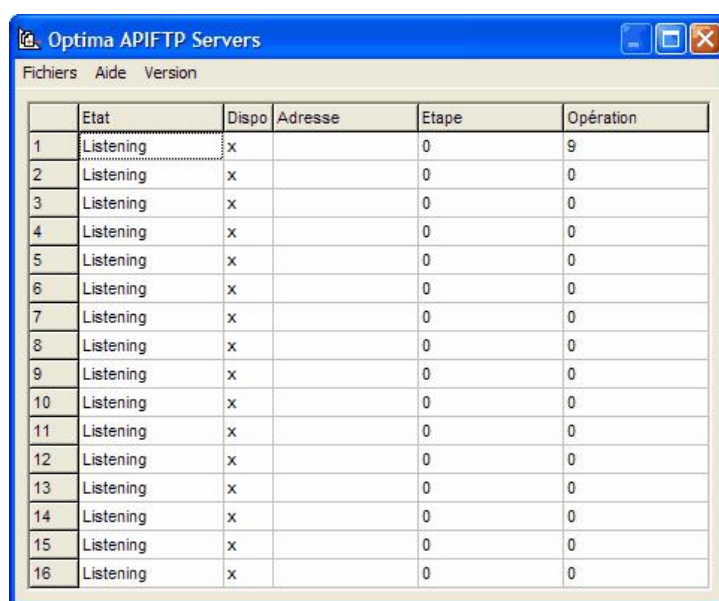
### 11.1 Optima APIFTP Servers

Ce logiciel permet de mettre en place un protocole de transfert de fichiers spécifique à Optima PLC :

- Transfert de fichiers entre PC distants, sur réseau local ou via Internet.
- Communication avec des stations fonctionnant sous Apigraf V10.
- Macros de transfert de fichiers intégrées dans la supervision Optima View.

#### 11.1.1 Page principale

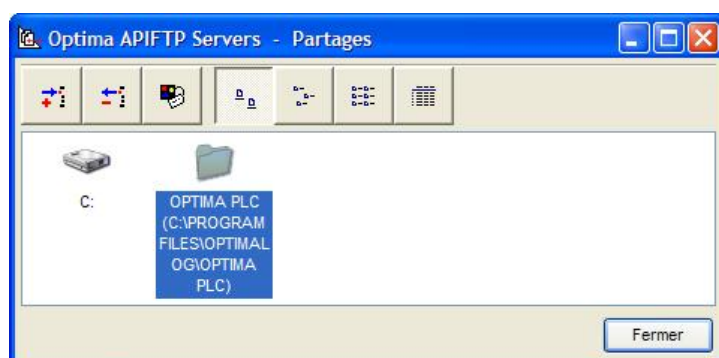
La page principale comporte un tableau de visualisation des sockets serveurs du protocole, permettant d'en visualiser l'activité.



	Etat	Dispo	Adresse	Etape	Opération
1	Listening	x		0	9
2	Listening	x		0	0
3	Listening	x		0	0
4	Listening	x		0	0
5	Listening	x		0	0
6	Listening	x		0	0
7	Listening	x		0	0
8	Listening	x		0	0
9	Listening	x		0	0
10	Listening	x		0	0
11	Listening	x		0	0
12	Listening	x		0	0
13	Listening	x		0	0
14	Listening	x		0	0
15	Listening	x		0	0
16	Listening	x		0	0

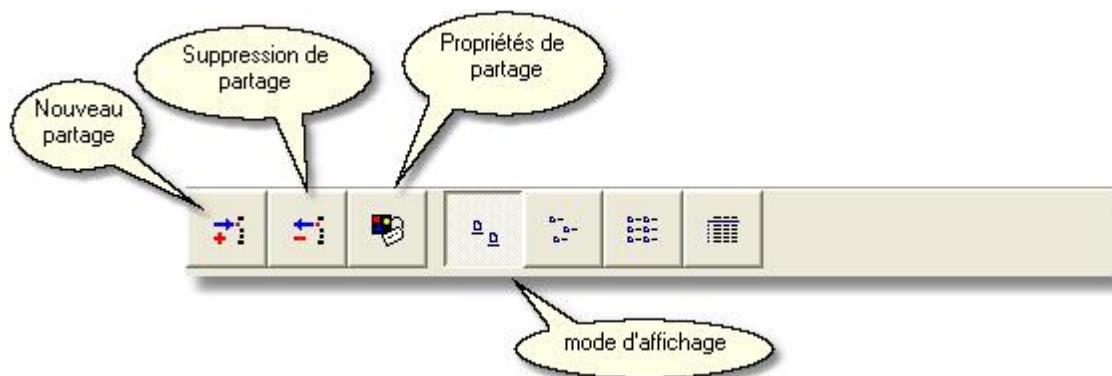
#### 11.1.2 définition des partages

L'option Partages du menu provoque la fenêtre de définition des partages.

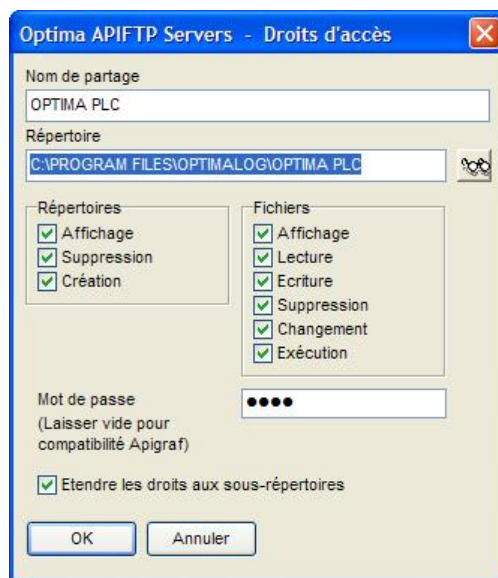


Cette fenêtre permet d'ajouter, modifier ou supprimer les partages définis.

Un "partage" est la définition d'un droit d'accès à un répertoire du PC.



Les propriétés des partages sont définies au moyen de la fenêtre suivante :



Plusieurs partages peuvent être définis pour le même répertoire avec des droits et des mots de passe différents, en leur donnant un nom de partage différent.

## 11.2 Codes de résultat APIFTP

### 11.2.1 opInProgress

Opération en cours  
Valeur : -1

La dernière opération demandée est actuellement en cours d'exécution.

### 11.2.2 opOK

Opération terminée avec succès  
Valeur : 0

### 11.2.3 opProtocolError

Erreur de protocole  
Valeur : 1

### 11.2.4 opConnectionError

Connexion impossible  
Valeur : 10

Le système distant ne répond pas. Il peut s'agir de plusieurs problèmes :

- Routage : la demande n'arrive pas à destination (vérifier avec une commande "ping" si la destination peut être atteinte).
- Blocage par un firewall : le système distant n'autorise pas l'ouverture du service nécessaire (port TCP numéro

10260).

- Le service distant n'est pas lancé : sous Windows, vérifier que Optima APIFTP Servers est démarré ; sous Apigraf, vérifier que InitAPI est démarré, et que les services réseau standards sont installés.
- Le service distant est saturé, tous les sockets sont déjà utilisés.

### 11.2.5 opConnectionAborted

Connexion interrompue

Valeur : 11

### 11.2.6 opConnectionTimeOut

Connexion terminée sur time out

Valeur : 12

### 11.2.7 opNoRessource

Pas assez de ressources matérielles

Valeur : 13

### 11.2.8 opDestinationUnreachable

Aucune ressource ne permet d'atteindre la destination

Valeur : 14

Le service distant demandé est sur une adresse IP que la station ne sait pas atteindre.

### 11.2.9 opVersionAnswerTimeOut

Pas de numéro de version de protocole reçu

Valeur : 15

### 11.2.10 opSourceNotFound

Fichier source inexistant

Valeur : 20

### 11.2.11 opUnableToOpenSource

Ouverture du fichier source impossible

Valeur : 21

### 11.2.12 opUnableToReadSource

Lecture du fichier source impossible

Valeur : 22

### 11.2.13 opExistFile

Fichier existant

Valeur : 23

Réponse positive lors d'un test d'existence de fichier.

### 11.2.14 opPathNotFound

Chemin inexistant

Valeur : 24

### 11.2.15 opAccessDenied

Accès refusé

Valeur : 25

Cette erreur indique un accès refusé à un fichier ou un répertoire, car les informations de partage ne permettent pas cet accès, ou parce que le code d'accès nécessaire n'est pas fourni ou est incorrect.

Avec APIFTP Servers, vérifier que le répertoire concerné est correctement défini dans les partages.

**11.2.16 opUnableToCreateDest**

Création du fichier destination impossible  
Valeur : 30

**11.2.17 opUnableToWriteDest**

Ecriture du fichier destination impossible  
Valeur : 31

**11.2.18 opFileSizeError**

Erreur de taille de fichier  
Valeur : 40

**11.2.19 opUnableToRenameDest**

Impossible de renommer la destination  
Valeur : 50

**11.2.20 opSourcelsDir**

La source est de type répertoire  
Valeur : 60

**11.2.21 opSourcelsVolumeID**

La source est de type identificateur de volume  
Valeur : 61

**11.2.22 opSourcelsWriteProtected**

la source est protégée en écriture  
Valeur : 62

**11.2.23 opUnableToDeleteSource**

Impossible de supprimer la source  
Valeur : 63

**11.2.24 opUnableToRenameSource**

Impossible de renommer la source  
Valeur : 70

**11.2.25 opUnableToCreateDir**

Impossible de créer le répertoire  
Valeur : 80

**11.2.26 opUnableToDeleteDir**

Impossible de supprimer le répertoire  
Valeur : 81

**11.2.27 opDirAlreadyExists**

Le répertoire existe déjà  
Valeur : 82

Une tentative a été faite pour créer un répertoire déjà existant.

**11.2.28 opUnableToGetVersion**

Impossible d'obtenir la version du fichier  
Valeur : 90

**11.2.29 opUnknownParameter**

Paramètre inconnu  
Valeur : 91

**11.2.30 opUnableToExecFile**

Impossible d'exécuter le fichier  
Valeur : 92

Le système distant n'est pas en mesure d'exécuter le fichier demandé.



### 11.2.31 opProtocolVersionError

Fonction non disponible avec la version du protocole  
Valeur : 199

Le service distant est d'une ancienne version qui n'autorise pas la fonction demandée.

### 11.2.32 opError

Erreur générale APIFTP  
Valeur : 200

Une erreur non répertoriée s'est produite au cours de la dernière opération.

# Index

## - A -

A 361  
 A propos d'Optima PLC Workshop 34  
 Affichage d'informations 389  
 Afficheurs 7 segments: 289  
 Aide 33  
 ANI 361  
 Animation: 199  
 ANO 361  
 Apigraf 452  
 Appel d'une page de supervision avec passage de paramètres: 297  
 Appel d'une page de supervision depuis une autre page: 297  
 Application ServApiOPC: 322  
 ARRAY 361  
 Aspect :: 101  
 AT 365  
 Attributs de variables 365  
 Autorisation de modification de variables automates via un lien DDE: 317

## - B -

Barre de défilement: 231  
 Barre de progression: 215  
 Barre d'outil de gestion de l'affichage des fenêtres 35  
 Barre d'outil des diagrammes ladder 54  
 Barre d'outil d'un diagramme Blocs de fonction 61  
 Barre d'outil d'un diagramme SFC 65  
 BIP (Macro PLC): 309  
 Bitmap animé: 207  
 Bloc de fonction 38  
 Bloc de programme 38  
 BOOL 359  
 Bouton API: 125  
 BOUTON MARCHE / ARRÊT: 233  
 Bouton simple: 122  
 BOUTON TYPE ARRÊT D'URGENCE: 235  
 BYTE 359

## - C -

C 361  
 Calculatrice 23  
 Calendrier/Horloge 23  
 Camembert: 197  
 CASE 370  
 Case à cocher: 115  
 Chaîne de caractères 359  
 Changement de langue: 315

Changement de taille d'un objet ou d'un groupe d'objets à l'aide de la fonction zoom: 111

CHAR 359  
 ChnMdp (Macro locale): 302  
 ChnUserLanguage (Macro locale): 302  
 Classeur: 164  
 Close (Macro locale): 302  
 COLONNE 3 COULEURS: 244  
 Compilation générale 26  
 Compilation totale 20  
 Confirmation 389  
 CONSTANT 365  
 Constantes  
   chaînes de caractères 366  
   dates 367  
   durée 367  
   numériques 366  
   typées 368

Constantes système 27  
 Consultation des DLL utilisateur: 314  
 Contenu des POU 41, 43, 45, 46, 47, 48, 50  
 Contenu d'un projet 2  
 Contenu d'une librairie 36  
 Copie d'objets à l'aide de la souris: 112  
 Cordon de bits 359  
 Création de diagrammes Bloc de fonction 60  
 Création de diagrammes Ladder 54  
 Création de diagrammes SFC 64  
 Création d'un nouveau fichier de langage: 315  
 Création d'un objet de supervision: 109  
 Création d'un programme DELPHI5 utilisant le paquet USERAPIPLC: 318  
 Création d'une action 66  
 Création d'une page de supervision: 100  
 Créer un POU 52  
 Curseur: 225  
 Cuve: 251

## - D -

Date et temps 358  
 DAY 363  
 DAYOFWEEK 363  
 Déclarations de variables 364  
 DECR### (Macro PLC): 309  
 Défilement: 166  
 Définition des données en lecture/écriture: 324  
 Définition du nombre de variables locales: 321  
 Définition d'un groupe: 324  
 Définition d'un item: 324  
 Demande de confirmation 389  
 Démarrage de l'application: 326  
 Déplacement et changement de taille d'un objet de supervision: 110  
 Description des variables locales: 321  
 Développement d'une DLL Utilisateur: 309  
 DINT 357

Document précédent 22

Document suivant 22

DT 361

Durée 358

DWORD 359

## - E -

Echelle: 195

Editeur Optima PLC 14

ELSE 370

ELSIF 370

END\_CASE 370

END\_FOR 370

END\_IF 370

END\_REPEAT 370

END\_STRUCT 360

END\_TYPE 360

END\_WHILE 370

Enregistrement d'objets Personnalisés: 298

Entiers 357

ExecProg (Macro locale): 304

Exécution 22

Exécution d'une page avec son icône: 316

Exécution d'une page depuis l'atelier: 316

Exemple de création d'un diagramme ladder 60

Exemple de création d'un diagramme SFC 79

Exemple de diagrammes blocs de fonction 63

Exemple de DLL Utilisateur: 312

Exemple de traduction: 315

Exemple : Création d'un lien DDE avec MICROSOFT®

Excel: 317

EXIT 370

## - F -

Fenêtre d'édition avec menu déroulant: 160

Fenêtre d'édition: 148

Fenêtre système 15

Fermeture d'une page depuis un bouton: 297

Fichiers créés dans un projet de supervision: 316

Fichiers de sortie texte 28, 390

Fonction 39

Fonction AllLinksOK: 319

Fonction IsLinkedToAPI: 319

Fonction LinkToAPI: 318

Fonction PLCState: 320

Fonction UnLinkToAPI: 319

Fond: 103

FOR 370

Format Date/Heure 23

Forme: 176

## - G -

Généralités: 309, 322

Gestion des contacts 55

Gestion des mots de passe: 298

Gestion du temps 387

Gestionnaire de fichiers 15

Grafcet 64

Graphe: 181

Grille: 104

## - H -

Histogramme: 192

Horodatage 358

HOURS 363

## - I -

I 361

IF 370

Image: 209

Importation d'objets personnalisés: 299

Impression 85

INCR### (Macro PLC): 309

Initialisation de types et variables multi-éléments 368

Initialisation de variables 368

Initialisation des types et variables 368

Insertion images succesives: 202

Installation de l'application: 325

Installation d'Optima PLC 4

Instances de POU 365

INT 357

Interfaces matérielles 27

INVSA### (Macro PLC): 309

## - J -

Jauge: 211

## - L -

L 361

Langue logiciel 11

Langue programmation 11

LED: 246

LEVIER SANS COULEUR: 235

Librairies "Projet" 41

Ligne: 178

LINT 357

Liste des fenêtres 24

Liste des POU 14

LREAL 357

LWORD 359

## - M -

Macros: 108  
 Menu Affichage - Barres d'outils: 89  
 Menu Affichage - Fenêtre Fonction: 89  
 Menu Automate 337  
 Menu Contrôle 334  
 Menu Création de raccourcis: 86  
 Menu de gestion des POU: 53  
 Menu de gestion des bibliothèques 51  
 Menu Drivers 343  
 Menu édition - Annuler: 86  
 Menu édition - Coller: 86  
 Menu édition - Copier: 86  
 Menu édition - Supprimer: 86  
 Menu édition - Tout sélectionner: 86  
 Menu Explorateur Apigraf IP: 88  
 Menu Expressions 340  
 Menu fichier - Enregistrer sous: 86  
 Menu fichier - Enregistrer: 86  
 Menu fichier - Nouveau: 86  
 Menu fichier - Ouvrir: 86  
 Menu fichier - Quitter: 86  
 Menu fichier - Sélection projet: 86  
 Menu Macros - Macros locales: 92  
 Menu Macros - Macros PLC: 92  
 Menu Macros - Macros Utilisateur: 92  
 Menu Objet - Ajouter: 90  
 Menu Objet - Importer: 91  
 Menu Objet - Supprimer: 90  
 Menu Options - Edition configuration: 97  
 Menu Options - Options générales: 95  
 Menu Sorties Texte 341  
 Menu Traductions utilisateur - Extraction Auto textes de la page: 87  
 Menu Traductions utilisateur - Nouveau fichier langage: 87  
 Menu Traductions utilisateur - Ouvrir un fichier langage: 87  
 Menu Variables 339  
 Menus contextuels des diagrammes ladder 59  
 Menus contextuels des diagrammes SFC 66  
 Menus contextuels d'un diagramme blocs de fonction 61  
 Message d'erreur 389  
 MINUTES 363  
 Mise en avant plan ou en arrière plan d'un objet: 112  
 Mode vidéo 22  
 MONTH 363

## - N -

Négation 57  
 Nom du serveur OPC: 324

## - O -

O 361  
 OF 361  
 OLE Container: 291  
 Options Editeur 24, 25  
 Options: 105

## - P -

Panneau: 169  
 Plages de valeurs 361  
 Police: 102  
 Position: 104  
 Première utilisation création d'un projet 8  
 Principes généraux: 99, 317, 318, 321  
 PrintPage (Macro locale): 305  
 Procédure ExecMacro: 319  
 Programmation 356  
     Avancée 387  
     Diagrammes Bloc de fonction 60  
     Diagrammes Ladder 54  
     Diagrammes SFC (Grafcet) 64  
     Langage Apigraf 375  
     Langage IL 371  
     Langage ST 369

## - Q -

Qualificateur d'action 67  
     D 69  
     DS 72  
     L 69  
     N 68  
     P 70  
     P1 et P0 70  
     S et R 68  
     SD 71  
     SL 72  
 Quitter 16

## - R -

R 361  
 Radio group: 117  
 REAL 357  
 Recette: 286  
 Réels 357  
 REPEAT 370  
 RETAIN 365  
 RETURN 370  
 RR 361  
 RunHelp (Macro locale): 305  
 RunHelpIndex (Macro locale): 305

RunHelpTopic (Macro locale): 305

## - S -

SA 361

Saisie des macros pour les objets: 112

SECONDS 363

SELECTEUR 2 / 3 POSITIONS: 236

SELECTEUR ARRÊT / AUTO / MANU: 238

Sélection d'un objet ou d'un groupe d'objets de supervision: 110

Sélection d'un projet: 326

Sélection projet 13

SET...= (Macro PLC): 309

SINT 357

Sous-groupe 39

Statut de l'application 329

STRUCT 360

Structure de données 360

SuperG (Macro locale): 306

Suppression d'objets de supervision: 111

Suppression d'objets personnalisés: 299

Suppression d'une DLL utilisateur d'un projet: 314

## - T -

Tableaux de données 361

Tableaux de mesures 389

Tâche 36

Techniques de gestion du temps 387

Temps 387

Texte défilant: 205

Texte: 113

THEN 370

TIME 358

TYPE 360

Types 38

Types dérivés 360

Types énumérés 361

Types génériques 360

## - U -

UDINT 357

UINT 357

ULINT 357

UNTIL 370

USINT 357

Utilisation des fichiers de langage dans les pages de supervision: 315

Utilisation d'une DLL Utilisateur dans un projet: 314

## - V -

Vanne: 254

Vanne1: 256

VAR 364

VAR\_EXTERNAL 364

VAR\_GLOBAL 364

VAR\_IN\_OUT 364

VAR\_INPUT 364

VAR\_OUTPUT 364

VAR\_TEMP 364

Variables globales 38

Visionneur HTML: 167

Visualisation des variables locales: 322

VisuText (Macro locale): 306

VOYANT DOUBLE: 242

VOYANT SIMPLE: 240

Vu mètre à aiguille: 223

Vu mètre: 213

## - W -

WEEKNB 363

WFCopy (Macro locale): 306

WFCreateDirectory (Macro locale): 306

WFDelete (Macro locale): 306

WFDeleteDirectory (Macro locale): 306

WFRename (Macro locale): 306

WFSetFileAttr (Macro locale): 307

WHILE 370

WNetSChangeFileName (Macro locale): 307

WNetSCopyFileRemote (Macro locale): 307

WNetSCreateDirectory (Macro locale): 307

WNetSDeleteDirectory (Macro locale): 307

WNetSReadAndMoveFile (Macro locale): 307

WNetSReadFile (Macro locale): 308

WNetSSetFileAttr (Macro locale): 308

WNetSWriteAndMoveFile (Macro locale): 308

WNetSWriteFile (Macro locale): 308

WORD 359

## - Y -

YEAR 363