

# **CRIMSON 3**

# **MANUEL DE**

# **L'UTILISATEUR**



---

Copyright © 2003-2009 Red Lion Controls Inc.

Tous droits internationaux réservés

Les informations contenues dans le présent manuel sont données de bonne foi mais restent sujettes à modification sans notification préalable. Ce manuel ne constitue pas une garantie et ne représente pas un engagement de la part de Red Lion Controls. Les sociétés, noms et données utilisés en tant qu'exemples sont fictionnels à moins qu'il en soit indiqué autrement. Aucune partie de ce document ne peut être reproduite ou transmise, sous quelque forme et par quelque moyen, électroniques ou mécaniques, sans l'autorisation écrite expresse de Red Lion Controls Inc.

Le logo Red Lion est une marque déposée de Red Lion Controls Inc.

Crimson et le logo Crimson sont des marques déposées de Red Lion Controls Inc.

Toutes les autres marques déposées sont la propriétés exclusives de leurs propriétaires respectifs.

Ecrit par Mike Granby et Jesse Benefiel.

---

# TABLE DES MATIERES

CONFIGURATION SYSTEME .....	1
INSTALLATION DU LOGICIEL .....	1
ENREGISTREMENT .....	2
RECHERCHE DE MISES A JOUR .....	2
INSTALLATION DES PILOTES USB.....	2
RESOLUTION DES PROBLEMES .....	4
ETAPES SUIVANTES .....	5
INFO-BULLES .....	7
APPARENCE DE LA FENETRE .....	7
LE PANNEAU DE NAVIGATION.....	8
LE PANNEAU DE RESSOURCES .....	8
LE PANNEAU D'EDITION .....	8
REDUIRE LES PANNEAUX .....	8
CATEGORIES .....	8
COMMUNICATIONS.....	8
TAGS DE DONNEES.....	9
PAGES D’AFFICHAGE .....	9
PROGRAMMES.....	9
SERVEUR WEB .....	9
JOURNAL DE DONNEES .....	9
SECURITE .....	10
NAVIGUER DANS LA BASE DE DONNEES .....	10
RECULER ET AVANCER.....	10
RACCOURCIS CATEGORIES .....	10
RACCOURCIS ELEMENTS .....	11
LISTES DE NAVIGATION .....	11
TRAVAILLER AVEC LES DOSSIERS .....	12
CLASSEMENT DES LISTES ET DES DOSSIERS .....	12
OPERATIONS DE GLISSER / DEPOSER .....	12
RECHERCHE DANS LES LISTES .....	12
ANNULER ET RETABLIR.....	12
RECHERCHE GLOBALE.....	13
TRAVAILLER AVEC LES BASES DE DONNEES .....	13
IDENTIFIANTS BASE DE DONNEES.....	13
ENREGISTRER UNE IMAGE .....	14
PROTECTION DES BASES DE DONNEES.....	14
CONVERTIR UNE BASE DE DONNEES.....	14
RECHERCHE D’ERREURS.....	15
TELECHARGEMENT VERS UN PERIPHERIQUE .....	15
CONFIGURATION DU LIEN .....	15
ENVOI DE LA BASE DE DONNEES .....	16
EXTRACTION DE BASES DE DONNEES .....	16
MISE EN PLACE DE LA CARTE COMPACTFLASH .....	17

FORMATAGE DE LA CARTE COMPACTFLASH.....	18
ENVOI DE L'HEURE ET DE LA DATE .....	18
CHOIX DU PORT SERIE .....	20
SELECTIONNER UN PROTOCOLE .....	20
OPTIONS DU PROTOCOLE.....	21
TRAVAILLER AVEC DES PERIPHERIQUES .....	21
PARAMETRES AVANCES.....	22
UTILISATION DU PORT ET DU PERIPHERIQUE .....	23
CONFIGURATION RESEAU .....	23
PARAMETRES ETHERNET.....	23
PORTS MULTIPLES .....	24
PARAMETRES DE ROUTING .....	24
PARAMETRES DE TELECHARGEMENT.....	25
AJOUT DE PORTS.....	26
SELECTION DE PROTOCOLE .....	26
UTILISATION DES PORTS VIRTUELS .....	27
UTILISATION DES CARTES D'EXPANSION .....	27
PROTOCOLES ESCLAVES.....	28
CHOIX DU PROTOCOLE.....	28
AJOUT DE BLOCS DE PASSERELLE .....	28
AJOUT D'ELEMENTS A UN BLOC.....	29
ACCES AUX BITS INDIVIDUELS.....	30
CONVERSION DE PROTOCOLE .....	30
MAITRE ET ESCLAVE .....	30
MAITRE ET MAITRE .....	31
OU CREER LE BLOC DE PASSERELLE ? .....	31
CONTROLE DES BLOCS MAITRES .....	32
TRANSFORMATION DE DONNEES .....	32
DESACTIVATION DES COMMUNICATIONS.....	33
TOUT SUR LES TAGS.....	35
SOURCES DE DONNEES.....	35
TYPES DE TAGS.....	35
ATTRIBUTS DES TAGS .....	36
AVANTAGES DES TAGS .....	36
EDITION DES PROPRIETES.....	37
PROPRIETES D'EXPRESSION .....	37
GESTION DES LANGUES .....	40
PROPRIETES DOUBLE SENS.....	41
PROPRIETES D'ACTION.....	41
PROPRIETES COULEUR .....	41
PROPRIETES DE JOURNALISATION.....	42
CREATION DE TAGS.....	42
DUPLICATION DE TAGS.....	43
EDITION DE TAGS MULTIPLES.....	43
UTILISATION DE LA COMMANDE COPIER DEPUIS .....	43
UTILISATION DE LA COMMANDE COLLAGE SPECIAL.....	43

SELECTIONS DE PROPRIETE .....	44
IMPORTER ET EXPORTER.....	45
TROUVER L'UTILISATION D'UN TAG .....	45
TAGS NUMERIQUES .....	45
PROPRIETES DE DONNEES.....	45
PROPRIETES DE FORMAT .....	48
PROPRIETES COULEUR .....	49
PROPRIETES D'ALARME .....	49
PROPRIETES D'ACTIVATION .....	51
PROPRIETES DE PLOT .....	52
PROPRIETES DE SECURITE .....	53
TAGS DRAPEAUX .....	53
PROPRIETES DE DONNEES.....	53
PROPRIETES DE FORMAT .....	55
PROPRIETES COULEUR .....	56
PROPRIETES D'ALARME .....	56
PROPRIETES D'ACTIVATION .....	58
PROPRIETES DE SECURITE .....	58
LIGNES DE TAGS .....	58
PROPRIETES DE DONNEES.....	59
PROPRIETES DE FORMAT .....	61
PROPRIETES COULEUR .....	61
PROPRIETES DE SECURITE .....	62
TAGS DE BASE .....	62
TAGS FLUX DE DONNEES.....	62
LECTURE TAGS NUMERIQUES.....	63
ECRITURE TAGS NUMERIQUES .....	63
UTILISATION SUR ECRITURE .....	63
PROPRIETES TAGS FLECHES .....	64
TYPES DE FORMATS.....	65
FORMAT GENERAL.....	66
FORMAT NUMERIQUE.....	66
FORMAT SCIENTIFIQUE .....	67
FORMAT HEURE ET DATE .....	68
FORMAT ADRESSE IP .....	69
FORMAT DEUX-ETATS .....	69
FORMAT MULTI-ETATS .....	69
TYPES DE COULEURS .....	71
OPERATIONS GENERALES .....	71
COULEURS FIXES .....	71
COLORATION DEUX-ETATS .....	72
COLORATION MULTI-ETATS .....	72
PRINCIPES DE BASE DE L'EDITEUR .....	75
TRAVAILLER AVEC LES PAGES .....	75
MODIFIER LE NIVEAU DU ZOOM.....	75

LE PANNEAU DE RESSOURCES.....	76
AJOUTER DES OBJETS A UNE PAGE .....	77
<b>TRAVAILLER AVEC LES PRIMITIVES .....</b>	<b>77</b>
SELECTION DES PRIMITIVES .....	77
UTILISATION DE LA BARRE RAPIDE .....	78
DEPLACER LES PRIMITIVES ENTRE LES PAGES.....	78
DEPLACER LES PRIMITIVES ENTRE LES BASES DE DONNEES .....	78
MODIFIER LA TAILLE DES PRIMITIVES .....	78
UTILISATION DES POIGNEES DE MISE EN PAGE .....	78
ALIGNEMENT INTELLIGENT .....	79
ALIGNEMENT RAPIDE .....	79
UTILISATION DE LA GRILLE .....	80
ALIGNEMENT DES PRIMITIVES .....	80
ESPACEMENT DES PRIMITIVES.....	81
REORGANISATION DES PRIMITIVES.....	81
DUPLICATION DES PRIMITIVES.....	81
EDITION DE PRIMITIVES MULTIPLES .....	82
PASSAGE VERS D'AUTRES ELEMENTS .....	83
<b>PROPRIETES DES PRIMITIVES .....</b>	<b>84</b>
AFFICHER OU CACHER LES PRIMITIVES .....	84
DEFINIR LES COULEURS DES PRIMITIVES.....	84
DEFINIR LES COULEURS DE CLIGNOTEMENT .....	86
DEFINIR LES COULEURS 2-ETATS.....	86
DEFINIR LES COULEURS 4-ETATS.....	87
DEFINIR LE MODE MELANGE .....	87
DEFINIR LES EXPRESSIONS DE COULEURS.....	87
DEFINIR DES JAUGES DE NIVEAU .....	88
DEFINIR DES FORMATS DE JAUGE.....	89
DEFINIR LES STYLES DES BORDURES.....	90
<b>UTILISATION DES GROUPES UTILISATION DE GROUPES .....</b>	<b>90</b>
FAIRE ET DEFAIRE DES GROUPES .....	90
EDITION A L'INTERIEUR DES GROUPES .....	91
EDITION DE GROUPES IMBRIQUES.....	91
<b>AJOUTER DU TEXTE A DES PRIMITIVES .....</b>	<b>92</b>
<b>AJOUTER DES DONNEES AUX PRIMITIVES .....</b>	<b>94</b>
<b>AJOUTER DES ACTIONS A DES PRIMITIVES.....</b>	<b>98</b>
PROTEGER DES ACTIONS .....	98
ACTIVER LES ACTIONS .....	99
L'ACTION ALLER A LA PAGE.....	99
L'ACTION DEFINIE PAR L'UTILISATEUR.....	100
L'ACTION APPUYER SUR LE BOUTON.....	101
L'ACTION MODIFIER LA VALEUR .....	102
L'ACTION MODIFIER VALEUR RAMPE.....	102
L'ACTION JOUER UNE MELODIE .....	103
L'ACTION CONNECTER UTILISATEUR .....	103
L'ACTION DECONNECTER UTILISATEUR .....	103
<b>AJOUTER DES ACTIONS A DES TOUCHES .....</b>	<b>104</b>
<b>EDITER LES PROPRIETES PAGES .....</b>	<b>105</b>
<b>PARAMETRES DE L'INTERFACE UTILISATEUR.....</b>	<b>107</b>

PROPRIETES GLOBALES .....	107
PROPRIETES ENTREE .....	108
PROPRIETES IMAGES .....	109
PROPRIETES DE POLICE.....	110
<b>GESTION DES IMAGES.....</b>	<b>110</b>
<b>GESTION DES POLICES.....</b>	<b>112</b>
<b>PRIMITIVES NOYAU.....</b>	<b>115</b>
PRIMITIVES GEOMETRIQUES .....	115
PRIMITIVES 3D.....	116
PRIMITIVES BOUTON .....	116
PRIMITIVES TEXTE ET DONNEES .....	117
PRIMITIVE LIGNE.....	118
PRIMITIVE IMAGE .....	118
MISE A L'ECHELLE PRIMITIVE .....	120
<b>FLECHES.....</b>	<b>122</b>
<b>POLYGONES ET ETOILES .....</b>	<b>123</b>
POLYGONES .....	123
ETOILES.....	123
<b>INFO-BULLES .....</b>	<b>124</b>
<b>FIGURES SEMI-HACHUREES.....</b>	<b>125</b>
<b>BOUTONS D'ACTION .....</b>	<b>125</b>
<b>BOUTONS ILLUMINES .....</b>	<b>126</b>
<b>INDICATEURS.....</b>	<b>127</b>
<b>BASCULES 2-ETATS .....</b>	<b>128</b>
<b>BASCULES 3-ETATS .....</b>	<b>130</b>
<b>SELECTEURS 2-ETATS .....</b>	<b>131</b>
<b>SELECTEURS 3-ETATS .....</b>	<b>131</b>
<b>PRIMITIVES HERITEES.....</b>	<b>132</b>
FRAGMENTS A ELLIPSE .....	132
SLIDERS RICHES.....	132
<b>PRIMITIVES SYSTEME .....</b>	<b>134</b>
VISUALISATION DU FORMAT .....	134
ALARMES DE VISUALISATION .....	135
VISUALISATION EVENEMENTS .....	135
FICHIER DE VISUALISATION.....	137
GESTIONNAIRE UTILISATEURS .....	137
VISUALISATEUR TENDANCES.....	138
STIMULATION DE LA CALIBRATION .....	140
TEST STIMULATION.....	140
<b>SELECTION DES LANGUES .....</b>	<b>141</b>
<b>CONFIGURATION DE LA TRADUCTION AUTOMATIQUE.....</b>	<b>142</b>
<b>TRADUIRE VOTRE BASE DE DONNEES.....</b>	<b>142</b>
SAISIE DES TRADUCTIONS .....	143
TRADUCTION AUTO GLOBALE .....	143
EXPORTATION ET IMPORTATION .....	143
APPLIQUER UN LEXIQUE .....	143

PREVISUALISATION DES TRADUCTIONS .....	144
BASCULER ENTRE LES LANGUES.....	144
CREATION D'UN WIDGET.....	145
RESUME .....	149
POURQUOI CETTE QUESTION EST-ELLE IMPORTANTE.....	150
PLUS DE DETAILS.....	150
DEFINITIONS DES DONNEES DES WIDGETS.....	150
REPLISSAGE WIDGETS.....	152
LIER LES DOSSIERS .....	152
LIAISONS AVANCEES.....	154
CORRESPONDANCE DE CLASSE .....	154
LIER PREFIXES .....	154
UTILISER LES LIAISONS POUR .....	155
WIDGETS DE DETAILS.....	155
ACTIVER LA CREATION DE DETAILS.....	156
DEFINITION DES ELEMENTS DE DONNEES .....	156
RESULTATS DE LA LIAISON .....	156
PAGES DETAILS MULTIPLES .....	157
CREATION DE JOURNAUX DE DONNEES.....	159
JOURNALISATION GROUPEE .....	160
CONTROLE DES LOTS.....	161
STOCKAGE DES FICHIERS JOURNAUX .....	161
PROCESSUS DE JOURNALISATION .....	161
ACCEDER AUX JOURNAUX .....	162
NOTE IMPORTANTE .....	163
PROPRIETES DU SERVEUR WEB .....	163
AJOUT DE PAGES WEB.....	165
UTILISATION D'UN SITE WEB PERSONNALISE .....	166
CREATION DU SITE .....	166
DONNEES EMBARQUEES.....	166
DEPLOIEMENT DU SITE.....	166
PRINCIPES DE BASE.....	167
SECURITE BASEE OBJET .....	167
UTILISATEURS NOMMES .....	167
DROITS UTILISATEURS.....	168
CONTROLE D'ACCES .....	168
JOURNALISATION DES ECRITURES .....	168
ACCES PAR DEFAUT .....	169
CONNEXION A LA DEMANDE.....	169
ACCES A LA MAINTENANCE .....	169
VERIFIER AVANT FONCTIONNEMENT .....	169
PARAMETRES DE SECURITE .....	170
CREATION DE COMPTES UTILISATEURS.....	171
SECURITE DES TAGS .....	171
SECURITE DES PAGES .....	172

FONCTIONS RELATIVES A LA SECURITE.....	172
GESTION DU TEMPS .....	173
CONFIGURATION DU SERVICE .....	173
SELECTION D'UN SERVEUR SMTP.....	175
CONFIGURATION DES FUSEAUX HORAIRES.....	175
UTILISATION DU SERVEUR FTP .....	176
CONFIGURATION DU SERVICE .....	176
SECURITE FTP.....	176
SYNCHRONISATION DES FICHIERS .....	177
CONFIGURATION DU SERVICE .....	177
UTILISATION DE LA MESSAGERIE ELECTRONIQUE .....	179
AJOUT DE CONTACTS.....	179
CONFIGURATION DU PROTOCOLE SMTP.....	179
CONFIGURATION DU PROTOCOLE SMS .....	181
ACTIVATION TCP / IP .....	184
PARTAGE DU PORT .....	184
CONNEXION VIA UN AUTRE PORT .....	185
CONNEXION VIA ETHERNET .....	185
PORTS VIRTUELS PURS.....	187
LIMITATIONS.....	187
AJOUT D'UNE CONNEXION D'APPEL ENTRANT .....	189
AJOUT D'UNE CONNEXION D'APPEL SORTANT .....	191
AJOUT D'UNE CONNEXION SMS.....	193
TRAITEMENT DES MESSAGES SMS .....	193
VERIFICATION DU STATUT DU MODEM.....	193
RESOLUTION DES PROBLEMES DE COMMUNICATION MODEMS .....	195
UTILISATION D'INTERFACES MULTIPLES.....	195
SUPPORT CARTE MEMOIRE .....	197
PROPRIETES GENERALES.....	197
PROPRIETES DE TRANSFERT .....	198
LISTE DES PROGRAMMES .....	199
USAGE DES PROGRAMMES .....	199
EDITION DES PROGRAMMES.....	199
AIDE .....	200
PANNEAU DE RESSOURCES .....	200
TYPES DE DONNEES PROGRAMME .....	201
PROPRIETES PROGRAMME .....	201
AJOUT DE COMMENTAIRES .....	203
RETOUR DE VALEURS.....	203
GARE !.....	203
PASSAGE D'ARGUMENTS .....	204
CONSEILS ET ASTUCES POUR LA PROGRAMMATION .....	204
ACTIONS MULTIPLES.....	204
INSTRUCTIONS IF.....	204
INSTRUCTIONS SWITCH .....	205
VARIABLES LOCALES .....	206
CREATIONS DE BOUCLES.....	206

VALEURS DE DONNEES .....	209
CONSTANTES.....	209
VALEURS DE TAGS.....	211
PROPRIETES DES TAGS.....	211
PROPRIETES DES PAGES .....	211
REFERENCES COMM.....	211
UN PEU DE MATHS .....	212
PRIORITE OPERATEUR.....	212
CONVERSION DE TYPE.....	212
COMPARAISON DE VALEURS .....	213
TEST DES BITS.....	213
CONDITIONS MULTIPLES.....	214
CHOIX DES VALEURS.....	214
MANIPULATION DES BITS .....	214
AND, OR ET XOR.....	215
BASCULE OPERATEURS .....	215
COMMANDE NOT .....	215
INDEXATION FLECHES .....	215
INDEXATION CHAINES .....	216
AJOUT DE CHAINES .....	216
APPEL PROGRAMMES .....	216
UTILISATION DES FONCTIONS .....	216
RESUME PRIORITES.....	216
CHANGER DE PAGE.....	219
CHANGER LES VALEURS NUMERIQUES .....	219
ASSIGNEMENT SIMPLE.....	219
ASSIGNEMENT ELEMENT .....	219
INCREMENTATION ET DECREMENTATION .....	219
MODIFICATION DE VALEURS DE BITS .....	220
UTILISATION DES PROGRAMMES .....	220
UTILISATION DES FONCTIONS .....	220
PRIORITE OPERATEUR.....	220

---

# **CRIMSON 3**

## **MANUEL DE**

### **L'UTILISATEUR**

---



# PREMIERS PAS

Bienvenue sur Crimson 3 – la toute dernière version du célèbre logiciel de configuration d'interface opérateur. Si vous possédez une version plus ancienne de Crimson, vous prendrez vite conscience de la grande quantité d'améliorations apportées par Crimson 3. Si vous êtes un nouveau venu, prenez soin de lire au moins les premiers chapitres de ce manuel afin de comprendre comment fonctionne le logiciel. De toutes manières vous vous rendrez rapidement compte que Crimson 3 vous permet de concevoir plus vite et plus facilement des systèmes d'interface opérateur de qualité.

## CONFIGURATION SYSTEME

Crimson 3 est conçu pour fonctionner sur n'importe quelle version de Microsoft Windows, à partir de Windows 2000. La configuration mémoire minimale est modeste et tout système qui répond aux exigences de base pourra supporter Crimson sans le moindre problème. Environ 100 MB d'espace disque seront nécessaires pour l'installation et vous devriez, dans l'idéal, bénéficier d'un affichage possédant une résolution suffisante pour afficher des pages sans avoir à vous déplacer trop souvent à l'intérieur de ces dernières. Pour votre périphérique VGA cible, nous recommandons un XGVA PC.

## INSTALLATION DU LOGICIEL

Crimson 3 est fourni sous la forme d'un installateur Windows classique, ou fichier `msi`. Vous aurez généralement téléchargé ce fichier depuis le site web de Red Lion, mais si vous avez choisi une autre source de téléchargement, veuillez vous assurer que Windows se satisfait de la signature numérique de logiciel, ce qui vous offrira l'assurance de télécharger le bon produit...



Comme montré ci-dessus, l'éditeur doit être Red Lion Controls Inc. et vous devez avoir la possibilité de cliquer sur le nom de l'éditeur pour vérifier la validité de la signature numérique. Une fois que vous avez mené ces vérifications si désiré, cliquez sur Exécuter pour lancer l'installation.

Le processus d'installation est classique et vous n'aurez pas à beaucoup intervenir après avoir indiqué le répertoire de destination. Une fois l'installation terminée, allez dans le menu Démarrer et recherchez le dossier Red Lion. Cliquez sur l'icône Crimson 3 pour lancer le logiciel.

## ENREGISTREMENT

Lors du premier lancement de Crimson 3, il vous sera demandé si vous souhaitez enregistrer le logiciel...



Si l'enregistrement de votre copie de Crimson 3 est optionnel, nous vous recommandons fortement de profiter de l'opportunité qui vous est faite de nous faire parvenir vos informations de contact afin que nous puissions vous informer des mises à jour et parutions de produits associés. L'enregistrement nécessitant une connexion internet active, vous pouvez ignorer cette étape si vous n'êtes pas en ligne. Crimson vous rappellera régulièrement que vous utilisez une copie non enregistrée du logiciel.

## RECHERCHE DE MISES A JOUR

Si vous possédez une connexion internet, vous pouvez utiliser la commande Recherche de mises à jour du menu Aide pour vérifier la présence d'une nouvelle version de Crimson 3. Si une version plus récente que celle que vous utilisez est trouvée, Crimson vous demandera la permission de télécharger la mise à niveau et de mettre automatiquement votre logiciel à jour. Vous pouvez également télécharger cette dernière manuellement à partir du site web Red Lion via la page Téléchargements de la section Support.

## INSTALLATION DES PILOTES USB

Si vous avez suivi les instructions accompagnant votre matériel, vous n'avez pas encore connecté ce dernier à votre PC. Maintenant que l'installation de Crimson 3 est achevée, vous pouvez connecter en toute sécurité votre périphérique à l'aide d'un câble USB standard. Windows devrait rapidement vous indiquer avoir trouvé un nouveau matériel à travers une fenêtre de dialogue semblable à celle-ci...



Conservez les valeurs par défaut des paramètres afin que Windows puisse localiser automatiquement le pilote et appuyez sur le bouton Suivant, Après un moment de réflexion – qui peut prendre quelques temps si de nombreux logiciels sont installés sur votre machine – une fenêtre de dialogue plutôt intimidante apparaîtra, vous avertissant que les pilotes Crimson ne possèdent pas de signature numérique...



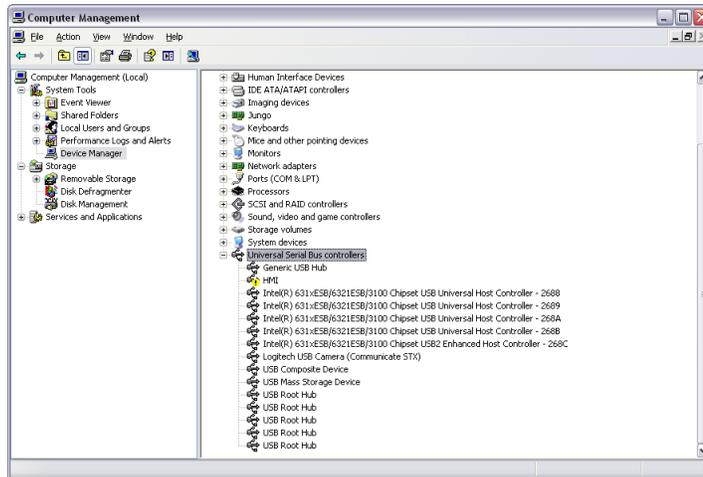
Appuyez sur le bouton Continuer et vous devriez rapidement voir apparaître cette fenêtre...



Veillez noter qu'il peut être nécessaire de répéter le processus d'installation des pilotes jusqu'à trois fois, Crimson utilisant des pilotes différents pour ses applications. Vous serez soulagé d'apprendre qu'il n'est plus nécessaire de réinstaller les pilotes à chaque fois qu'un nouveau périphérique est connecté. Cela signifie en clair que nous avons cessé de rêver à supporter plus d'un seul périphérique sur un segment USB, mais nous ne doutons pas que cela vous conviendra parfaitement.

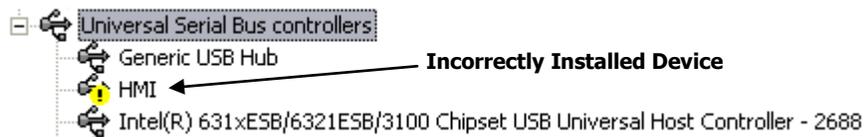
## RESOLUTION DES PROBLEMES

Si vous n'avez pas suivi les consignes et avec branché le périphérique à votre PC avant d'installer Crimson 3, ou si vous avez perdu vos nerfs et cliqué sur STOP lorsque l'effrayante fenêtre de pilote non signé est apparue, il est possible qu'une installation annulée ait rendu impossible une installation correcte des pilotes. Pour vérifier cette option, ouvrez le Gestionnaire de périphériques Windows par un clic droit sur l'icône Poste de travail et en sélectionnant la commande Gestion. Une fenêtre similaire à cette dernière devrait apparaître...

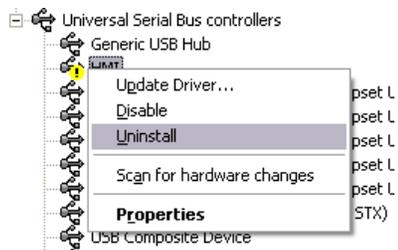


Le processus exact permettant d'accéder à cette fenêtre varie d'un système d'exploitation à l'autre mais l'idée de base est la même : trouver l'icône Poste de travail – soit sur le bureau soit via le Menu Démarrer – effectuer un clic droit et choisir Gestion. Si cette option ne fonctionne pas, sélectionnez l'option Système dans le Panneau de configuration et activez le gestionnaire de périphériques à partir de l'onglet Matériel.

Si vous rencontrez un problème avec vos pilotes USB, vous verrez une icône jaune comportant un point d'exclamation apparaître dans la sous-catégorie contrôleurs USB. Cette dernière peut être appelée HMI, Loader ou autre. L'image ci-dessous présente cette situation...



Pour régler ce problème, faites un clic-droit sur l'élément à réparer et choisissez Désinstaller à partir du menu qui s'affiche alors...



Après avoir demandé confirmation, Windows ôtera cet élément de votre système. Vous pouvez alors débrancher votre périphérique Crimson. Au bout de quelques secondes, rallumez ce dernier et Windows relancera automatiquement le processus d'installation des pilotes.

Comme mentionné ci-dessus, Crimson utilise des pilotes différents pour le loader de base et pour l'environnement runtime Crimson. Il est donc possible que vous ayez à répéter ce processus pour chaque pilote bien qu'il soit peu probable que vous dépassiez l'étape du loader de base si cette installation échoue.

## **ETAPES SUIVANTES**

L'interface utilisateur de Crimson 3 ayant été entièrement repensée, nous suggérons aux utilisateurs de Crimson de parcourir au minimum le chapitre suivant. Nous vous suggérons également de jeter un œil aux chapitres consacrés aux tags et à la configuration de l'affichage des pages, certains concepts utilisés dans Crimson 2 ayant été simplifiés et de nouvelles méthodes plus rapides et intuitives ayant été introduites. Si vous découvrez Crimson, essayez de lire au minimum jusqu'au début du chapitre sur les Widgets.

Bonne chance, et surtout amusez-vous !

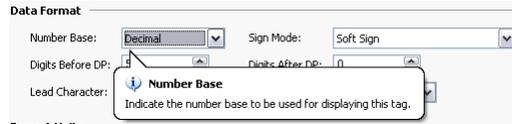


# PRINCIPES DE BASE

Pour lancer Crimson, cliquez sur l'icône Crimson 3 dans la section Red Lion Controls du menu démarrer. Après quelques secondes, le logiciel apparaîtra. La nouvelle interface propose une navigation et une construction de bases de données plus rapides. Nous espérons que cette nouvelle interface vous permettra d'améliorer sensiblement votre productivité.

## INFO-BULLES

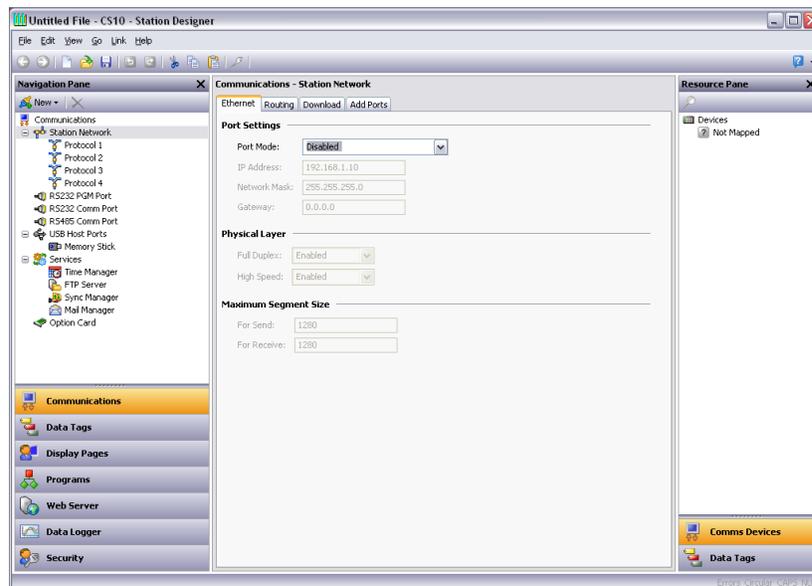
La première fonctionnalité dont nous parleront est le système d'info-bulles...



Ce système vous permet de visualiser les informations correspondant à chaque élément de Crimson. Il est contrôlé par l'icône située dans l'angle supérieur droit de la barre d'outils ou via la commande options du menu Aide. Le mode par défaut permet d'afficher le texte en appuyant sur la touche F1, une méthode rapide d'afficher les informations concernant un paramètre sur lequel vous avez des doutes ou les paramètres concernant un champ donné. Elles ne pourront que vous simplifier la vie ! De manière alternative le mode Sur passage de souris permettra d'afficher l'aide lorsque le pointeur de votre souris est stationné durant un certain temps sur un élément. Le mode Sur sélection affichera toujours l'aide concernant le champ sélectionné.

## APPARENCE DE LA FENETRE

La fenêtre principale de Crimson comprend trois parties ...



## LE PANNEAU DE NAVIGATION

La partie gauche de la fenêtre est appelée Panneau de navigation. Il permet de se déplacer entre les différentes catégories d'éléments composant un fichier Crimson. Chaque catégorie est représentée par une barre située à la base du panneau. Cliquer sur cette barre permettra de naviguer jusqu'à la section correspondante. La section supérieure du panneau de navigation présente les éléments disponibles dans la catégorie actuelle et fournit une barre d'outils permettant la manipulation de ces derniers. Si vous souhaitez agrandir la section supérieure, étirez à l'aide de la souris la ligne de division entre cette dernière et les barres de catégorie.

## LE PANNEAU DE RESSOURCES

La partie droite de la fenêtre principale est appelée Panneau de ressources. Il permet d'atteindre tous les éléments nécessaires lors de l'édition d'une catégorie. Tout comme le panneau de navigation, il contient plusieurs catégories auxquelles il est possible d'accéder via les barres de catégorie. Les éléments d'une catégorie de ressource donnée peuvent être glissés-déposés aux emplacements sur lesquels vous souhaitez les utiliser. Par exemple, un tag de données peut être pris depuis le panneau de ressources et déposé dans un champ de configuration de manière à rendre ce dernier dépendant de la valeur du tag sélectionné. De nombreux éléments peuvent également être activés par double-clic, paramétrant ainsi le champ concerné.

## LE PANNEAU D'ÉDITION

La partie centrale de la fenêtre est utilisée pour l'édition des éléments sélectionnés. En fonction de la sélection, elle peut contenir des onglets dont chacun présente un ensemble de propriétés d'un élément donné, ou un éditeur spécifique à l'élément édité.

## REDUIRE LES PANNEAUX

Le Panneau de configuration et / ou le panneau de ressources peuvent être réduits afin de laisser plus d'espace disponible au panneau d'édition. Pour fermer un panneau, cliquez sur le 'x' situé en haut à droite de sa barre de titre. Il sera alors remplacé par une barre plus petite comportant des flèches. Cliquer sur cette barre permettra d'étendre (ou de restaurer) le panneau associé. Une icône permet de verrouiller l'emplacement d'un panneau temporairement étendu.

## CATEGORIES

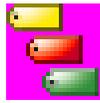
Les principales catégories de la base de données Crimson sont les suivantes...

### COMMUNICATIONS



Cette catégorie spécifie quels protocoles doivent être utilisés sur les ports série et ports Ethernet du périphérique cible. Lorsque les protocoles principaux sont utilisés (c.-à-d. les protocoles pour lesquels le matériel Red Lion initie les transferts de données de et vers le périphérique à distance) vous pouvez également utiliser cette icône pour spécifier un accès vers un ou plusieurs autres périphériques. Vous pouvez également utiliser cette catégorie pour déplacer des données entre les périphériques à distance via le convertisseur de protocole, de manière à configurer les cartes d'expansion et à configurer les services.

## TAGS DE DONNEES



Cette catégorie définit les éléments de données qui deviendront données d'accès au sein des périphériques à distance, ou qui stockeront des informations au sein des périphériques cibles. Chaque tag possède plusieurs propriétés, dont la propriété formatage de données qui précise comment les données contenues dans un tag doivent apparaître au niveau de l'affichage du périphérique ou dans d'autres contextes, comme dans des pages web. En précisant ces informations au sein du tag, Crimson vous permet d'éviter d'avoir à saisir les données de formatage à chaque fois qu'un tag est affiché. D'autres propriétés plus avancées comprennent des alarmes pouvant être activées dans de nombreuses circonstances, ou activations, qui lancent des actions programmables lorsque les conditions prédéfinies sont réunies.

## PAGES D'AFFICHAGE



Cette catégorie vous permet de créer et d'éditer des Pages d'affichage. L'éditeur de pages vous permet d'afficher de nombreux éléments graphiques désignés sous le nom de primitives. Ces dernières peuvent être des éléments simples, rectangles et lignes, ou plus complexes pouvant être liés à la valeur d'un tag ou d'une expression spécifiques. Par défaut, ces primitives utilisent les informations de formatage définies lors de la création du tag – ces dernières pouvant être réécrites si nécessaire. Vous pouvez également utiliser l'éditeur afin de spécifier quelles actions doivent être accomplies lorsque les touches et primitives sont appuyées, relâchées ou maintenues.

## PROGRAMMES



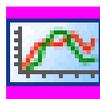
Cette catégorie permet de créer et d'éditer des Programmes, grâce au langage de programmation de type C spécifique à Crimson. Ces derniers peuvent effectuer des opérations complexes de prise de décision ou de manipulation de données, basés sur des éléments de données situés au sein du système. Ils permettent d'étendre les fonctionnalités de Crimson bien au-delà des fonctions standard incluent dans le logiciel, offrant l'assurance que même les applications les plus complexes peuvent être prises en charge.

## SERVEUR WEB



Cette catégorie permet de configurer le serveur web de Crimson et de créer et éditer des pages web. Le serveur web peut fournir un accès à distance au périphérique cible via de nombreux mécanismes. Vous pouvez tout d'abord utiliser Crimson pour créer automatiquement des pages web contenant des listes de tags, chacune formatée en fonction des propriétés de ces derniers. Deuxièmement, vous pouvez créer un site personnalisé à l'aide d'un éditeur HTML tierce partie tel que Microsoft Front Page et inclure par la suite un texte spécial permettant à Crimson d'insérer des valeurs de tags *live*. Finalement, vous pouvez activer l'accès à distance, unique à Crimson, et contrôler les paramètres qui permettent à un navigateur web de visualiser l'affichage du périphérique cible et de contrôler son clavier. Le serveur web peut également être utilisé pour accéder aux fichiers CSV à partir du journal de données.

## JOURNAL DE DONNEES



Cette catégorie permet de créer et d'assurer la gestion de logs de données qui peuvent enregistrer individuellement autant de données qu'il est nécessaire sur la carte CompactFlash du périphérique cible. Les données peuvent être enregistrées à

la vitesse de une par seconde. Les valeurs enregistrées seront stockées au format CSV, facilement importable dans de nombreuses applications comme par exemple Microsoft Excel. Il est possible d'accéder aux fichiers via la carte CF, en l'installant comme lecteur externe sur un PC connecté au périphérique cible par port USB ou en accédant à cette dernière via les serveurs web ou FTP de Crimson à l'aide d'un port Ethernet ou d'un modem.

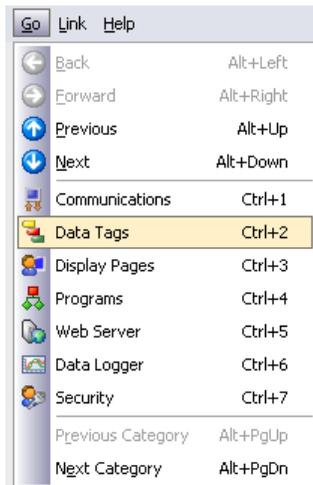
## SECURITE



Cette catégorie est utilisée pour créer et gérer les comptes utilisateurs sur le périphérique cible, ainsi que les droits d'accès liés à ces derniers. Des noms réels peuvent également être donnés, ce qui permet au journal de sécurité non seulement d'enregistrer quelles données ont été modifiées et quand, mais également concernant qui. Les droits nécessaires à la modification d'un tag particuliers ou à l'accès à une page sont définis via les propriétés de sécurité de chaque élément individuel. Des droits peuvent également être assignés afin de permettre ou d'empêcher l'accès aux serveurs web ou FTP.

## NAVIGUER DANS LA BASE DE DONNEES

La manière la plus simple de naviguer dans la base de données Crimson est de cliquer sur les barres de catégories du panneau de navigation, puis de cliquer sur l'élément que vous souhaitez éditer. Toutefois, de nombreux raccourcis permettent une manipulation encore plus rapide et de fait une meilleure productivité. La plupart de ces raccourcis peuvent être accédés via le menu Aller à ou via une combinaison de touches associée...



## RECULER ET AVANCER

La première icône de la barre des tâches, ou la combinaison des touches **ALT+GAUCHE** permettent de reculer vers les éléments précédemment sélectionnés. L'icône suivante, ou combinaison des touches **ALT+DROITE** permet d'avancer à nouveau, revenant à l'élément avec lequel vous avez commencé. Cette option est très utile lorsque vous souhaitez basculer d'une catégorie de bases de données à l'autre.

## RACCOURCIS CATEGORIES

A chaque catégorie est associé un raccourci clavier comprenant la touche **CTRL** et un nombre indiquant la position de la catégorie dans le panneau de navigation. Par exemple, la section Communications peut être ouverte en appuyant sur les touches **CTRL+1**. Vous pouvez

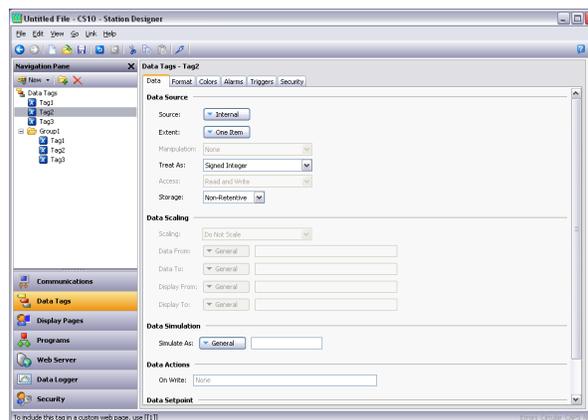
également vous déplacer dans la liste de catégories en utilisant les combinaisons **ALT+HAUT** et **ALT+BAS**.

## RACCOURCIS ELEMENTS

Si vous travaillez dans le panneau d'édition, vous avez la possibilité de basculer entre les différents éléments à l'aide des combinaisons clavier **ALT + HAUT** et **ALT + BAS**. Crimson passera vers l'élément précédent ou suivant de la liste et essaiera de maintenir le champ de données sélectionné. Cette option est très utile si vous souhaitez modifier le même champ pour plusieurs éléments. Ainsi, vous n'avez pas à revenir en permanence au champ ou à basculer vers le panneau de navigation pour modifier les éléments.

## LISTES DE NAVIGATION

Plusieurs catégories de Crimson contiennent des listes d'éléments. Par exemple, lors de la sélection de la catégorie Tags de données le panneau de navigation affichera une liste de tous les tags de données présents dans votre base de donnée, permettant de sélectionner et d'éditer ces derniers...



Les éléments présentés dans les listes de navigation peuvent être manipulés de plusieurs manières...

- Pour rechercher rapidement un élément, saisissez les premières lettres de son nom. Crimson sélectionnera le premier élément qui correspond aux caractères que vous avez saisis. Saisir plus de caractères permettra d'affiner la sélection, et appuyer sur la touche **ECHAP** permettra de saisir une nouvelle séquence de caractères.
- Pour créer un élément, cliquez sur le bouton Nouveau dans la barre de navigation du panneau. Les listes qui ne prennent en charge qu'un seul type d'objet, vous pouvez également utiliser les touches **ALT + INS**. Le bouton Nouveau de la barre d'outils mai offre une liste d'articles disponibles, vous permettant de choisir le type de l'élément que vous souhaitez créer.
- Pour supprimer un élément, utilisez soit l'icône Supprimer situé dans la barre d'outils du panneau de navigation, soit les touches **ALT + SUPPR**. Si vous supprimez un dossier, tous les éléments contenus dans ce dernier seront également supprimés. Le système affiche des avertissements pour les suppressions multiples qui peuvent toujours être annulées – via la commande **Annuler**.

- Pour renommer un élément, sélectionnez ce dernier et appuyez sur la touche **F2**. Vous pouvez alors saisir le nouveau nom puis appuyer sur la touche **ENTREE**. De manière alternative, sélectionnez l'élément puis effectuez un clic droit sur son nom. Appuyez alors à nouveau sur **ENTREE** lorsque vous avez terminé.

### **TRAVAILLER AVEC LES DOSSIERS**

Certaines listes supportent la possibilité de grouper les éléments en dossiers. Les dossiers peuvent être créés à l'aide de l'icône Nouveau Dossier située dans le panneau de navigation, et peuvent être renommés et supprimés comme les autres éléments. La création d'un élément dans un dossier sélectionné placera automatiquement ce dernier dans le dossier.

### **CLASSEMENT DES LISTES ET DES DOSSIERS**

Une liste de toute la navigation ou le contenu d'un dossier peut être triés par un clic droit sur l'élément racine ou le dossier le cas échéant, et en sélectionnant l'une des commandes de tri. Mais articles sont triés par ordre croissant ou décroissant ordre alphabétique. Les dossiers sont toujours placés avant les autres éléments, quel que soit l'ordre de tri est appliqué.

### **OPERATIONS DE GLISSER / DEPOSER**

Les éléments des listes des articles de navigation peuvent être glissés-déposés dans la liste pour modifier leur position ou de les déplacer entre les dossiers. Maintenant enfoncée la touche **CTRL** enfoncée tout en faisant glisser se traduira par une copie de l'élément d'origine en cours de création. La gauche à la position droite d'un élément mais parfois être utilisés pour décider où placer un élément dans la hiérarchie des dossiers. Si l'élément est placé dans le mauvais dossier, essayez de déplacer à gauche ou à droite pour arriver à la bonne position.

Les éléments de base de données, comme les tags, Pages d'affichage ou autre chose, mais aussi être glissé entre les fichiers de base de données par l'ouverture de deux copies de Crimson et en faisant glisser les éléments en question du volet de navigation de la base de données source avec celle de la base de données cible. Si la catégorie appropriée dans la cible n'est pas déjà sélectionnée, maintenir temporairement l'élément qui est glissé sur la barre de catégorie nécessaire pour une seconde ou deux sélectionnera cette catégorie, ce qui évite d'interrompre et de répéter l'opération de glisser.

### **RECHERCHE DANS LES LISTES**

Alors que le raccourci détaillé ci-dessus est utile pour sauter directement à un article unique, vous mais ont parfois envie de trouver tous les articles qui ont des noms contenant une chaîne particulière. Cela peut être accompli en utilisant la commande Rechercher article repris sur la barre d'outils le Panneau de navigation. Cette commande va rechercher dans la liste actuelle, et qui placent tous les articles correspondants dans la liste globale des résultats de recherche. Vous pouvez parcourir cette liste en utilisant le **F4** et les combinaisons de touches **MAJ + F4**, ou afficher la liste dans son intégralité en appuyant sur **F8**. Pour plus d'informations sur les fonctions de recherche globale, reportez-vous à la section plus loin dans ce chapitre.

### **ANNULER ET RETABLIR**

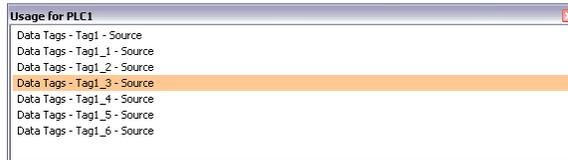
Crimson 3 met en place la possibilité de défaire et refaire la structure. Cela signifie que vous pouvez charger une base de données, le modifier pendant des heures, puis le retourner à son état initial en tenant simplement les touches **CTRL + Z** combinaison de touches. Vous pouvez ensuite ré-appliquer vos modifications en appuyant sur **CTRL + Y**. Toutes vos actions sont

rappelons-le, et Crimson va naviguer entre les articles et catégories automatiquement en marche arrière ou re-mise en œuvre de changements.

## RECHERCHE GLOBALE

Crimson fournit plusieurs options pour effectuer des recherches dans une base de données. Au niveau le plus simple, vous pouvez rechercher une chaîne de texte n'importe où dans la base de données en appuyant sur les touches **CTRL + F + SHIFT** combinaison de touches. Sinon, comme vous le verrez plus loin, vous pouvez chercher des expressions qui contiennent des erreurs, ou des éléments qu'un tag référence ou un outil de communication. Toutes ces opérations de recherche le lieu de leur production dans la liste globale Résultat de la recherche, vous permettant d'examiner les résultats ou pour naviguer dans les deux sens entre les éléments qui ont été localisés.

La liste des résultats peuvent être affichée à tout moment en appuyant sur la touche **F8**...



La barre de titre de la fenêtre décrit l'opération de recherche qui a produit la liste, tandis que chaque ligne contient la description d'un élément qui correspondent aux critères de recherche. Dans l'exemple ci-dessus, un clic droit sur un dispositif de communication et en sélectionnant la commande détecter les utilisations énumérées tous les lieux où l'appareil a été référencé. Double-clic sur une entrée donnée sauter directement à cette question, tandis que le **F4** et **MAJ + F4** combinaisons de touches peuvent être utilisées à l'étape d'avant en arrière dans la liste. Les commandes associées à cette fonctionnalité peuvent également être consultées via le menu Edition.

## TRAVAILLER AVEC LES BASES DE DONNEES

Crimson stocke toutes les informations sur une configuration particulière dans ce qu'on appelle un fichier de base de données. Ces fichiers portent l'extension de `cd3`, bien que Windows Explorer permet de masquer cette extension si elle est laissée dans sa configuration par défaut. Alors que Crimson 3 bases de données sont encore essentiellement des fichiers texte, ils sont compressés et ne peuvent donc pas être directement édités en utilisant un éditeur de texte comme Notepad. Comme on peut s'y attendre, les bases de données sont manipulées via les commandes se trouvant dans le menu Fichier. La plupart de ces commandes sont standards pour toutes les applications Windows, et ne nécessitent pas plus d'explications.

### IDENTIFIANTS BASE DE DONNEES

Chaque base de données créée par Crimson se voit attribuer un identifiant unique. Cet identifiant est utilisé lors du téléchargement d'une nouvelle base de données pour déterminer si le périphérique cible devrait effacer de sa mémoire interne et de suppression des fichiers journaux enregistrés pour CompactFlash. Si l'identifiant correspond à celui de la base de données déjà dans le dispositif, la base de données est supposée être simplement une version différente du même fichier, afin que les données soient conservées. Inversement, si les identifiants sont différents, les données sont effacées. Lorsque vous utilisez la commande Enregistrer sous dans le menu Fichier pour enregistrer une copie d'un fichier de base de données, Crimson vous demandera si vous voulez affecter un nouvel identificateur. Sélectionnez Oui si cela va être un nouveau projet, et sélectionnez Non si vous enregistrez simplement une copie de sauvegarde de ce qui est essentiellement la même base de données. Cela assurera que les données de rétention du périphérique cible est claire ou conservées, comme c'est le cas.

## ENREGISTRER UNE IMAGE

Le point spécifique à Crimson sur le menu Fichier est Enregistrer l'image. Cette commande permet la création d'un fichier qui peut ensuite être utilisé pour mettre à jour la base de données dans un terminal via CompactFlash ou en option via une clé mémoire USB. Le fichier contient une forme non modifiable de la base de données, plus les mises à jour de firmware et boot loader requise pour exécution. Insertion d'une image fichier appelé `image.ci3` dans le répertoire racine de la carte CompactFlash de l'appareil cible et remettre en place alors le dispositif se traduira par le chargeur de démarrage, le firmware et base de données mis à jour en utilisant le contenu du fichier image. Notez que les fichiers image peut éventuellement contenir télécharger des informations, permettant une version modifiable du fichier base de données susceptible d'être extraite à partir d'un terminal qui a été mis à jour en utilisant l'image.

## PROTECTION DES BASES DE DONNEES

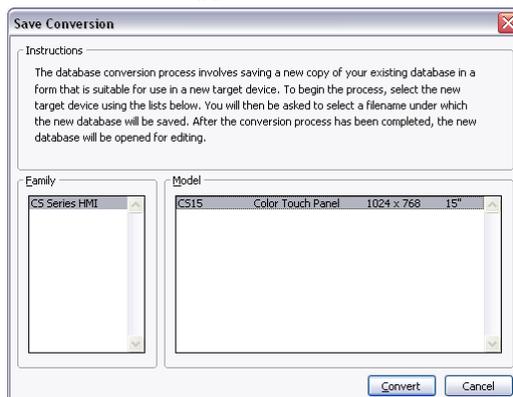
Les bases de données peuvent être protégées par mot de passe en utilisant la commande Protection dans le menu Fichier...



Le paramètre d'accès par défaut est utilisé pour définir le niveau d'accès ne sera permis sans entrer le mot de passe base de données. Un réglage de Read-Only Access permettra à la base de données pour être ouvert, mais ne permettra pas que des modifications soient apportées ou sauvegardé. Une valeur de n L'accès sera interdit tout accès sans le mot de passe.Par défaut, le plein accès permettra à la base de données pour être ouvert pour l'édition sans mot de passe a été saisie.Mots de passe perdus peuvent être récupérés par Red Lion Controls pour un prix minime.

## CONVERTIR UNE BASE DE DONNEES

Une base de données conçue pour un périphérique cible peut être converti pour utiliser sur un autre en utilisant la commande Enregistrer de conversion dans le menu Fichier. Les conversions qui peut être réalisée dépendra de l'appareil cible d'origine, mais la plupart des combinaisons sont supportés.

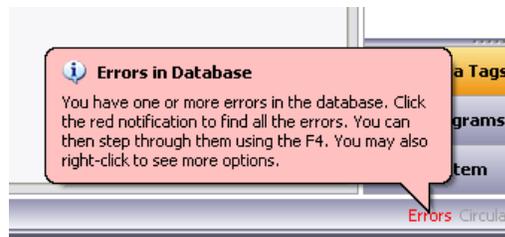


Le processus de conversion est lancé en sélectionnant le nouveau périphérique cible en utilisant la boîte de dialogue ci-dessus. Vous serez alors invité à entrer un nouveau nom de fichier, et la base de données convertie sera sauvegardé sur le disque. Afin d'éviter la destruction accidentelle des bases de données existantes, vous mai pas convertir une base de données sans l'enregistrer sous un nouveau nom. Une fois la base de données convertie a été sauvegardé, il sera ouvert pour l'édition et de révision.

Le processus de conversion redimensionne les pages d'affichage pour s'adapter au nouveau format d'affichage et les communications remappe dispositifs pour les ports appropriés sur le nouveau dispositif fondé sur qu'ils utilisent l'interface RS-232 ou RS-485 de la couche physique. Il mai pas être possible de convertir une base de données dans son intégralité si, par exemple, le nouvel appareil a moins de port de communication que l'original. Vous mai donc avoir à effectuer quelques ajustements après la conversion.

### RECHERCHE D'ERREURS

Certaines opérations peuvent produire des erreurs dans votre base de données. Par exemple, vous mai supprimer un appareil de communication, ou que vous mai définir un tag correspond à une expression basée sur elle-même, produisant ainsi une référence circulaire. Crimson vous mettre en garde contre de telles erreurs au moyen d'un ballon rouge qui apparaît au-dessus de la barre d'état...



L'info-bulle va s'estomper après quelques secondes, mais le feu rouge dans la barre d'état restera pour vous rappeler de la condition d'erreur. En cliquant sur l'indicateur de rechercher toutes les erreurs ou les références circulaires, et les placer sur la liste mondiale Résultats de la recherche afin que vous mai examen à l'aide du **F4** standard et les combinaisons de touches **MAJ + F4**. Vous mai également le bouton droit sur l'indicateur pour accéder aux commandes de recompiler la base de données entière, ou pour optimiser la façon dont les communications appareil sont organisés. Recompilation base de données manuelle est rarement nécessaire, comme Crimson exerce généralement les mesures nécessaires sans intervention d'utilisateur.

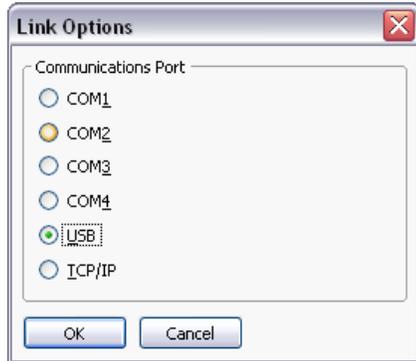
### TELECHARGEMENT VERS UN PERIPHERIQUE

Les fichiers de base de données Crimson sont téléchargés dans le périphérique cible à l'aide du menu Lien. Le processus de téléchargement prend généralement que quelques secondes, mais peut prendre un peu plus longtemps sur le télécharger au préalable si Crimson doit mettre à jour le microprogramme de l'appareil, ou si l'appareil ne contient pas une version plus ancienne de la base de données actuelle. Après ce premier téléchargement, Crimson utilise un procédé connu sous le nom de téléchargement supplémentaire pour s'assurer que seuls les changements à la base de données sont transférées. Cela signifie que les mises à jour peuvent être faites en quelques secondes, réduisant ainsi votre temps de cycle de développement et de simplifier le processus de débogage.

### CONFIGURATION DU LIEN

Le lien de programmation entre le PC et le périphérique cible peuvent être effectué en utilisant un port RS-232, un port USB ou une connexion TCP / IP. Bien que les connexions

TCP / IP sont généralement effectuées par l'intermédiaire du port Ethernet du panel, ils peuvent également être établies par une ligne commutée en lien. Avant de télécharger, utilisez le lien Options de la commande de veiller à ce que vous avez choisi la bonne méthode...



Notez que cette boîte de dialogue ne fournit aucune méthode pour sélectionner une adresse IP cible lorsque vous utilisez TCP / IP pour le téléchargement. Cette information est maintenant stockée dans le fichier de base de données et est configurée via l'onglet de téléchargement de l'élément de configuration réseau. Cette méthode rend plus facile de basculer entre plusieurs bases de données sans avoir à re-configurer l'IP cible à chaque fois.

### ENVOI DE LA BASE DE DONNEES

Une fois le lien econfiguré, les bases de données peuvent être téléchargées en utilisant soit le Link-Envoyer ou Link-commandes Update. L'ancienne allez envoyer la base de données entière, qu'ils soient ou non des objets individuels dans le fichier a changé. Ce dernier enverra seulement des changements, et se fera généralement une plus courte période de temps à compléter. La commande Update est généralement la seule que vous aurez besoin, comme la pourpre installateur reviendra à un système complet envoyer si le téléchargement incrémentiel échoue pour quelque raison. Comme un raccourci, notez que vous pouvez accéder Link-Mise à jour par le symbole de l'éclair boulon sur la barre d'outils, ou via la touche **F9** sur le clavier.

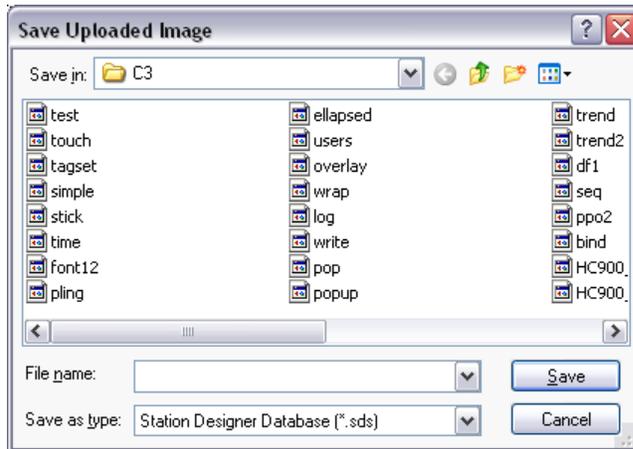


Notez que le téléchargement via TCP / IP est basé sur une carte CompactFlash en cours d'installation dans le panneau, si le firmware de l'appareil doit être mis à niveau. Depuis mai, vous souhaitez effectuer ces mises à niveau à un moment donné dans le temps, il est fortement recommandé que vous installiez une carte CompactFlash dans n'importe quel périphérique à laquelle le protocole TCP / IP téléchargements sont susceptibles d'être réalisées. Notez également que le protocole TCP / download IP doit être activé via les paramètres de réseau dans la catégorie Communications.

### EXTRACTION DE BASES DE DONNEES

L'adossement de soutien au commandement Charger peut être utilisé pour charger Crimson quant à savoir si elle doit ou non inclure les informations nécessaires au soutien de télécharger la base de données lorsque vous envoyez une base de données pour un

périphérique cible. Ce paramètre est stocké dans la base de données, et peut donc être configuré sur un fichier par fichier. Upload Soutenir va ralentir le processus de téléchargement quelque peu et peut échouer avec les bases de données extrêmement volumineuses contenant de nombreuses images incorporées, mais il veillera à ce que, si vous perdez votre fichier de base de données, vous serez en mesure d'extraire une image modifiable à partir du périphérique.



Notez que si vous perdez votre fichier de base de données et que vous n'avez pas de soutien upload activé, vous ne serez pas en mesure de reconstituer votre fichier sans partir de zéro. Pour extraire une base de données à partir d'un panneau, utiliser le Link-commande Extract. Cette commande va télécharger la base de données, et vous invite à saisir un nom sous lequel vous voulez enregistrer le fichier. Le fichier sera alors ouvert pour l'édition. Si la base de données est protégée par mot de passe, vous peut être nécessaire d'entrer le mot de passe avant si on peut ouvrir. En d'autres termes, ce qui permet en amont passera pas contourner la protection de mot de passe.

## MISE EN PLACE DE LA CARTE COMPACTFLASH

Si vous êtes connecté à un dispositif approprié via le port USB, vous pouvez demander à Crimson de monter la carte CompactFlash de l'appareil comme un disque dans Windows Explorer. Vous pouvez utiliser cette fonctionnalité pour enregistrer des fichiers sur la carte ou pour lire les informations du Journal de données. Le disque est monté et démonté en envoyant des commandes en utilisant le flash monter / démonter des options de flash dans le menu Lien. Une fois qu'une commande a été envoyée, le dispositif cible sera remis à zéro, et Windows se rafraîchira les fenêtres appropriées Explorer.



Certaines précautions doivent être prises lors du montage de la carte CompactFlash...

- Lorsque la carte est montée, l'appareil cible informera périodiquement le PC si les données sur la carte a été modifiée. Cela signifie que le PC et l'appareil subira une performance mineurs frappés si la carte est montée au cours des opérations d'enregistrement de données plus longtemps que nécessaire.
- Si vous écrivez à la carte CompactFlash de l'ordinateur, l'appareil cible ne sera pas en mesure d'accéder à la carte jusqu'à ce que Windows libère son verrou sur la carte. Cette mai prendre plusieurs secondes, et limitera les opérations d'enregistrement de données pendant ce temps, et d'empêcher l'accès aux pages web personnalisée. Crimson utilise la RAM de l'appareil afin de s'assurer qu'aucune donnée n'est perdue, mais si un trop grand nombre écrit sont exécutés tels que la carte est gardé sous clef pendant quatre minutes ou plus, les données de mai jeté.
- Vous ne devriez jamais enter d'utiliser Windows pour formater une carte CompactFlash que vous avez monté via Crimson, que ce soit via l'Explorateur ou à partir de l'invite de commande. Windows ne pas se verrouiller correctement la carte durant les opérations de format, et le format de mai donc ne pas être fiable et le plomb à subséquente perte de données. Voir ci-dessous pour plus de détails sur la façon de formater une carte d'une manière fiable.

## FORMATAGE DE LA CARTE COMPACTFLASH

La seule méthode de formatage d'une carte se fait via la commande en format Flash dans le menu Lien. La sélection de cette commande va expliquer que le processus de formatage, vous détruisez toutes les données stockées sur la carte CompactFlash et vous offre la possibilité d'annuler l'opération. Si vous choisissez de continuer, le panneau d'opérateur sera chargé de formater la carte. Notez que ce processus mai prendre plusieurs minutes pour une grande carte. Slow formats sur les panneaux qui effectuent l'enregistrement de données mai donc aboutir à des lacunes dans les données enregistrées.

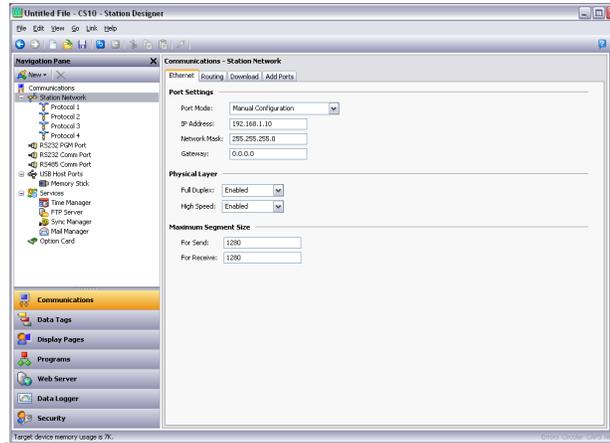
## ENVOI DE L'HEURE ET DE LA DATE

La commande envoi d'heure et de date peut être utilisée pour régler l'horloge du périphérique cible à correspondre à celle du PC sur lequel Crimson est exécuté. Cette commande envoie également le fuseau horaire et l'heure avancée Réglage de l'heure sur le périphérique cible, afin que les fonctionnalités avancées de l'époque directeur doit être utilisé. Assurez-vous que l'horloge de votre PC est correctement paramétrée avant de faire ça!



# UTILISATION DES COMMUNICATIONS

L'étape suivante est celle de la configuration des ports des communications du module maître afin d'indiquer les protocoles que vous souhaitez utiliser et les périphériques distant auxquels vous voulez accéder. Ces opérations s'effectuent à partir de la fenêtre Communications.



Comme vous pouvez le voir, la fenêtre Communications répertorie les ports de l'unité disponibles sous la forme d'une arborescence. Les modules maîtres sont dotés de trois ports série principaux et vous pouvez ajouter deux ports supplémentaires sous la forme d'une carte d'extension. Target devices may also provide one or two Ethernet ports capable of running several communications protocols simultaneously.

## CHOIX DU PORT SERIE

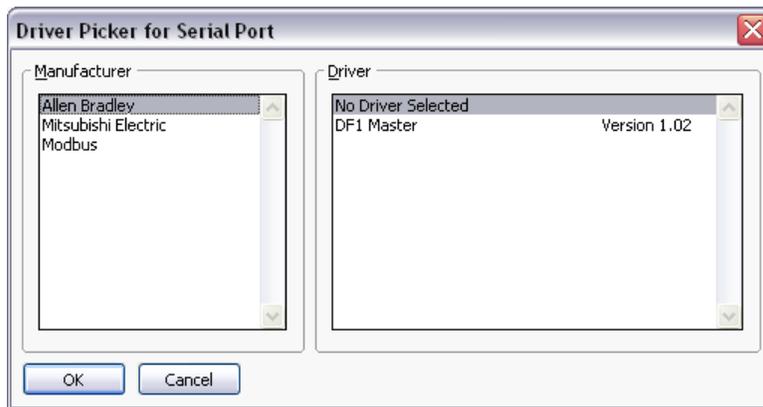
Au moment de décider lequel des ports série de l'appareil cible à utiliser pour les communications, notez que certains appareils (et certaines cartes optionnelles) multiplexent un contrôleur de série unique entre les ports. Cela implique que si l'un ou l'autre des ports est utilisé pour un protocole esclave, l'autre port ne seront pas disponibles, et que si un protocole de type DH-485 est employé l'autre port sera également désactivé. Crimson vous avertira si vous essayez de créer une configuration qui enfreint ces règles.

Notez également que le port d'un dispositif de programmation cible peut être utilisé comme un port de communication supplémentaire, mais qu'il ne sera pas disponible au téléchargement dans ces circonstances. Ce n'est pas un problème si l'appareil possède un port USB qui est utilisé pour les téléchargements, et il est donc fortement recommandé d'utiliser cette méthode si vous voulez connecter des appareils via le port de programmation. Lorsque le port USB n'est pas utilisé, vous devez fournir une méthode pour réactiver les téléchargements de série par l'exécution de la commande `StopSystem ()` en réponse à une action utilisateur.

## SELECTIONNER UN PROTOCOLE

Pour sélectionner un protocole pour un port particulier, cliquez sur l'icône du port, dans le Panneau de navigation et appuyez sur le bouton Choisir à côté du champ Pilote dans le Panneau d'édition.

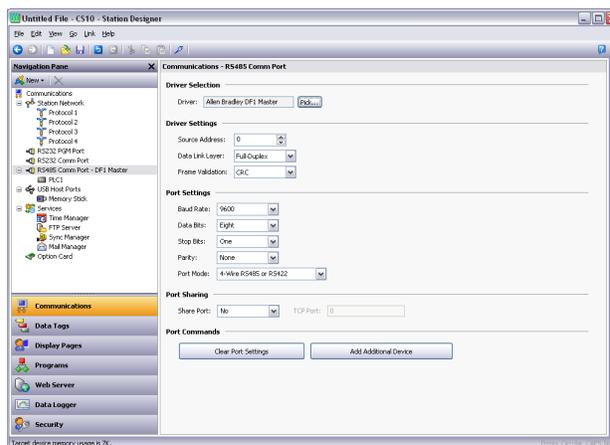
La boîte de dialogue suivante s'affiche...



Sélectionnez le fabricant et le pilote appropriés et appuyez sur le bouton OK pour fermer la boîte de dialogue. Le port est alors configuré pour utiliser le protocole approprié et une seule icône de périphérique est créée dans le volet gauche. Si vous configurez un port série, les différents champs Paramètres du port (Débit en bauds, Bits de données, Bits d'arrêt et Parité) sont définis sur les valeurs correspondant au protocole en question. Vous devez évidemment vérifier ces paramètres pour vous assurer qu'ils correspondent aux paramètres du périphérique en question.

## OPTIONS DU PROTOCOLE

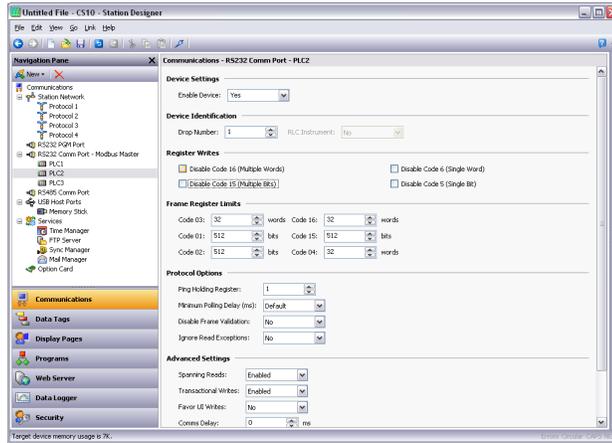
Certains protocoles nécessitent une configuration supplémentaire de paramètres spécifiques. Ils apparaissent dans le Panneau d'édition lorsque l'icône de port correspondant est sélectionnée. L'exemple ci-dessous montre les paramètres additionnels pour un pilote Allen-Bradley DF-1 qui figurent sous la section Paramètres du pilote du Panneau d'édition...



## TRAVAILLER AVEC DES PERIPHERIQUES

Comme nous venons de l'expliquer, lorsque vous sélectionnez un protocole de communication, un seul périphérique est créé sous l'icône du port correspondante. Dans le cas d'un protocole maître, cela représente le périphérique à distance initial qui est adressé via le protocole. Si le protocole prend en charge l'accès à plusieurs périphériques, vous pouvez utiliser le bouton Ajouter un périphérique supplémentaire qui est compris dans les propriétés de l'icône du port afin d'ajouter d'autres périphériques cibles. Chaque périphérique est

représenté via une icône dans le volet gauche de la fenêtre Communications et, selon le protocole en question, peut posséder plusieurs propriétés qui doivent être configurées...



Dans l'exemple ci-dessus, le protocole Modbus Universal Master a été sélectionné, et deux appareils supplémentaires ont été créés, ce qui indique qu'un total de trois appareils à distance doivent être consultés. Le Panneau d'édition présente les propriétés de chaque périphérique. La propriété Activer le périphérique est présente pour tous les appareils avec le solde des champs sont spécifiques au protocole qui a été sélectionné. Notez que les dispositifs sont les noms donnés par défaut par Crimson quand ils sont créés. Ces noms peuvent être modifiés en sélectionnant l'icône appropriée dans le Panneau de navigation, en appuyant sur F2 et en tapant le nouveau nom du périphérique.

## PARAMETRES AVANCES

En plus des réglages de l'appareil mentionné ci-dessus, ce dispositif offre également un certain nombre de paramètres avancés qui peuvent être utilisés pour optimiser les communications comportementales...

- L'option *Spanning Reads* spécifie si Crimson pourra optimiser les opérations de lecture par la lecture des blocs de données, même si ces ne sont pas actuellement sur le point comm scan ou référencés dans la base de données. Par exemple, Go To se lit comme activée pour une base de données qui enregistre les références D1, D2 et D4, une seule commande sera émise sur quatre registres à partir de D1.
- Les options *Ecritures transactionnelles* spécifient si une série de modifications à une valeur de données dans Crimson devrait aboutir à une série correspondante d'opérations d'écriture, ou si seulement la valeur écrite sera transférée.
- Les options *Ecriture favorite interface utilisateur* spécifient s'il faut donner la priorité aux opérations d'écriture qui résultent directement des actions des utilisateurs. Ceci est utile lorsque vous travaillez avec une base de données qui effectue un grand nombre de communications de base en raison de ses activités de conversion de protocole ou programmatiques.
- L'option *Comms Delay* indique un délai qui sera inséré entre deux opérations de communication pour cet appareil. Cette dernière est utile lorsque vous travaillez

avec des périphériques distants qui sont incapables de suivre avec de bonnes performances Crimson, ou quand une baisse de priorité comm doit être donnée à un périphérique.

## UTILISATION DU PORT ET DU PERIPHERIQUE

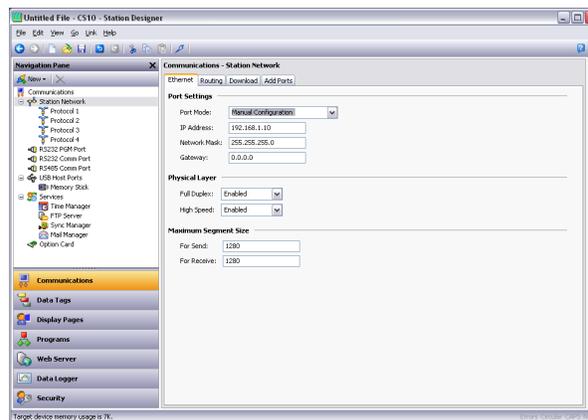
Vous pouvez trouver tous les éléments qui font référence à un dispositif de communication ou à l'un des périphériques connectés à un port particulier par un clic droit sur cette options dans le Panneau de navigation et en sélectionnant la commande Rechercher. Les éléments résultant seraont affichés en Résultat Global de recherche des listes, et peuvent être consultés au moyen de la touche **F4** et des combinaisons de touches **MAJ + F4**. La liste elle-même peut être affichée ou masquée en appuyant sur **F8**.

## CONFIGURATION RESEAU

Le dispositif cible la configuration IP du réseau est édité par l'icône Réseau dans le Panneau de navigation. Lorsque l'icône est sélectionnée, le Panneau d'édition affiche plusieurs onglets, chacun d'eux permettant à un ensemble donné de propriétés d'être configuré.

### PARAMETRES ETHERNET

Les premiers onglets permettent de configurer les ports Ethernet du périphérique cible...



### PARAMETRES DE PORT

Le champ Mode Port contrôle si le port est ou non ativé, et la méthode selon laquelle le port peut obtenir sa configuration IP. Si DHCP est sélectionné, l'appareil cible tentera d'obtenir une adresse IP et les paramètres associés à partir d'un serveur DHCP sur le réseau. Si le DHCP a échoué, une adresse IP sera attribuée automatiquement en utilisant APIPA. (Si l'appareil est configuré pour utiliser des Protocoles esclaves ou pour générer des pages web, cette option n'a de sens que si le serveur DHCP est configuré pour renvoyer une adresse IP connue à l'adresse MAC de l'unité, car autrement, les utilisateurs pourront ne sauront pas comment contacter le périphérique!)

Si le mode le plus commun de configuration manuelle est sélectionné, l'adresse IP, le masque de réseau et les champs de la passerelle doivent être remplis avec les informations appropriées. Les valeurs par défaut fournies à ces domaines ne seront presque jamais adaptées à votre application! Assurez-vous de consulter votre administrateur réseau lors de la sélection des valeurs appropriées, et soyez sûr d'entrer ces valeurs avant de connecter le périphérique

cible à votre réseau. Si vous ne le faites pas, il est possible, quoique peu probable, que vous risquez de causer des problèmes sur votre réseau.

Sélection IEEE 802.3 Ce mode permettra au port une communication de bas niveau, mais ne sera pas affecté à une adresse IP, TCP ou opérante. Ce mode n'a de sens que lorsque vous utilisez les pilotes sur Ethernet bruts, comme les certains protocols automatisés.

### COUCHE PHYSIQUE

Les options de la couche physique contrôlent le type de connexion que le maître essaie de négocier avec le hub auquel il est connecté. En règle générale, vous pouvez laisser ces options à leur état par défaut, mais si vous rencontrez des problèmes lors de la création d'une connexion fiable (principalement lorsque vous vous connectez directement à un PC sans l'intervention d'un hub ou d'un switch), essayez de désactiver les opérations Full Duplex et haute vitesse pour voir si elles résolvent le problème.

### TAILLE MAXIMALE DU SEGMENT

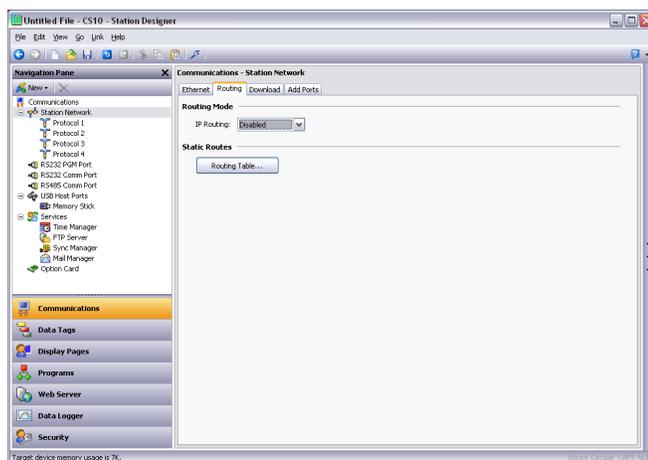
Les options de taille maximale du segment permettent de contrôler les paramètres TCP MSS pour envoi et réception. Vous n'aurez généralement pas à modifier ces paramètres, les valeurs par défaut convenant pour pratiquement toutes les applications et tous les réseaux.

### PORTS MULTIPLES

Si vous utilisez plus d'un port Ethernet, notez que seulement un seul port doit avoir une passerelle par défaut définie, et que chaque port doit avoir une adresse réseau spécifique. Crimson n'enverra jamais qu'un paquet de données à une IP donnée vers une seule interface. Ainsi une configuration qui, par exemple, définit le port Ethernet d'abord comme 192.168.100.1 et le second comme 192.168.100.2 entraînera dans tous les paquets pour le réseau 192.168.100.0 va dans le premier port, empêchant ainsi le deuxième port de fonctionner correctement.

### PARAMETRES DE ROUTING

Le deuxième onglet permet de configurer les options de routage Ethernet...



### MODE ROUTING

L'option de routage IP est utilisée pour activer ou désactiver les paquets de routage entre les interfaces. Si cette option est activée, les paquets IP reçus sur un port Ethernet ou modem qui

sont destinés aux les appareils raccordés à un autre port seront transmis selon les besoins. La désactivation de cette option permettra d'éviter cette transmission. Le réglage nécessaire sera fait en fonction de la topologie de votre réseau.

### TABLE DE ROUTING

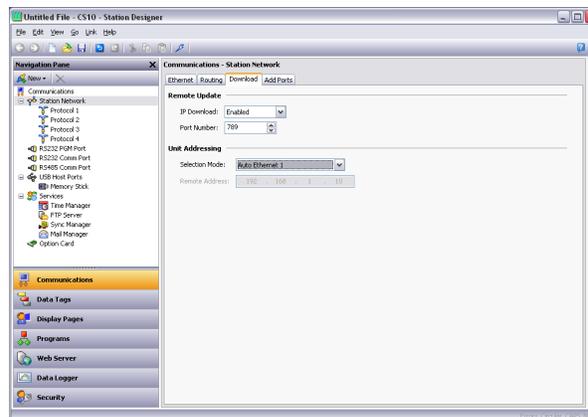
La table de routage définit des routes statiques supplémentaires pour stack TCP / IP Crimson.



Dans l'exemple ci-dessus., un seul itinéraire a été spécifié. Crimson demande de transmettre tous les paquets destinés à des adresses IP commençant par 192.168.3 au routeur situé sur le réseau local à l'adresse 192.9.200.8. Une fois encore, les paramètres exacts requis seront fonction de la topologie du réseau auquel le périphérique cible est connecté.

### PARAMETRES DE TELECHARGEMENT

Le troisième onglet permet de configurer des téléchargements vers le périphérique cible sur TCP / IP...



### MISE A JOUR A DISTANCE

L'option Télécharger IP est utilisée pour activer ou désactiver le protocole TCP / IP téléchargements, tandis que l'option Numéro de port spécifie le port TCP devant être utilisé pour ce type de téléchargement. La valeur par défaut de 789 ne devrait pas être modifiée sans raison solide.

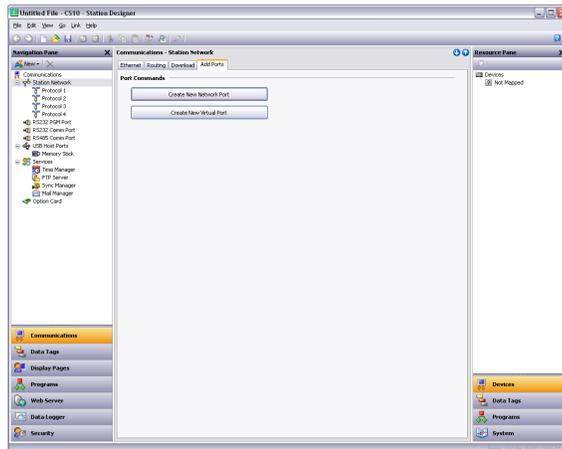
### UNITES CONSULTEES

Ces paramètres sont utilisés pour spécifier l'adresse IP utilisée par le logiciel de configuration Crimson lorsque la méthode de téléchargement TCP est sélectionné dans le lien Options de la boîte de dialogue. Le mode Auto utilise l'adresse IP configurée pour le port Ethernet sélectionné. (Bien entendu, le port doit être configuré manuellement pour cela ait un sens.) Le

mode manuel permet d'inscrire une adresse IP via le champ d'adresses distantes. Notez que cette information est sauvegardée dans le cadre de la base de données, vous permettant de basculer facilement entre les unités sur le même réseau.

## AJOUT DE PORTS

Le quatrième onglet peut être utilisé pour ajouter des protocoles réseau supplémentaires...



En appuyant sur le bouton Créer un nouveau Réseau de Port, vous ajouterez un protocole réseau supplémentaire, à concurrence du nombre maximal de ports pris en charge par le dispositif cible. En appuyant sur le bouton Créer un nouveau port virtuel vous effectuerez une opération similaire, mais ajouterez un port capable d'émuler une connexion série sur TCP / IP. Chaque type de port peut être supprimé en le sélectionnant dans le Panneau de navigation et en appuyant sur **ALT + DEL** ou en sélectionnant l'option « supprimer » depuis la barre d'outils.

## SELECTION DE PROTOCOLE

Une fois que vous avez configuré le port Ethernet, vous pouvez sélectionner les protocoles que vous souhaitez utiliser pour vos communications. Vous pouvez utiliser simultanément jusqu'à quatre protocoles et de nombreux protocoles prennent en charge plusieurs périphériques distants. C'est-à-dire que plusieurs options s'offrent à vous lorsque vous décidez de la façon d'associer les protocoles et les périphériques afin d'obtenir les résultats que vous souhaitez.

Supposez par exemple que vous vouliez vous connecter à deux périphériques esclaves à l'aide de Modbus sur TCP / IP. Votre première option consiste à utiliser deux sockets (protocole1 et protocole2) du port Ethernet et à les configurer comme deux maîtres TCP / IP Modbus, avec un seul périphérique connecté à chaque protocole. Pour la plupart des protocoles, cette option permettra d'obtenir des performances supérieures car les communications simultanées seront autorisées avec les deux périphériques. Toutefois, cela consommera deux des quatre protocoles, limitant ainsi votre capacité à vous connecter via d'autres protocoles dans des applications complexes.

La seconde option consiste donc à utiliser un seul socket (Protocole1) configuré comme un maître TCP / IP Modbus et à ajouter un autre périphérique de façon à ce que vous puissiez accéder aux deux esclaves via le même pilote. Cette option offrira des performances légèrement réduites car Crimson interrogera chaque périphérique l'un après l'autre, au lieu de communiquer avec les deux simultanément. Cependant, elle permettra de conserver les sockets Ethernet, autorisant d'autres applications complexes sans manquer de ressources.

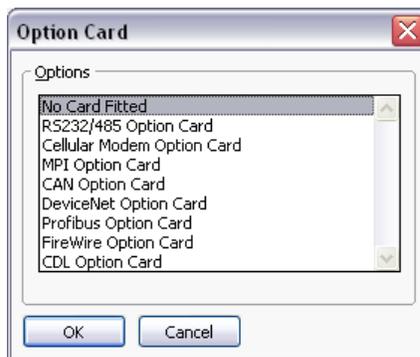
## UTILISATION DES PORTS VIRTUELS

Comme mentionné ci-dessus, Crimson supporte l'ajout de ports virtuels à la configuration du réseau. Un port de communication virtuel fonctionne comme un port série, mais envoie et reçoit ses données via une liaison TCP / IP. Les ports virtuels peuvent être configurés soit en mode actif soit en mode passif. Dans le premier cas, Crimson tentera d'ouvrir une connexion TCP / IP à un périphérique distant spécifié, tandis que dans le deuxième cas, Crimson va écouter sur un réseau TCP / IP spécifique les connexions entrantes. Les ports virtuels sont généralement utilisés pour communiquer avec des périphériques via des serveurs à distance en série : un protocole série standard est utilisé, mais les données de protocole sont encapsulées dans des paquets TCP / IP.

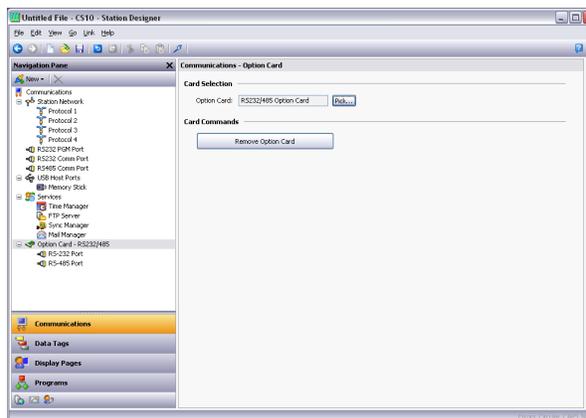
## UTILISATION DES CARTES D'EXPANSION

Certains dispositifs cibles supportent l'ajout de cartes d'extension. Un certain nombre de cartes sont disponibles, y compris des modèles tels que les protocoles de bus CANopen, Profibus ou DeviceNet. Les instructions d'installation sont fournies avec chaque carte, veuillez donc vous référer à la fiche de données pour les informations fournies sur la façon d'adapter la carte à l'appareil. Lorsque la carte est installée, la configuration est effectuée en sélectionnant l'icône appropriée dans le Panneau de navigation, et en cliquant sur le bouton Choisir côté de la propriété Option Carte.

Une boîte de dialogue similaire à celle illustrée ci-dessous sera affichée...



La sélection de la carte appropriée va ajouter une ou plusieurs icônes du volet de navigation, qui représentent le port supplémentaires ou les ports qui sont mis à disposition par la carte. Ces ports peuvent alors être configurés de la manière habituelle. L'exemple ci-dessous montre une carte d'extension en série...



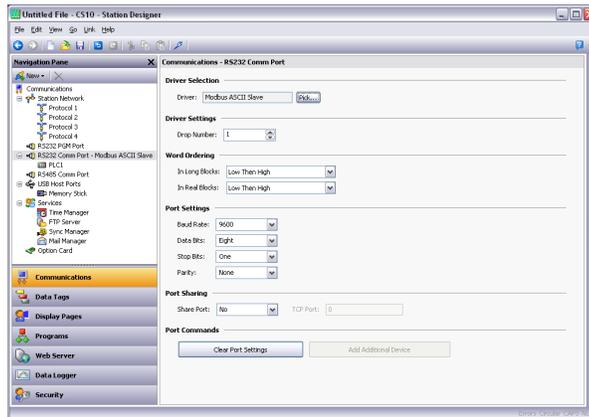
Les ports supplémentaires peuvent être configurés en suivant les instructions fournies dans les sections précédentes. Notez que les pilotes disponibles pour un port dépendront du type de connexion que ce dernier supporte. Par exemple, la carte d'extension CANOpen possède un port qui ne supportera que les pilotes conçus pour le standard de communication CAN.

## PROTOCOLES ESCLAVES

Pour les protocoles maîtres (ç.a.d. ceux pour lesquels le G3 initie la communication), il n'y a pas de configuration supplémentaire sous l'icône Communication. Pour les protocoles esclaves (ç.a.d. ceux où le périphérique Crimson répond à des demandes de périphériques extérieurs), cependant, la procédure est légèrement plus complexe vu que vous devez indiquer quelles données vous souhaitez exposer au périphérique extérieur.

### CHOIX DU PROTOCOLE

Comme pour les protocoles maîtres, la première étape est de sélectionner le protocole pour le port de communication que vous souhaitez utiliser. L'exemple ci-dessous montre le port RS232 du G3 configuré en Modbus ASCII esclave...

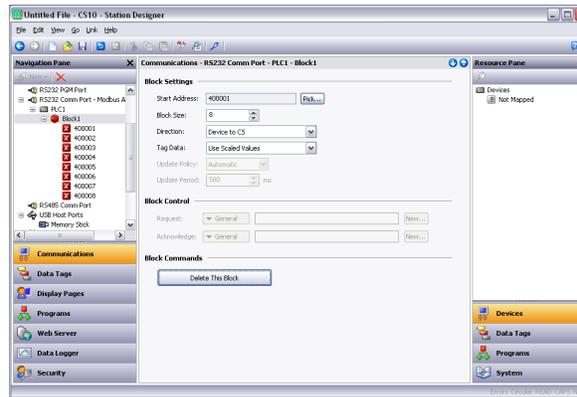


Notez qu'une unité a été créée automatiquement pour ce protocole. Dans le cas des protocoles maîtres, celle-ci représente l'unité à laquelle Crimson va accéder. Dans ce cas cependant, elle représente le périphérique lui-même. Il n'y aura donc qu'une seule unité et les paramètres tels que numéro de station qu'esclave seront présent généralement plutôt sur le port de communication que sur l'unité.

### AJOUT DE BLOCS DE PASSERELLE

Une fois le protocole configuré, vous devez maintenant décider quel type d'adresses vous souhaitez que le protocole esclave expose. Dans cet exemple, nous souhaitons utiliser les registres Modbus 40001 à 40008 pour permettre la lecture et l'écriture de certaines variables de notre base de données. Premièrement, sélectionnez l'icône du périphérique dans le panneau de navigation, puis cliquez sur le bouton Ajouter bloc de passerelle dans le panneau d'édition.

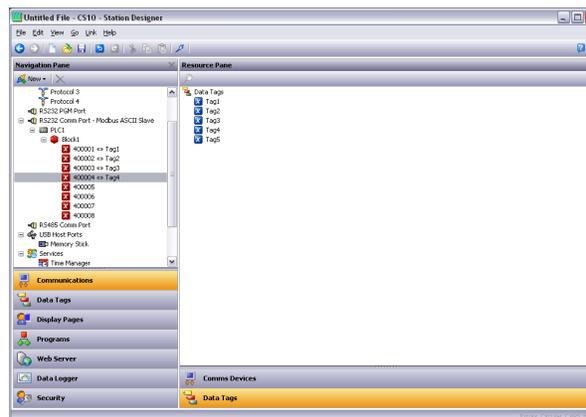
Une icône représentant le bloc apparaîtra sous le périphérique...



Dans l'exemple ci-dessus, « Adresse de départ » a été configurée sur 40001 pour indiquer le début du bloc et la taille de 8 indique le nombre de registres sur lesquels nous voulons mapper les données. Enfin, nous avons configuré la propriété « Direction » comme Périphérique à Crimson pour indiquer que nous souhaitons que le périphérique soit capable de lire et écrire les données exposées par ce bloc.

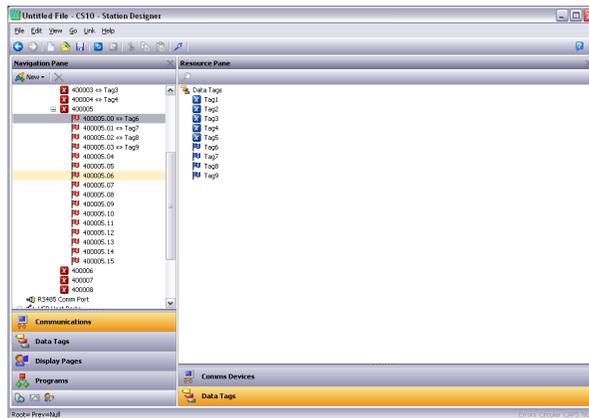
### AJOUT D'ÉLÉMENTS A UN BLOC

Une fois que le bloc a été créé et que sa taille a été définie, des entrées s'affichent dans le volet gauche de la fenêtre et représentent chacun des registres que le bloc contient. Lorsque vous sélectionnez l'un des registres, un panneau de ressource étendu apparaît et vous permet d'accéder à tous les éléments de données disponibles. Ces derniers comprennent à la fois des tags issus de votre base de données et des registres de données issus de tout périphérique de communication maître que vous aurez configuré...



Pour indiquer que vous voulez un registre particulier au sein de votre bloc passerelle afin de correspondre à un élément de données particulier, il suffit de faire glisser ce point grâce aux ressources du Panneau de navigation, en le déposant sur l'entrée appropriée du bloc passerelle. L'exemple ci-dessus montre comment les quatre premiers registres du le bloc ont été cartographiés à des balises appelées Tag1 par Tag4, indiquant que les accès 40001 à 40004 devraient être mis en correspondance avec les variables respectives.

## ACCES AUX BITS INDIVIDUELS

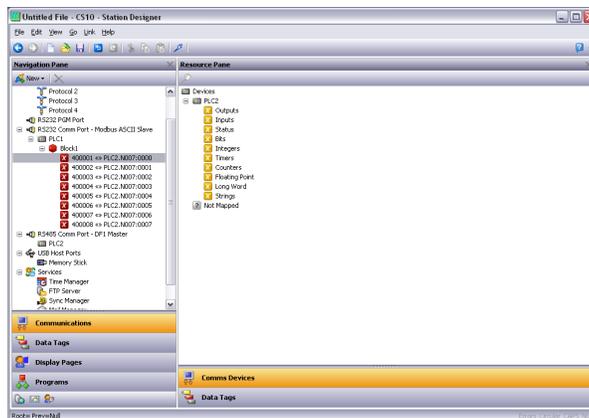


Si votre application le nécessite, vous pouvez développer des éléments individuels au sein du bloc passerelle de leurs bits constitutifs, et mapper un élément de données différent pour chaque bit. Pour ce faire, faites un clic droit sur l'élément en question, et sélectionnez Etendre Bits dans le menu contextuel. Le Panneau de navigation sera mis à jour pour les bits qui composent le registre, et ceux-ci pourront être mappés en utilisant le processus de glisser-déposer décrit ci-dessus.

## CONVERSION DE PROTOCOLE

Outre le mapping des données des cartes, les blocs de passerelle peuvent être utilisés pour exposer des données qui sont obtenues à partir d'autres périphériques distants ou pour déplacer des données entre ces deux périphériques. Cette fonctionnalité unique de conversion de protocole permet une intégration plus étroite entre les éléments de votre système de commande, même lors de l'utilisation de périphériques simples et peu onéreux.

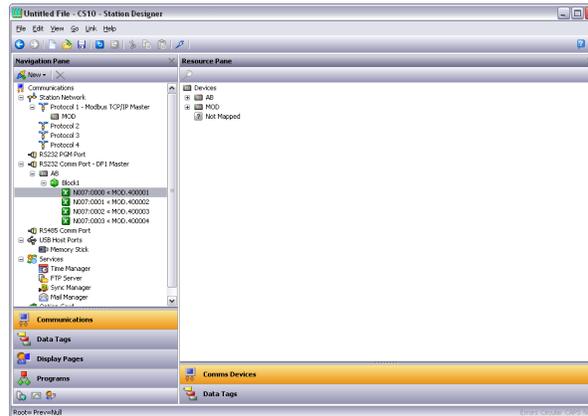
## MAITRE ET ESCLAVE



L'exposition de données à partir d'autres périphériques sur un protocole esclave est simplement une extension du processus de mapping décrit ci-dessus, sauf que cette fois, au lieu de glisser une étiquette à partir du volet droit, vous devez étendre le périphérique maître approprié et le glisser sur l'icône qui représente les registres que vous souhaitez exposer. Il vous est alors demandé d'entrer une adresse de départ dans le périphérique maître ainsi que le nombre de registres à mapper pour que les mappings soient créés comme dans l'illustration.

Dans cet exemple, les registres N7:0 à N7:7 d'un automate Allen-Bradley ont été exposés à l'accès via le protocole TCP / IP Modbus sur les registres 40001 à 40008. Crimson garantit automatiquement que ces données sont lues à partir de l'API Allen-Bradley afin de répondre aux requêtes Modbus et convertir automatiquement les écritures vers les registres Modbus en écritures vers l'API. Ce mécanisme permet même de connecter de simples API sur un réseau

## MAITRE ET MAITRE



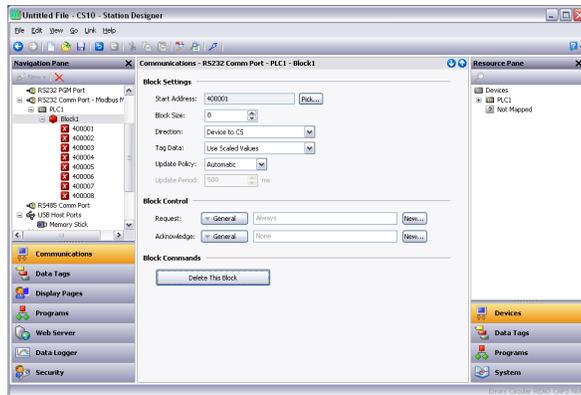
Pour déplacer les données entre deux périphériques maîtres, il vous suffit de sélectionner l'un d'entre eux et de créer un bloc de passerelle pour ce périphérique. Vous pouvez ajouter des mnémoniques aux autres registres du périphérique comme vous le feriez lors de l'exposition des données sur un protocole esclave. Encore une fois, C2 lit ou écrit automatiquement les données comme requis, en déplaçant de façon transparente les données entre les périphériques. L'exemple ci-dessus montre la façon de déplacer les données à partir d'un périphérique Modbus dans un périphérique SLC-500.

## OU CREER LE BLOC DE PASSERELLE ?

Vous pouvez vous demander si vous devez créer le bloc de passerelle dans le périphérique Allen-Bradley (comme dans cet exemple) ou dans le périphérique Modbus. La première chose à noter est que vous n'avez pas besoin de créer plusieurs blocs pour effectuer des transferts dans une seule direction. Si vous créez un bloc dans AB pour lire à partir de MOD et un bloc dans MOD pour écrire dans AB, vous effectuerez simplement deux fois le transfert et vous ralentirez tous les processus ! La deuxième observation est que la décision concernant la « possession » du bloc de passerelle par le périphérique est arbitraire. En règle générale, vous devez créer vos blocs afin de réduire le nombre de blocs dans la base de données. C'est-à-dire que si les registres du périphérique Allen-Bradley se trouvent dans une seule plage, mais si les registres du périphérique Modbus sont dispersés dans l'API, le bloc de passerelle doit être créé dans le périphérique Allen-Bradley afin de supprimer le besoin de créer plusieurs blocs pour accéder aux différentes plages du périphérique Modbus.

## CONTROLE DES BLOCS MAITRES

Les blocs passerelle au sein des appareils maîtres possèdent plusieurs propriétés supplémentaires...



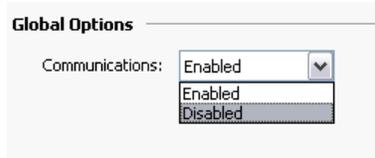
- Comme pour les blocs d'esclaves, la propriété *Tag de données* choisit comment les Tags de données sont mappés vers et à partir du bloc. Comme vous pourrez le découvrir dans le prochain chapitre, un ensemble de tags de données peut être soumis à divers stades de transformation. Cette propriété sélectionne où, dans le processus de transformation du bloc passerelle, les données seront obtenues et injectées.
- La propriété *Mise à jour de la politique* est utilisée pour définir la manière dont les mises à jour de bloc sont effectuées. La valeur par défaut de réglage automatique forcera les blocs lus à se mettre à jour en permanence, et les blocs écrits à ne transférer que les valeurs qui ont changé. Un réglage en continu de tous les blocs entraînera la mise à jour en continu. Un réglage de Temps écoulé mettra à jour tous les blocs au taux défini par la propriété *Mise à jour période*, la totalité d'une écriture bloc étant écrite à chaque fois.
- Les propriétés *Demander* et *Accepter* sont utilisées pour contrôler la synchronisation des mises à jour bloc par l'intermédiaire de tags ou d'autres éléments de données. Si la propriété *Accepter* est laissée vide, *Demander* permettra de lancer une activation avec une valeur nulle de désactivation de bloc et une valeur non nulle, ce qui lui permet de fonctionner. Si la valeur *Accepter* est définie, la demande fonctionnera comme un standard double.

## TRANSFORMATION DE DONNEES

Vous pouvez également utiliser les blocs de passerelle pour effectuer des opérations mathématiques que votre API n'est pas en mesure de gérer. Par exemple, vous pouvez lire un registre à partir de l'API, le mettre à l'échelle, en extraire la racine carrée et la réécrire sur un autre registre API. Pour cela, reportez-vous à la section « Etiquettes de données » et créez une variable mappée pour représenter la valeur d'entrée qui sera lue à partir du périphérique API. Créez ensuite une formule pour représenter la valeur de sortie en définissant l'expression mathématique (le calcul) de façon à effectuer le calcul requis. Vous pouvez enfin créer un bloc de passerelle à destination du registre de sortie, puis glisser la formule dessus pour demander à Crimson de réécrire cette valeur recalculée sur l'API.

## DESACTIVATION DES COMMUNICATIONS

Crimson donne la possibilité de désactiver toutes les communications basées pilote par le biais d'une propriété contenue dans l'élément de niveau supérieur de la catégorie Communication...



La désactivation des communications peut être utile au cours du développement quand vous n'avez pas les dispositifs à distance disponibles à votre emplacement actuel. En mode désactivé, Crimson définit initialement les tags correspondant à leurs valeurs simulées, puis leur permet d'être changés, comme si ces dernières étaient écrites selon des dispositifs associés. Si vos communications s'arrêtent sans raison, assurez-vous que vous n'avez pas ce paramètre réglé sur désactivé !



# TRAVAILLER AVEC LES TAGS

Une fois que vous avez configuré les options de communication pour votre base de données, l'étape suivante consiste à la définition des étiquettes que vous souhaitez afficher ou manipuler. Vous pouvez réaliser cette configuration en Sélectionnant la catégorie tags de données dans le panneau de navigation. Les tags peuvent être créés, supprimés et manipulés via les opérations standard mentionnées précédemment dans ce manuel.

## TOUT SUR LES TAGS

Les tags de données ou étiquettes sont des entités nominatives qui représentent des éléments de données.

### SOURCES DE DONNEES

Les tags peuvent récupérer les données depuis de nombreuses sources...

- Les étiquettes peuvent être *mappées* sur les variables dans les modules attachés ainsi que sur les registres des périphériques distants (API). De plus, Crimson lit automatiquement les valeurs correspondantes lorsque l'étiquette est référencée ou affichée. De la même façon, si vous modifiez une étiquette mappée, Crimson écrit automatiquement la nouvelle valeur sur le périphérique distant.
- Un tag peut être *interne*, auquel cas il représente un ou plusieurs éléments de données au sein de l'appareil basé sur Crimson. Les tags internes peuvent être marqués comme rétentifs, auquel cas ils garderont leurs valeurs à travers un cycle d'alimentation, ou la non rétentifs, auquel cas ils seront remis à zéro au moment du redémarrage.
- Un tag peut être une *expression*, dans ce cas, il représente un calcul basé sur les autres données, en utilisant éventuellement des opérateurs mathématiques et une ou plusieurs des fonctions internes ou définies par l'utilisateur Crimson. Les tags expressions sont utilisées pour calculer des valeurs dérivées pour le traitement interne ou le transfert vers des périphériques distants.

### TYPES DE TAGS

Crimson supporte trois types de tags...

- Les *Tags numériques* représentent des valeurs de nombre entière ou de point flottant.
- Les *Tags drapeau* représentent une valeur on ou off (active ou désactivée).
- Les *Tags de ligne* représentent des lignes de caractères unicode.

Chacun des trois principaux types de tags peuvent représenter une valeur unique ou un tableau de valeurs. Un tableau est un ensemble d'éléments possédant des propriétés similaires regroupés et accessibles par une valeur d'index. Les tableaux mappés correspondent aux registres multiples dans le dispositif cible.

Un quatrième type de tag est le *Tag de base*. Ceci est une version simplifiée d'un tag, qui peut seulement représenter des chaînes ou des expressions numériques. Il lui manque de nombreuses fonctionnalités puissantes des balises standard. Il est généralement utilisé pour coder les éléments de données simples, comme les constantes.

## ATTRIBUTS DES TAGS

Les tags sont dans Crimson des objets riches qui définissent diverses propriétés communes...

- Une *étiquette de tag* est transférable via une chaîne de caractères lisible, utilisé pour étiqueter automatiquement les champs de données se référant à cet article de données. Elle est également utilisée par le serveur web et le Journal de données de l'étiquette associée aux éléments de données.
- La *description d'un tag* est une chaîne non traduisible utilisée pour fournir une annotation à la fin du tag. Elle n'est pas normalement visualisable par l'utilisateur de l'appareil cible, mais peut être exposée à des fins de diagnostic.
- Le *format* d'un tag est un ensemble de paramètres qui définissent la méthode par laquelle les données du tag doit être présentées pour l'affichage. Le format peut être laissé sous forme générale, dans ce cas Crimson utilisera les règles de formatage par défaut, ou peut être mis en correspondance à l'un des nombreux types de formatage. Par exemple, une valeur numérique peut être affichée en format scientifique ou peut être utilisée pour sélectionner un certain nombre de chaînes de texte différentes.
- Le *coloriage* d'un tag est un ensemble de paramètres qui définissent comment le texte de la balise doit être affiché ou quelles sont les couleurs à utiliser pour représenter l'état de la balise. Encore une fois, un certain nombre de colorations différentes existent, permettant d'apposer une étiquette pour modifier l'apparence en fonction de plusieurs conditions. Avant-plan et couleurs de fond sont définis par paires, et peuvent être consultés individuellement par les primitives d'affichage.
- Le *descripteur de sécurité* d'un tag définit les règles d'accès à utiliser lors du changement de la balise, et si oui ou non ces modifications doivent être enregistrées.

Les tags de base ne proposent pas les options de format, de coloriage et de descripteur de sécurité. En outre, les tags numérique et drapeaux définissent des alertes et des déclencheurs.

## AVANTAGES DES TAGS

Crimson vous permet de placer directement un module ou un registre d'API sur une page d'affichage de façon à ce que vous puissiez configurer une base de données simple sans ne jamais avoir à définir de tags de données. Il peut être intéressant de prendre quelques moments afin de comprendre les bénéfices que le léger apport de travail que représente la manipulation des tags peut apporter...

- Les étiquettes vous permettent de nommer des éléments de données (Mnémonique). Ainsi, vous pouvez savoir à quels éléments de données de l'API vous vous référez. D'autre part, si les données de l'API changent ou si vous décidez de basculer dans une famille d'API complètement différente, vous pouvez simplement mapper à nouveau les étiquettes et éviter de devoir apporter d'autres modifications à votre base de données.
- Les étiquettes vous permettent d'éviter de devoir rentrer les mêmes informations plusieurs fois. Lorsque vous créez une étiquette, vous spécifiez la façon dont l'étiquette s'affiche. Dans le cas d'une étiquette Entier, cela signifie que vous demandez à Crimson combien de décimales seront utilisées et les unités, le cas

échéant, qui seront ajoutées à la valeur. Lorsque vous placez une étiquette sur une page d'affichage, Crimson sait comment la formater sans que vous ne deviez rien faire d'autre. De la même façon, si vous décidez de modifier la mise en forme et, peut-être, de basculer d'un jeu d'unités vers un autre, vous pouvez le faire en un endroit sans devoir modifier les pages d'affichage les unes après les autres.

- Les étiquettes sont utilisées comme base de l'animation couleur. Les différentes couleurs d'une étiquette peuvent être utilisées pour spécifier les autres primitives d'animation qui sont affichées.
- Les étiquettes représentent la clé de la mise en oeuvre des protocoles esclaves. Crimson considère ces protocoles comme des mécanismes d'exposition des éléments de données dans le maître (CSMSTRSX ou GT ou Data Station Plus ou G3). Cela permet d'accéder aux mêmes données via plusieurs ports de façon que, par exemple, une configuration de machine pourrait être modifiée par un package SCADA local et un package semblable fonctionnant sur Ethernet à partir d'un site distant. Sans les étiquettes, rien ne serait exposé et ce mécanisme ne pourrait pas être mise en oeuvre !
- Les étiquettes sont utilisées chez Crimson pour implémenter de nombreuses fonctionnalités avancées. Si vous souhaitez utiliser des fonctionnalités comme les alarmes, les Triggers, l'enregistrement de données ou le serveur Web, vous devrez utiliser des étiquettes. Les données de mise en forme à partir de la définition de l'étiquette sont généralement requises par toutes ces fonctionnalités, ainsi les étiquettes sont obligatoires pour garantir le fonctionnement.

Pour résumer, les étiquettes automatisent de nombreuses tâches lors de la programmation, vous permettant de gagner du temps. Même si vous décidez de ne pas utiliser les étiquettes, la majorité des chapitres ultérieurs de ce manuel se rapportent aux concepts abordés dans ce chapitre. Vous devez donc le lire soigneusement avant de continuer.

## **EDITION DES PROPRIETES**

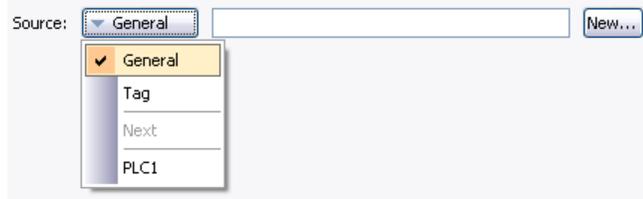
La plupart des propriétés sont éditées selon des modalités qui sont évidentes pour quiconque a utilisé un système d'exploitation Windows. Par exemple, il peut être nécessaire d'entrer une valeur numérique, ou de sélectionner un élément dans une liste déroulante. Certains types de propriétés, cependant, offrent des options de montage plus complexes, décrites ci-dessous.

### **PROPRIETES D'EXPRESSION**

Les expressions des propriétés sont susceptibles d'être appliqués à...

- Une valeur constante.
- Le contenu d'un bloc de données.
- Le contenu d'un registre dans un appareil de communication à distance.
- Une combinaison de ces éléments reliés entre eux en utilisant des opérateurs mathématiques différents.
- La valeur de retour d'un programme local.

Dans son état par défaut, le bouton fleche situé immédiatement après l'étiquette de la propriété montre que le champ est en mode Général. La zone d'édition à droite du bouton montre une chaîne grisée indiquant le comportement par défaut de la propriété. Un exemple d'une expression vide de propriété sans valeur par défaut est présenté ci-dessous...



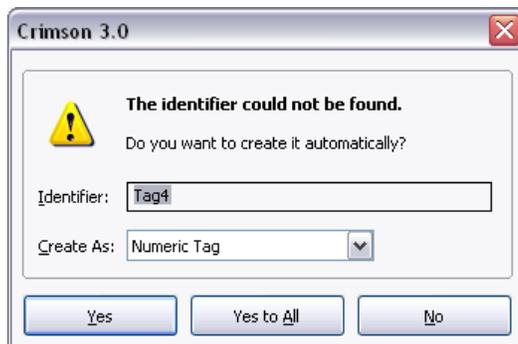
Si vous êtes familier avec la syntaxe des expressions Crimson-une description complète de cette dernière est présente dans le chapitre des expressions d'écriture- vous pouvez modifier la propriété en saisissant une expression directement dans la boîte d'édition.

#### SELECTIONNER UN TAG

Pour définir une propriété d'expression sur une balise existante, vous avez quatre options. Premièrement, vous pouvez faire en sorte que le champ cible soit sélectionné, puis double-cliquer sur la balise requise dans le Panneau de ressources. Deuxièmement, vous pouvez faire glisser le tag depuis le Panneau de Ressources et le déposer sur le champ cible. Troisièmement, vous pouvez sélectionner la balise dans le menu déroulant, l'activer via par le bouton fléché et ne pas oublier que vous pourriez avoir déplacé la cible vers le champ dans la première place ! Enfin, vous pouvez simplement le faire de manière classique ancienne et saisir le nom de la balise dans la propriété d'expression.

#### CREER UN TAG

Pour appliquer une propriété d'expression à un nouveau tag, vous avez également quatre options. Premièrement, pour les expressions qui définissent la source d'un élément de données, vous pouvez sélectionner l'option Nouveau Tag dans le menu déroulant, activé par le bouton fléché. Deuxièmement, pour la plupart des domaines d'expression vous pouvez appuyer sur le bouton Nouveau qui est affiché à côté de la zone d'édition en mode général. Troisièmement, si vous avez déjà sélectionné une balise, vous pouvez appuyer sur le bouton Choisir et sélectionnez Nouveau tag dans la boîte de dialogue qui s'affiche. Enfin, vous pouvez saisir le nom de la nouvelle balise dans le cadre d'une expression, que Crimson vous invite à entrer via une boîte de dialogue similaire à celle présentée ci-dessous...



Dans cet exemple, un référencement d'expression Tag4 expression a été inscrit, mais aucune balise n'est présente dans la base. Crimson repère l'erreur et vous demande si vous souhaitez créer cette balise automatiquement. La liste déroulante peut être utilisée pour sélectionner le type de la nouvelle balise, et contiendra les options appropriées pour le contexte dans lequel cette dernière a été utilisée. Le bouton Oui pour tous bouton peut être utilisé pour demander à

Crimson d'utiliser les données par défaut et de créer n'importe quel type d'autres balises manquantes contenues dans cette expression sans aucune autre sollicitation.

#### REFERENCES COMM.

Pour sélectionner un registre à partir d'un périphérique de communication, sélectionnez un appareil dans la liste déroulante. Une boîte de dialogue s'affichera, vous permettant de choisir un registre au sein de ce périphérique de communication à distance. Les différents dispositifs de communication sont indiqués à la fin du menu, dans l'ordre dans lequel ils ont été créés. Vous pouvez également sélectionner l'option suivante du menu déroulant, plaçant ainsi la balise actuelle au même niveau que le dernier registre PLC utilisé. Par exemple, mapper un tag registre 32-bit Modbus sur 40001, puis sélectionner Suivant mapera le tag suivant à 40003.

#### EDITER UNE EXPRESSION

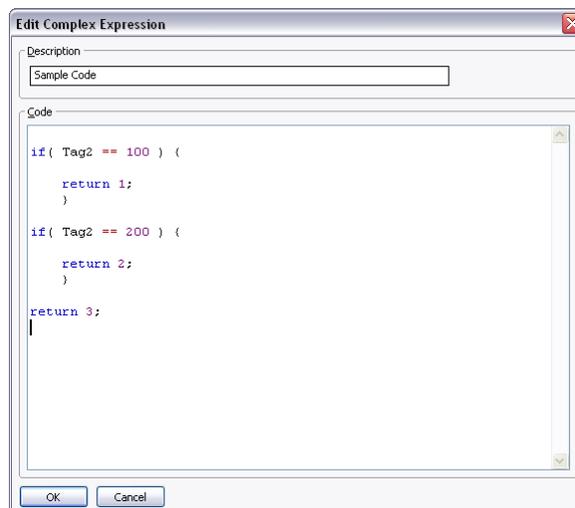
Comme mentionné ci-dessus, les expressions générales sont généralement éditées directement dans la boîte d'édition de la propriété. Toutefois, elles peuvent également être éditées via le bouton Modifier, activant ainsi une boîte de dialogue dédiée qui permet de visualiser plus d'options relatives à l'expression...



L'éditeur utilisé dans cette boîte de dialogue est le même que celui utilisé pour créer des programmes globaux. Il prévoit donc une coloration de syntaxe. Vous pouvez également accéder à l'aide sur les fonctions du système en plaçant votre curseur dans ou à la fin du nom de fonction et en appuyant sur F1.

#### EXPRESSIONS COMPLEXES

Si votre expression est trop complexe pour tenir dans une seule ligne, vous pouvez sélectionner l'option Complexe du menu déroulant et créer un programme local...



L'instruction `return` est utilisée afin de fournir la valeur de l'expression, comme si vous aviez contacté un programme global. Notez encore une fois que l'éditeur de programme est utilisé, fournit une coloration syntaxique ainsi qu'une possibilité d'auto-installation et que le mécanisme F1 décrit ci-dessus peut être utilisé pour consulter des informations d'aide sur les

fonctions du système. Le texte de description vous permet de faire une note rapide de sur le fonctionnement du programme. Il sera affiché à côté de la propriété à titre de référence. Pour plus d'informations sur l'écriture et l'édition des programmes, se référer aux chapitres suivants de ce manuel.

## GESTION DES LANGUES

Les bases de données Crimson sont conçues pour prendre en charge un fonctionnement multilingue grâce à laquelle toute chaîne qui est présentée à l'utilisateur de IHM peut être affichée dans l'une des différentes langues. Pour vous permettre de définir ces traductions, les propriétés qui contiennent ces chaînes disposent d'un bouton nommé Traduction, situé à leur droite.

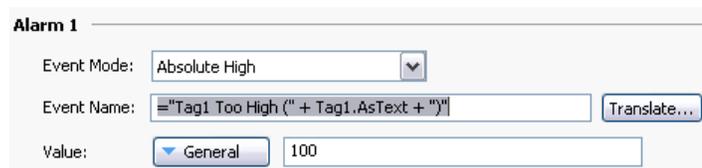


Pour entrer les traductions, cliquez sur le bouton et la boîte de dialogue suivante s'affiche...



Les langues répertoriées dans la boîte de dialogue sont définies au niveau base de données. Se référer au chapitre sur la localisation des informations pour plus d'informations sur la façon dont elles sont sélectionnées, sur le fonctionnement du programme d'auto-translation et sur la façon de changer l'exécution d'une langue. Notez que si vous n'entrez pas de texte pour une langue particulière, et que cette langue est ensuite sélectionné par l'opérateur, Crimson utilisera le texte de la langue par défaut à la place.

Les chaînes à traduire peuvent également être définies comme étant égales aux expressions. Par exemple, alors qu'un nom d'alarme devrait généralement être réglé lors de la configuration, un concepteur de bases de données pourrait décider que l'alarme contient la valeur de la balise qui a déclenché l'alarme. Les expressions peuvent être inscrites en les préfixant avec un signe égal, comme on le ferait avec une feuille de calcul d'édition, comme le montre l'exemple ci-dessous



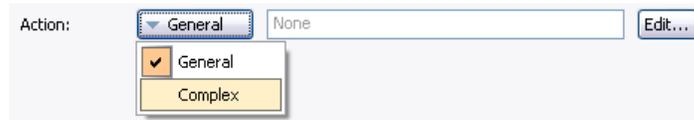
Notez l'utilisation de la propriété `AsText` de la balise permettant à sa valeur d'être consultée sous forme de chaîne en fonction de son réglage de format. Se référer au chapitre des consacré à l'écriture des expressions pour plus de détails.

### PROPRIETES DOUBLE SENS

Les propriétés telles que les chaînes à traduire qui sont susceptibles d'être calées sur une valeur constante ou une expression à double sens. En plus d'accepter les expressions préfixées par un signe égal, elles peuvent être mises en correspondances aux valeurs des tags en faisant simplement glisser la balise appropriée dans le Panneau de Ressources et en le déposant dans le champ.

### PROPRIETES D'ACTION

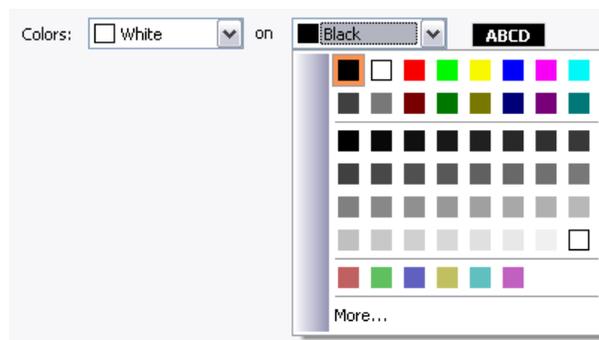
Les propriétés d'action sont utilisées dans les balises pour définir les actions à effectuer en cas de déclenchement des conditions remplies, ou lorsque la valeur du tag est modifiée. Elles sont éditées via un menu déroulant et une boîte de dialogue similaire à celle utilisée pour l'édition des expressions est affichée...



Comme pour les expressions, le bouton Edition peut être utilisé pour ouvrir une plus grande fenêtre d'édition, et les actions complexes peuvent être créés par le biais de programmes locaux.

### PROPRIETES COULEUR

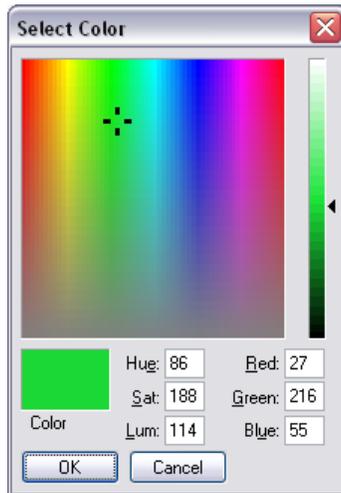
Les propriétés Couleur de l'étiquette représentent une couleur de premier plan et d'arrière-plan, qui est utilisée pour afficher l'état de l'étiquette en forme textuelle. Ces couleurs peuvent être utilisées pour définir la couleur des autres primitives d'animation. L'exemple ci-dessous présente un couple de couleurs en cours de modification...



La liste déroulante contient les couleurs suivantes...

- Les seize couleurs VGA standard.
- Trente-deux tons de gris qui varient entre le noir et blanc.
- Toutes les autres couleurs utilisées dans la base de données, jusqu'à la limite de vingt-quatre.

L'option Plus, située en bas de la liste, peut être utilisée pour appeler la boîte de dialogue de définition des couleurs...



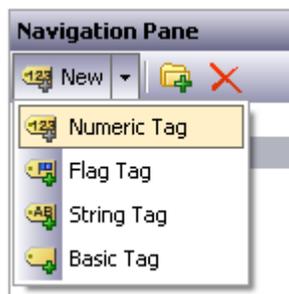
Grâce à cette boîte de dialogue, vous pouvez définir une couleur de plusieurs façons. Vous pouvez les choisir dans la palette, dans la fenêtre de « l'arc en ciel » ou entrer les paramètres TSL ou RVB explicites. Si la couleur sélectionnée n'a pas été précédemment utilisée dans la base de données et n'est pas une teinte de gris standard, elle sera ajoutée aux couleurs personnalisées affichées dans le menu déroulant.

## PROPRIETES DE JOURNALISATION

Lorsque vous entrez les balises de catégorie de données dans le Panneau de navigation, vous remarquerez un certain nombre de propriétés relatives à l'enregistrement des événements. Ces propriétés contrôlent si et comment les événements générés par des balises ou par leurs alarmes seront enregistrés sur carte CompactFlash. Elles sont analogues aux propriétés définies par les journaux de données, et vous pouvez donc consulter le chapitre correspondant de ce manuel pour plus d'informations sur la façon dont elles peuvent être utilisées.

## CREATION DE TAGS

Les tags de données sont créés et autrement manipulés via les méthodes habituelles, dans le Panneau de navigation. Vous remarquerez que vous pouvez créer des dossiers pour organiser vos tags, et que le bouton Nouveau de la barre d'outils contient une flèche déroulante pour vous permettre de sélectionner le type de tag à insérer. Le côté gauche du bouton Nouveau crée un tag du même type que le dernier que vous avez créé, ce qui rend plus facile de créer plusieurs tags sans utiliser la liste déroulante à chaque fois.



## DUPLICATION DE TAGS

La commande de duplication intelligente du menu Edition peut être utilisée pour créer une nouvelle copie d'un tag existant, augmentant sa source de données de référence à l'élément de données suivant.

La définition de la « prochaine » balise dépend du type exact de l'élément de données. A titre d'exemple, l'utilisation de cette fonctionnalité sur un tag 16-bit mappé Modbus sur registre 40001 produira un tag mappé à 40002, tandis qu'un tag mappé à `Array [2]`, produira un tag mappé à `Array [3]`.

Avec cette installation, il est beaucoup plus facile de créer des ensembles de tags faisant référence à des éléments de données séquentielles.

## EDITION DE TAGS MULTIPLES

Vous pouvez modifier les propriétés de plusieurs balises en même temps. Crimson supporte cette opération en vous faisant modifier une balise, puis vous permettant de définir les propriétés des autres balises égales à celles de celle que vous avez d'abord éditée. Crimson propose deux méthodes pour ce faire.

### UTILISATION DE LA COMMANDE COPIER DEPUIS

La commande Copier depuis peut être utilisée pour copier les propriétés sélectionnées d'un tag donné sur un ou plusieurs tags de la Liste de navigation. Pour utiliser cette commande, sélectionnez les balises cibles, puis effectuez un clic-droit pour accéder au menu contextuel. (Notez que la liste de navigation pour les balises supporte la sélection multiple via les touches **MAJ** et **CTRL**) Sélectionnez un tag et le curseur changera pour vous permettre de sélectionner la balise à partir de laquelle l'opération de copie doit être effectuée. En fonction de la commande qui a été choisie, une ou plusieurs propriétés de la balise source seront ensuite appliqués aux balises cible.

### UTILISATION DE LA COMMANDE COLLAGE SPECIAL

La commande Collage spécial peut être utilisée pour obtenir le même résultat, mais via une autre méthode qui permet aussi aux propriétés d'être copiées entre les bases de données et / ou entre plusieurs instances de Crimson. Tout d'abord, sélectionnez la balise source et utilisez la commande Copier pour la mettre dans le Presse-papiers. Ensuite, sélectionnez les balises cible dans la liste de navigation, (ici aussi, des sélections multiples peuvent être faites). Enfin, effectuez un clic droit sur la sélection pour accéder au menu contextuel, puis sélectionnez la commande Collage spécial. La boîte de dialogue suivante s'affiche...



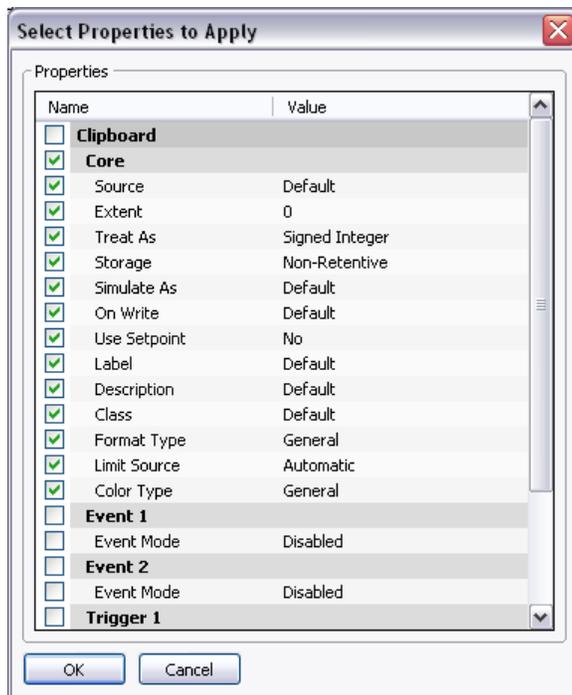
Propriétés sélectionnées à partir de la balise source seront appliquées aux balises cible.

## SELECTIONS DE PROPRIETE

Les deux méthodes décrites ci-dessus vous permettent de définir quelles sont les propriétés à copier...

- Copies de *Mapping* de la propriété Source. Cette option copie également toutes les propriétés qui contrôlent options de communications d'un tag telles que l'étendue, l'accès, et tous les autres éléments qui sont contenus dans la section Source de données.
- *Mise à l'échelle* copie la propriété de mise à l'échelle.
- *Format* copie le type de format.
- *Coloriage* copie le type de coloration, et l'objet associé à colorier.
- *Alarmes* copie toutes les propriétés de l'alarme 1 et l'alarme 2.
- *Activateurs* copie toutes les propriétés des activateurs 1 et 2
- *Sécurité* copie toutes les propriétés de la page Sécurité du tag.

En outre, l'option *Sélectionner* peut être utilisée pour sélectionner les propriétés à copier...



Cette liste contient une présentation hiérarchique de toutes les propriétés définies par la balise source, de leur organisation en fonction de la configuration utilisée avec la balise d'édition, ainsi que la valeur attribuée à chacune. Les propriétés ou les groupes de propriétés peuvent être sélectionnés ou désélectionnés en cochant les cases associées. Seules les propriétés marqués seront modifiées, vous permettant un contrôle de bas niveau de ce qui est copié d'un tag à un autre.

## IMPORTER ET EXPORTER

La sélection des balises de données dans la Liste de navigation permet d'accéder aux boutons qui peuvent être utilisés pour exporter et importer les Tags de données dans votre base de données. Les tags peuvent être exportés vers l'un des fichiers texte Unicode ou ANSI ou édités via des applications telles que Microsoft Excel. Le fichier d'exportation est divisé en sections basées sur le type de marque, le type de format et le type de coloration. Chaque section contient un nombre de colonnes, dont la signification peut être déterminée en consultant les rubriques ci-dessous.

## TROUVER L'UTILISATION D'UN TAG

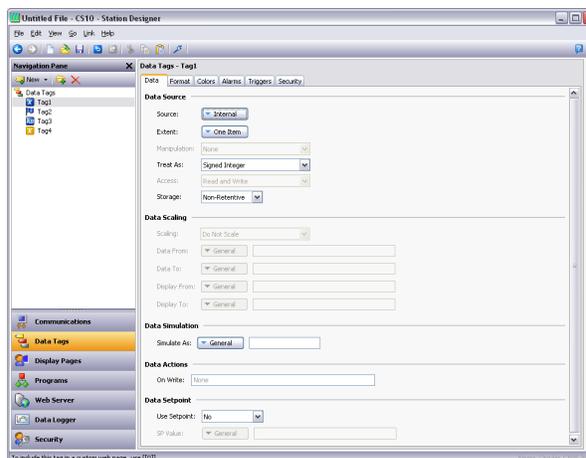
Vous pouvez trouver tous les éléments qui font référence à une balise grâce à un clic droit sur ce point dans le Panneau de navigation puis en sélectionnant la commande Rechercher. Les éléments résultant seront placés sur la liste Résultats de la recherche, et peuvent être consultés par le biais des commandes **F4** et **MAJ + F4** touches. La liste elle-même peut être affichée ou masquée en appuyant sur **F8**.

## TAGS NUMERIQUES

Un tag numérique représente un ou plusieurs nombres entiers ou des valeurs en virgule flottante. Crimson effectue tous les calculs internes à l'aide d'entiers 32 bits, afin que toutes les données soient converties en un de ces formulaires avant le traitement. Les tags numériques mappés peuvent supporter un certain nombre de transformations établies entre les données brutes et les données qui seront utilisées par Crimson. La procédure exacte est définie en détail plus loin dans ce chapitre.

## PROPRIETES DE DONNEES

Un tag numérique a les propriétés suivantes sur son onglet Données...



## SOURCES DES DONNEES

- La propriété *Source* définit l'emplacement où le tag reçoit ses données. Le résultat du réglage est inscrit dans une balise interne, mais la liste déroulante peut être utilisée pour sélectionner une expression générale, un autre tag de données ou un élément situé sur un périphérique distant.

- La propriété *Etendue* permet de choisir entre une balise unique ou un tableau. Si vous sélectionnez un tableau, vous devez entrer le nombre requis d'éléments. Les tableaux ne sont pas autorisés pour les balises dont la source est une expression. Pour les éléments mappés, le nombre de registres pouvant être lus dépend du type de données définis pour l'adresse. Par exemple, un tableau de deux éléments qui ont été mappés sur un registre de type Word se traduira dans quatre registres, deux mots étant nécessaire pour chaque valeur. Un tableau similaire mappé à un type de données Word n'a besoin que de deux registres.
- La propriété *Manipulation* définit la transformation d'une première étape qui est appliqué en tant que données de communication puis transférée dans une balise mappée. Les options suivantes sont disponibles en fonction du type de données exacte étant utilisées...

MANIPULATION	DESCRIPTION
Aucune	Les données ne seront pas modifiées.
Inverser Bits	Chaque bit de données aura son état inversé.
Rétablir bits	Le bit le plus significatif dans les données sera échangé avec le bit le moins significatif, les morceaux intermédiaires n'étant pas traitées de la même façon.
Rétablir Octets	L'octet le plus significatif dans les données sera échangé avec l'octet le moins significatif et ainsi de suite. Uniquement disponible pour les éléments de données de 16 bits ou plus en taille.
Rétablir mots	Le mot le plus important dans les données sera échangé avec le mot le moins significatif. Uniquement disponible pour les données d'exactly 32 bits de taille.

- La propriété *Traiter comme* pour les tags internes définit le type de données du tag. Pour les tags mappés, elle définit la façon dont les données manipulées seront interprétées par Crimson. La propriété sera par défaut lorsque le tag est associé, mais peut être changé. Les options suivantes peuvent être disponibles en fonction du type de données exacte des données de communication...

TRAITER COMME	DESCRIPTION
Entier signé	Les données seront traitées en tant qu'entier 32-bit signé, avec des données de valeur plus petites étendues. Par exemple, une valeur 16 bits de 0x8000 sera convertie en 0xFFFF8000.
Entier non signé	Les données seront traitées en tant qu'entier 32-bit signé, avec des éléments de données étendus à zéro. Par exemple, une valeur de 32 bits de 0x8000 sera convertie en 0x00008000. Uniquement disponible pour les éléments de données de moins de 32-bits de taille.
Point flottant	Les données seront traitées en tant que valeur 32-bit à virgule flottante. Uniquement disponible pour les éléments de données de 32 bits de taille.

- La propriété *Accès* est utilisée pour les balises cartographiées, définissant le type d'opérations de communications autorisées. Les tags internes peuvent toujours être lus et écrits, et les tags d'expression sont toujours en lecture seule.
- La propriété *Stockage* est utilisée pour indiquer si la balise sera conservée via le cycle de l'appareil cible. Cette option est généralement utilisée pour les balises internes, mais les tags mappés en écriture seule peuvent aussi voir leurs valeurs retenues.

#### MISE A L'ÉCHELLE DES DONNEES

- La propriété de *Mise à l'échelle* utilisée pour les tags mappé, permettant de définir une étape de mise à l'échelle finale qui sera appliquée aux données. les données peuvent soit être élargies à l'entier ou à la virgule flottante, quelle que soit la façon dont Crimson traite les informations de communication. Par exemple, une valeur entière peut être réduite à une valeur en virgule flottante, auquel cas Crimson examinera la balise en virgule flottante. De même, une valeur en virgule flottante pourrait être reconvertie en un nombre entier, sans avoir à changer son ampleur.
- La propriété *Origine des données* de et *Destination des données* définit l'ampleur la transformation qui se produit sur la plage de lecture et de la transformation qui se produit sur une écriture. Les valeurs doivent correspondre au type de données visées, par exemple seules les données traitées comme en virgule flottante peuvent avoir des valeurs non intégrales inscrites dans ces domaines.
- Les propriétés *Afficher depuis* et *Afficher vers* définissent la portée de la transformation en lecture et écriture. Les valeurs doivent correspondre au type de données spécifiées à l'échelle.

#### SIMULATION DE DONNEES

- La propriété *Simuler sous* définit la valeur devant être utilisé pour le tag lorsque vous travaillez dans l'éditeur de pages. La saisie d'une valeur sensible permet une meilleure représentation de l'apparence de la page. Cette valeur est également utilisés comme valeur par défaut de la balise par le dispositif cible, si la communication est globalement désactivée.

#### ACTIONS DONNEES

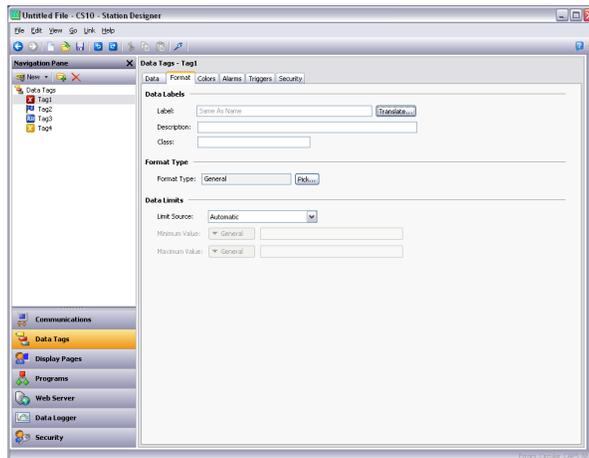
- La propriété *Ecrire Sur* définit une action qui peut être invoquée lorsque la balise est modifiée. La variable système *Données* contiendra la valeur des nouvelles données lors de l'écriture et quand l'action est exécutée. Le recours aux propriétés d'écriture est abordé plus loin dans ce chapitre.

#### POINTS PREDEFINIS

- La propriété *Utilisation de points* prédéfinis est utilisée pour activer ou désactiver un point de consigne pour cette balise.
- La propriété *Valeur SP* définit une expression une autre balise autre que celles que le tag est théoriquement censé suivre. Cette consigne peut ensuite être utilisée dans des alarmes ou des primitives afin de mettre en œuvre diverses fonctions.

## PROPRIETES DE FORMAT

Un identificateur numérique présente les propriétés suivantes sur son onglet Format...



### ETIQUETTES DE DONNEES

- La propriété *Etiquette* a été discutée ci-dessus, sous Attributs du tag.
- La propriété *Description* a été discutée ci-dessus, sous Attributs du tag.
- La propriété *Classe* est réservée à une expansion future.

### TYPE DE FORMAT

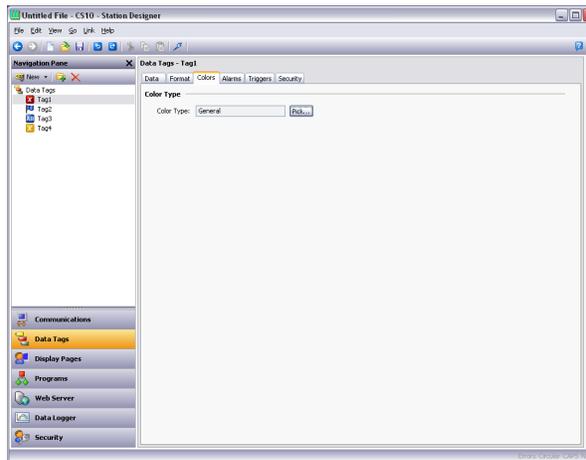
- La propriété *Type de format* permet de sélectionner le format de ce tag. Les différents types de formats sont discutés en détail dans un chapitre suivant, comme le sont les autres propriétés qui peuvent apparaître en fonction du type de format que vous avez sélectionné.

### LIMITES DE DONNEES

- La propriété *Source Limite* définit comment les limites des données de la balise d'entrée sont définies. Le paramètre par défaut est spécifié sous l'onglet des données utilisées comme source primaire, ainsi que le format objet utilisé comme une solution de repli. Si la source ne peut définir une plage, la plage maximale prise en charge pour le type de données de la balise est utilisée. Un réglage à partir du format peut être utilisé pour forcer le format d'objet à utiliser, tandis qu'un réglage défini par l'utilisateur peut être utilisé pour permettre la saisie manuelle des limites.
- Les *Valeurs minimum* et *Maximum* des propriétés de valeur sont utilisées pour définir manuellement les limites de la saisie de données lorsque la source limite est fixée auprès de l'utilisateur défini.

## PROPRIETES COULEUR

Un tag numérique possède les propriétés suivantes sur l'onglet Couleurs...

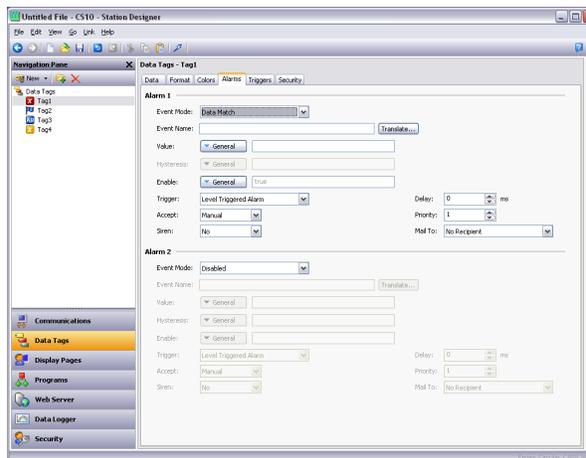


### TYPE DE COULEUR

- La propriété *Type de couleur* définit la couleur pour cette balise. Les différents types de coloration sont discutés en détail dans le chapitre suivant, comme le sont les autres propriétés qui peuvent apparaître en fonction de l'option que vous avez sélectionnée.

## PROPRIETES D'ALARME

Un tag numérique possède les propriétés suivantes sur son onglet Alarmes...



### POUR CHAQUE ALARME

- La propriété *Mode événement* est utilisée pour indiquer la logique qui sera utilisée pour décider si l'alarme doit être activée. Les tableaux ci-dessous présentent les modes disponibles.

MODE	L'ALARME SE DÉCLENCHERA QUAND...
Données correspondent	La valeur de la balise est égale à la <i>valeur</i> de l'alarme.

MODE	L'ALARME SE DÉCLANCHERA QUAND...
Données ne correspondent pas	La valeur de la balise n'est pas égale à la <i>valeur</i> de l'alarme.
Absolu élevé	La valeur de la balise dépasse la <i>valeur</i> de l'alarme.
Absolu faible	La valeur de la balise est inférieure à la <i>valeur</i> de l'alarme.
Hausse de la valeur	La valeur de la balise augmente selon la <i>valeur</i> de l'alarme.
Baisse de la valeur	La valeur de la balise baisse selon la <i>valeur</i> de l'alarme.
Changement de la valeur	La valeur de la modification des balises change selon la <i>valeur</i> de l'alarme.

Les modes suivants sont disponibles lorsqu'une consigne est définie...

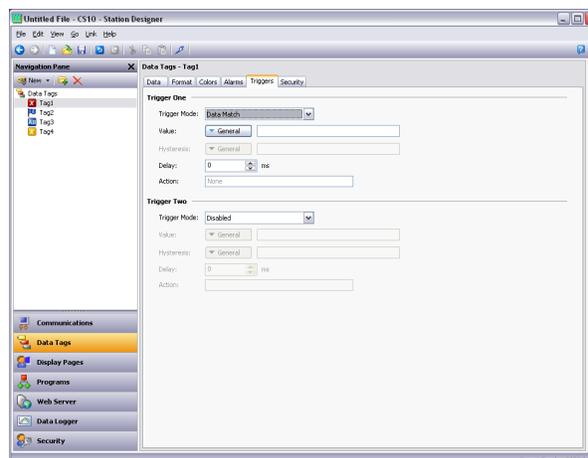
MODE	L'ALARME SE DÉCLANCHERA QUAND...
Ecart élevé	La valeur de la balise est supérieure au <i>point prédéfini</i> de la balise, d'un montant égal ou supérieur à la <i>valeur</i> de l'alarme.
Faible écart	La valeur de la balise est inférieure au <i>point prédéfini</i> de la balise, d'un montant égal ou supérieur à la <i>valeur</i> de l'alarme.
Hors bande	Le tag se déplace en dehors de la bande, qui est égale à deux fois la largeur de la <i>valeur</i> de l'alarme et est centrée sur le <i>point prédéfini</i> du tag
En bande	Le tag se déplace à l'intérieur de la bande, qui est égale à deux fois la largeur de la <i>valeur</i> de l'alarme et est centrée sur le <i>point prédéfini</i> du tag

- La propriété *Nom de l'événement* définit le nom qui sera affiché dans la visionneuse d'alarme ou dans le journal des événements en se référant à cet événement.
- La propriété *Valeur* définit soit la valeur absolue sur laquelle l'alarme sera activée, l'écart de la valeur de consigne ou la variation de valeur qui doivent se produire depuis la dernière alarme déclenchée. L'interprétation exacte dépend du mode d'événement tel que décrit ci-dessus.
- La propriété *Hystérésis* est utilisé pour prévenir une oscillation d'alarme quand le processus est à proximité de la condition d'alarme. Par exemple, pour une alarme haute absolue, l'alarme devient active lorsque la balise dépasse la valeur de l'alarme, mais se désactive lorsque la balise est à nouveau inférieure à la valeur d'un montant supérieur ou égal à l'hystérésis de l'alarme. Rappelez-vous que cette propriété agit toujours de manière à maintenir une alarme, une fois cette dernière activée, et afin de ne pas modifier le point où l'activation se produit.
- La propriété *Activer* définit une expression qui active ou désactive l'alarme. Un valeur autre que zéro ou vide active l'alarme.

- La propriété d' *Activation* est utilisée pour indiquer si l'alarme devra être déclenchée ou non. Dans le premier cas, l'alarme se déclenche quand la condition prévue par le mode événement devient active. Dans le dernier cas, l'alarme reste à l'état actif lorsque la situation persiste. Cette propriété peut également être utilisée pour indiquer que cette alarme doit être utilisée comme un seul événement. Dans ce cas, l'alarme sera déclenchée mais ne se traduira pas dans un état d'alarme. Au contraire, un événement sera enregistré dans la mémoire interne et éventuellement vers la carte CompactFlash.
- La propriété *Délai* est utilisée pour indiquer combien de temps l'alarme doit perdurer avant de devenir active. Cette propriété spécifie également la quantité de temps durant laquelle la condition d'alarme ne doit plus exister avant reprise ultérieure activations.
- La propriété *Accepter* est utilisée pour indiquer si l'utilisateur sera tenu d'accepter explicitement une alarme avant qu'elle ne soit plus affichée.
- La propriété *Priorité* est utilisée pour contrôler l'ordre dans lequel les alarmes sont affichées parcrimson. Plus la valeur numérique du champ est élevée, plus l'alarme sera affichée en début de liste.
- La propriété *Sirène* est utilisée pour indiquer si oui ou non l'activation de cette alarme devrait aussi activer l'appareil sondeur cible. Ce ce dernier est actif, l'affichage du panneau clignote également afin de mieux attirer l'attention sur l'état d'alarme.
- La propriété *Envoyer à* spécifie l'entrée du carnet d'adresses e-mail à laquelle un message doit être envoyé lorsque une alarme est activée. Reportez-vous au chapitre concernant la configuration e-mail.

## PROPRIETES D'ACTIVATION

Voici les propriétés d'un tag numérique sur l'onglet Activation...



### POUR CHAQUE ACTIVATION

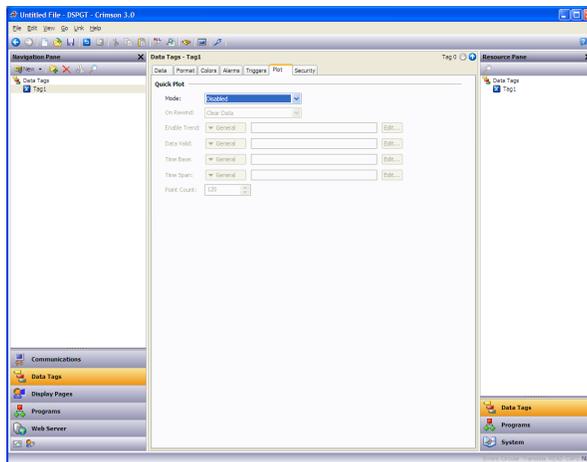
- La propriété *Mode d'activation* est le même pour l'onglet Alarmes.
- La *Valeur de l'hystérésis* et les propriétés sont décrites pour l'onglet Alarmes.

- La propriété *Retard* est identique à celle de l'onglet Alarmes..
- La propriété *Action* permet d'indiquer l'action qui doit être effectuée lors de l'activation du déclencheur. Reportez-vous à la section Ecriture d'actions pour obtenir une description de la syntaxe utilisée pour définir les différentes actions disponibles.

## PROPRIÉTÉS DE PLOT

La fonctionnalité Quick Plot a été rajoutée aux tags numériques et offre une méthode simple de suivi graphique des valeurs des tags. Une fois active et configurée, il est possible de marquer un tag sur une page d'affichage depuis la catégorie "Primitives de base" du Panneau de ressources. Cliquez sur la primitive "Quick Plot" et déplacez-la sur la page d'affichage désirée. Redimensionnez-la si nécessaire.

L'onglet Plot présente les propriétés suivantes...



- La propriété *Mode* permet de définir la manière dont les données sont enregistrées. Le mode *continu* enregistre sur tampon circulaire, supprimant au fur et à mesure les valeurs les plus anciennes. Il s'agit du mode le plus communément utilisé. La relative *coup unique* lancera enregistrement lorsque l'activation devient True et s'arrêtera une fois le tampon rempli ou l'activation False. Les valeurs de durée utilisées pour rechercher la position dans le slot seront liées au moment du démarrage de ce dernier. La propriété *coup unique absolu* est identique, mais les valeurs de durée sont basées sur zéro.
- *Sur retour* permet de préciser quoi faire en cas de retour dans la durée, ce qui est possible du fait que la base de durée est une variable. Les options disponibles permettent alors soit d'effacer les données après la durée pour laquelle nous sommes revenus en arrière, ou de les transférer dans le tampon, de manière à ce que les anciennes données soient retenues mais transférées.
- *Activer* permet de lancer et d'arrêter la tendance.
- *Données valides* permet d'enregistrer les vides dans les données sans mettre fin à la tendance et, de fait, d'abandonner toutes les données au moment du redémarrage.

- Par défaut, la *base de durée* est celle du système. Pour les applications spécialisées (telles que les enregistrements depuis un contrôleur externe) cette dernière peut être externe au système.
- *Plage de durée* correspond au nombre de bases de durées à enregistrer dans le tampon. Veuillez noter que cette dernière est généralement plus importante que la plage de décompte de points et que, avec cette variable, elle permet de définir le nombre de points que chaque slot peut accueillir.
- *Nombre de points* correspond au nombre de points à stocker dans le tampon. Le quick plot étant conçu pour un affichage de base des changements d'un tag sur la durée, cette valeur est généralement peu importante, inférieure par exemple au nombre de pixels de l'affichage.

## PROPRIETES DE SECURITE

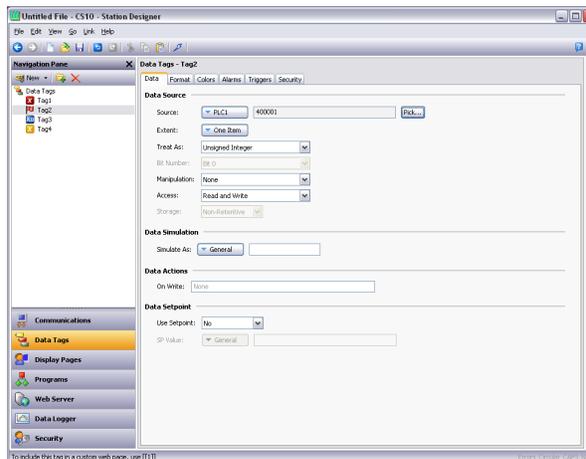
Se référer au chapitre Sécurité pour plus de détails Utilisation de descripteurs Sécurité.

## TAGS DRAPEAUX

Un tag drapeau représente une ou de plusieurs- valeurs on-off et est considéré comme possédant un type de données internes sous forme de nombre entier, quel que soit le type de données sous-jacentes. Les tags drapeaux marqués permettent des transformations simples entre les données brutes et les données qui seront utilisées par Crimson.

## PROPRIETES DE DONNEES

Un tag drapeau possède les propriétés suivantes sur son onglet Données...



## SOURCE DE DONNEES

- La propriété *Source* définit l'emplacement où le tag reçoit ses données. La liste déroulante peut être utilisée pour sélectionner une expression générale, un autre tag de données ou un élément d'un périphérique distant.
- La mesure la *Propriété* est utilisée pour choisir entre une balise de l'élément unique ou un tableau. Si vous sélectionnez un tableau, vous devez entrer le

nombre requis d'éléments. Les tableaux ne sont pas autorisés pour les balises dont la source est une expression. Pour les tags mappés, le nombre exact de registres pouvant être lus à partir des périphériques à distance dépend du type des registres à laquelle le tag est associé.

- La valeur *Traiter comme* est utilisée pour définir comment les valeurs on ou off devant être tirées des données de communication premières et *vice versa*. Les paramètres suivants peuvent être disponibles, en fonction du type de données sous-jacentes...

TRAITER COMME	RÉSULTAT
Entier non signé	La balise peut être True si la donnée est non-nulle, ou False si elle est nulle. Une valeur True sera écrite comme une valeur entière de 1, tandis qu'une valeur False sera écrite comme nulle. Pour un tableau mappé, chaque élément du tableau correspond à un seul élément de comm. de données. Ce paramètre est disponible pour toutes les données comm de 8 bits ou plus.
Point flottant	La balise peut être True si la donnée est non-nulle, ou False si elle est nulle. Une valeur True permet d'être écrit en 32-bit virgule flottante valeur de 1, tandis qu'une valeur False sera écrite comme zéro. Pour un tableau mappé, chaque élément du tableau correspond à un seul élément des comm. de données. Ce paramètre est disponible pour les données comm. d'exactly 32 bits de taille.
Bit Array Little Endian	Un seul bit est extrait de ces données. Pour un élément unique, le champ Nombre sélectionne l'outil, avec le bit le moins significatif étant 0. Pour un tableau, chaque élément est un seul bit. Le premier élément du tableau est le dernier bit significatif, le second est l'élément le plus important dans la suite et ainsi de suite. Un tableau 8 éléments mappés à un type de données octet dans un automate sera donc lu tous les 8 bits d'un registre unique.
Bit Array Big Endian	Comme ci-dessus, sauf que les bits sont inversés, avec un champ de bits de Bit 0 permettant d'accéder au bit le plus significatif.

- La propriété *Numéro de bit* extrait un bit simple à partir d'éléments de données multi-bits pour les tags mappés.
- La propriété *Manipulation* définit la transformation qui est appliquée à l'état du tag après que l'option Traiter comme la logique a été appliquée lors de la lecture des données. La seule option possible est d'inverser l'état de la balise. Il n'existe pas grand-chose d'autre que vous pouvez faire avec une valeur de bit unique !
- La propriété *D'accès* est utilisée pour les tags mappés afin de définir le type d'opérations de communications autorisées. Les tags internes sont toujours en lecture et écriture autorisées, et les tags d'expression sont toujours en lecture seule.

- La propriété *Stockage* est utilisée pour indiquer si la balise sera conservée grâce au cycle de l'appareil cible. Elle est généralement utilisée pour les balises internes, les tags mappés en lecture seule sont également retenus.

#### SIMULATION DE DONNEES

- La propriété *Simuler sous* définit la valeur devant être utilisé pour le tag lorsque vous travaillez dans l'éditeur de page. La saisie d'une valeur sensible permet une meilleure représentation de l'apparence de la page. Cette valeur est également utilisée comme valeur par défaut de la balise par le dispositif cible, si la communication est globalement désactivée.

#### ACTIONS DE DONNEES

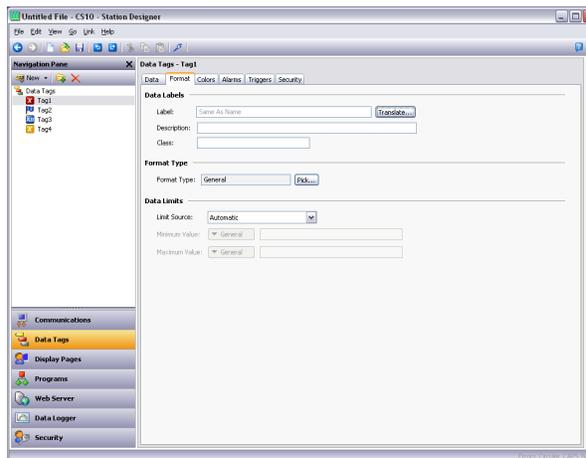
- La propriété *Ecrire sur* définit une action qui peut être invoquée lorsque le tag est modifié. La variable système *Données* contiendra la valeur de nouvelles données lorsque l'écriture se produit et quand l'action est exécutée. Le recours aux propriétés d'écriture est abordé plus loin dans ce chapitre.

#### POINT PREDEFINI

- La propriété *Utilisation du point prédéfini* permet d'activer ou de désactiver un point de consigne pour cette balise.
- La propriété *Valeur SP* définit une expression ou un autre tag que celui devant être suivi. Cette consigne peut ensuite être utilisée dans des alarmes ou des primitives afin de mettre en œuvre diverses fonctions.

### PROPRIETES DE FORMAT

Un tag drapeau possède les propriétés suivantes sur son onglet Format...



#### ETIQUETTES

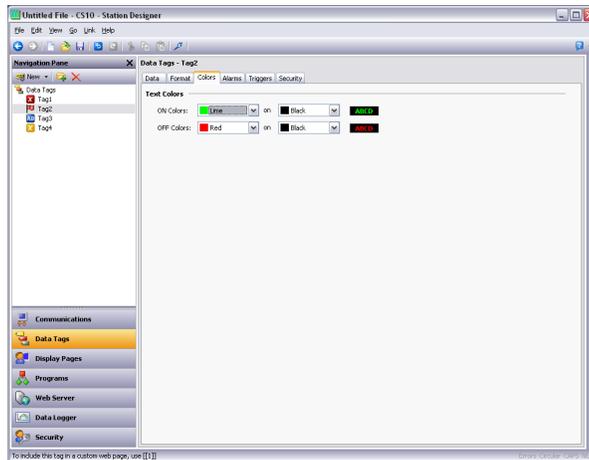
- La propriété *Etiquette* a été discutée ci-dessus sous attributs du tag.
- La propriété *Description* a été discutée ci-dessus sous attributs du tag.
- La propriété *Classe* est réservée pour une expansion future.

## FORMAT DE DONNEES

- Les propriétés *ON* et *OFF* font partie du format double sens de l'objet qui est sélectionné de façon permanente pour les balises marquées. Elles définissent le texte qui sera utilisé pour afficher le tag dans les états respectifs. Aucun autre type de format n'est supporté.

## PROPRIETES COULEUR

Un identificateur numérique possède les propriétés suivantes sur l'onglet Couleurs...

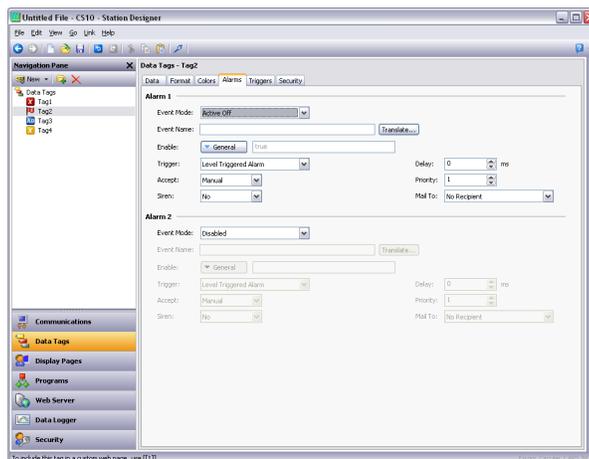


## COULEURS TEXTE

- Les propriétés *Couleurs ON* et *Couleurs OFF* font partie de l'état à deux couleurs qui est toujours utilisé pour les tags marqués. Elles définissent le premier plan et des paires de couleur de fond qui peuvent éventuellement être utilisées pour représenter le tag dans chaque état. Aucune autre couleur n'est supportée.

## PROPRIETES D'ALARME

Un tag drapeau possède les propriétés suivantes sur son onglet Alarmes...



## POUR CHAQUE ALARME

- La propriété *Mode Evénement* est utilisée pour indiquer la logique qui sera appliquée pour décider si l'alarme doit s'activer. Le tableau ci-dessous présente les modes disponibles.

MODE	L'ALARME SE DÉCLENCHERA QUAND...
Actif On	La balise est True.
Active Off	La balise est False.
Changement d'Etat	La balise change.

Les modes suivants ne sont disponibles que si une consigne est définie...

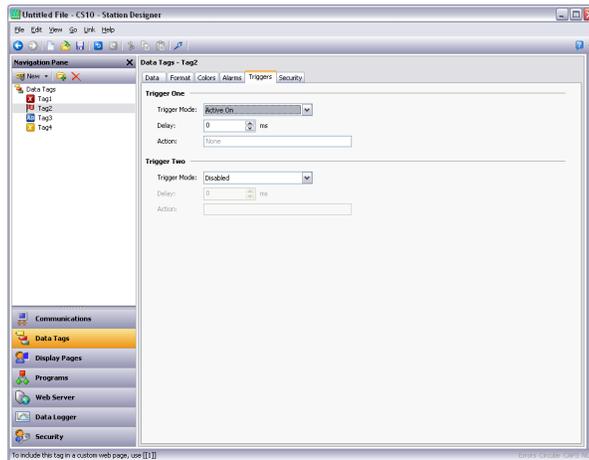
MODE	L'ALARME SE DÉCLENCHERA QUAND...
Pas égale à la SP	La balise n'est pas égale à son point prédéfini.
Off Lorsque SP sur	La balise ne répond pas à une consigne ON.
On Lorsque SP Off	La balise ne répond pas à une consigne OFF.
Égal à SP	La balise est égale à son point prédéfini.

- La propriété *Nom Evénement* définit le nom qui sera affiché dans la visionneuse d'alarme ou dans le journal des événements selon le cas. Crimson suggère un nom par défaut basé sur la légende du tag et le mode événement qui a été sélectionné.
- La propriété *Activer* définit une expression qui active ou désactive l'alarme. Le champ doit contenir une valeur différente de zéro.
- La propriété *Activation* est utilisée pour indiquer si l'alarme devra être déclenchée ou selon quel niveau. Dans le premier cas, l'alarme se déclenche quand la condition prévue par le mode événement passe au premier plan. Dans ce dernier cas, l'alarme reste active lorsque la situation persiste. Cette propriété peut également être utilisée pour indiquer que cette alarme doit être utilisée comme un seul événement. Dans ce cas, l'alarme sera déclenchée mais ne se traduira pas dans un état d'alarme. Par contre, l'événement sera enregistré dans la mémoire interne et éventuellement vers la carte CompactFlash.
- La propriété *Délai* est utilisée pour indiquer combien de temps l'alarme doit perdurer avant de devenir active. Cette propriété spécifie également l'adurée de temps où la condition d'alarme doit cesser avant reprise ultérieure.
- La propriété *Accepter* est utilisée pour indiquer si l'utilisateur sera tenu d'accepter explicitement une alarme avant qu'elle ne soit plus affichée.
- La propriété *Priorité* est utilisée pour contrôler l'ordre dans lequel les alarmes sont affichées. Plus la valeur numérique est importante, plus l'alarme sera affichée en début de liste.
- La propriété *Sirène* est utilisée pour indiquer si oui ou non l'activation de cette alarme devrait aussi activer l'appareil sondeur cible. Si ce dernier est actif, l'affichage du panneau clignote également afin de mieux attirer l'attention sur l'état d'alarme.

- La propriété *Envoyer* à spécifie l'entrée du carnet d'adresses e-mail à laquelle un message doit être envoyé lorsque une alarme est activée. Reportez-vous au chapitre concernant la configuration e-mail.

## PROPRIETES D'ACTIVATION

Un tag drapeau possède les propriétés suivantes sur son onglet Activateur...



### POUR CHAQUE ACTIVATEUR

- La propriété *Mode Activateur* est la même que pour l'onglet Alarmes.
- La propriété *Délai* est la même que pour l'onglet Alarmes.
- La propriété *Action* est utilisée pour indiquer quelles mesures doivent être effectuées lorsque le déclencheur est activé. Se référer au chapitre des actions d'écriture pour une description de la syntaxe utilisée pour définir les différentes actions qui sont disponibles.

## PROPRIETES DE SECURITE

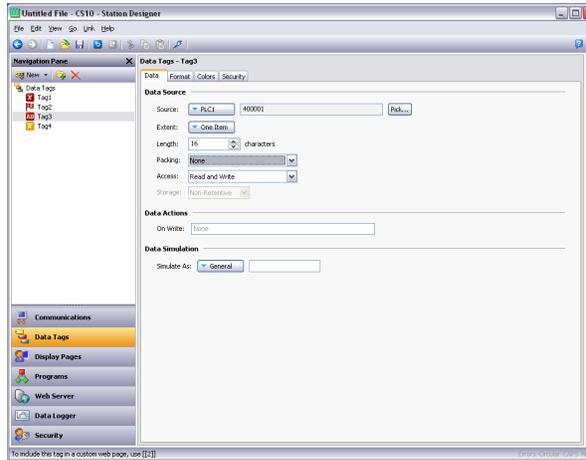
Se référer au chapitre Sécurité pour plus de détails sur les descripteurs de Sécurité.

## LIGNES DE TAGS

Une ligne de tag représente une ou plusieurs chaînes de caractères Unicode. Si Crimson 3 fonctionne entièrement en Unicode, il peut lire et écrire des chaînes de 8 bits de sources. Les tags mappés supportent plusieurs encodages, permettant à un ou plusieurs caractères d'être extraits à partir d'un registre.

## PROPRIETES DE DONNEES

Une ligne de tag possède les propriétés suivantes sur son onglet Données...



### SOURCE DE DONNEES

- La propriété *Source* définit l'emplacement où le tag reçoit ses données. Les résultats sont inscrits par défaut dans une balise interne, mais la liste déroulante peut être utilisée pour sélectionner une expression générale, une autre étiquette de données ou un élément à partir d'un périphérique distant.
- La propriété *Etendre* est utilisée pour choisir entre une balise de l'élément unique ou un tableau. Si vous sélectionnez un tableau, vous devez entrer le nombre requis d'éléments. Les tableaux ne sont pas autorisés pour les balises dont la source est une expression. Pour les tags mappés, le nombre exact de registres pour être lu à partir des périphériques à distance dépend du type des registre auxquels le tag est associé, de la longueur et des paramètres de packing.
- La propriété *Longueur* définit la longueur de la chaîne. Les chaînes internes non rétentives n'ont pas à avoir de longueur définie, car elles peuvent stocker une chaîne de n'importe quelle longueur raisonnable.
- La propriété *Packing* est utilisée pour les étiquettes mappées afin de définir comment la valeur de chaîne Unicode est tirée des données brutes comm. et *vice versa*. Les paramètres suivants peuvent être disponibles, en fonction du type de données sous-jacentes...

PACKING	RÉSULTAT
Aucun	Chaque élément de données comm. est utilisé comme source pour un seul caractère de la chaîne. Les valeurs de 8 bits seront traitées comme ASCII, tandis que les valeurs de 16 bits et plus seront traitées comme UNICODE.
ASCII Big Endian	Chaque unité 8-bit dans l'élément de données est utilisée comme source pour un seul caractère ASCII, les 8 bits les plus significatifs étant utilisés pour le premier caractère. Uniquement disponible pour les éléments de données de 16 bits ou plus.

PACKING	RÉSULTAT
ASCII Little Endian	Chaque unité 8-bit dans l'élément de données est utilisée comme source pour un seul caractère ASCII, les 8 bits les moins significatifs étant utilisés pour le premier caractère. Uniquement disponible pour les éléments de données de 16 bits ou plus.
Unicode Big Endian	Chaque unité de 16 bits dans l'article de données est utilisée comme source pour un seul caractère Unicode, les 16 bits les plus significatifs étant utilisés pour le premier caractère. Uniquement disponible pour les éléments de données de 32 bits.
Unicode Little Endian	Chaque unité de 16 bits dans l'article de données est utilisée comme source pour un seul caractère Unicode, les 16 bits les moins significatifs étant utilisés pour le premier caractère. Uniquement disponible pour les éléments de données de 32 bits.
Hex String Little Endian	Chaque unité de 4 bits dans l'article de données est utilisée comme source pour un caractère hexadécimal unique dans l'intervalle '0'-'9' et 'A'-'F', les 4 bits les moins significatifs étant utilisé en premier.
Hex String Big Endian	Chaque unité de 4 bits dans l'article de données est utilisée comme source pour un caractère hexadécimal unique dans l'intervalle '0'-'9' et 'A'-'F', les 4 bits les plus significatifs étant utilisé en premier.

- La propriété *Accès* est utilisée pour les tags mappés afin de définir le type d'opérations de communications devant être autorisées. Les tags internes sont toujours en lecture et écriture, et les tags d'expression sont toujours en lecture seule.
- La propriété *Stockage* est utilisée pour indiquer si la balise sera conservée via le cycle de l'appareil cible. Cette dernière est généralement utilisée pour les balises internes, mais les tags mappés en lecture seule ont aussi leurs valeurs retenues.

#### SIMULATION DE DONNEES

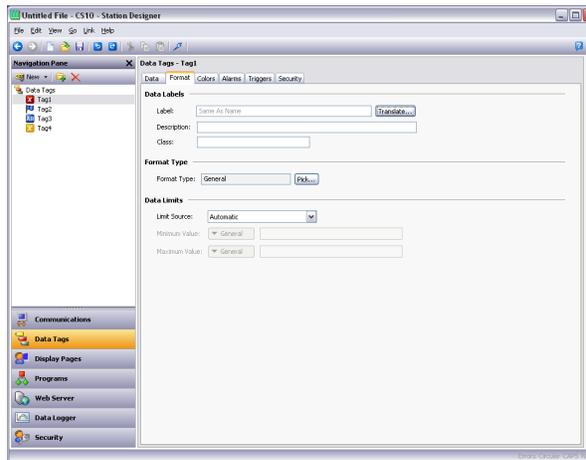
- La propriété *Simuler sous* définit la valeur supposée être utilisé pour le tag lorsque vous travaillez dans l'éditeur de pages. La saisie d'une valeur sensible permet une meilleure représentation de l'apparence de la page. Cette valeur est également utilisée comme valeur par défaut du tag par le dispositif cible, si la communication est globalement désactivée.

#### ACTIONS DONNEES

- La propriété *Sur Ecriture* définit une action qui peut être invoquée lorsque le tag est modifié. Le système de *Données* variables contiendra la valeur des nouvelles données lorsque l'écriture se produit et quand l'action est exécutée. Le recours aux propriétés d'écriture est abordé plus loin dans ce chapitre.

## PROPRIETES DE FORMAT

Une ligne de tags possède les propriétés suivantes sur son onglet Format...

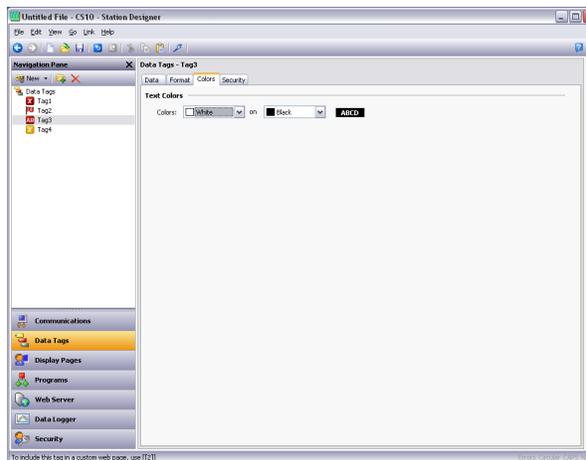


### ETIQUETTES DONNEES

- La propriété *Légende* a été discutée ci-dessus sous attributs du tag.
- La propriété *Description* a été discutée ci-dessus sous attributs du tag.
- La propriété *Classe* est réservée à une expansion future.

## PROPRIETES COULEUR

Une ligne de tags possède les propriétés suivantes sur l'onglet Couleurs...



### COULEURS TEXTE

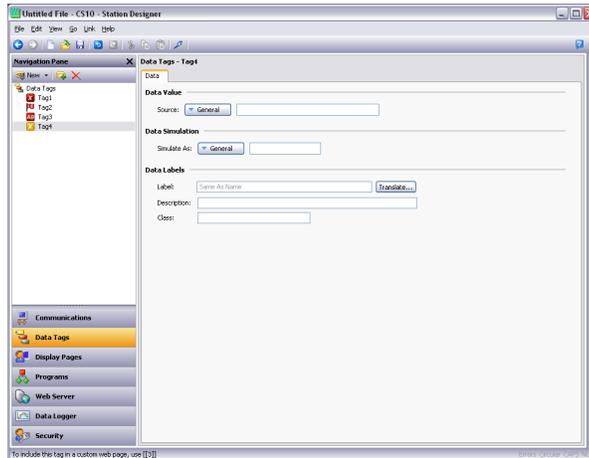
- La propriété *Couleurs* fait partie de la coloration fixe qui est toujours utilisée pour les tags drapeau. Elle définit le premier plan et une paire de couleurs de fond qui peuvent éventuellement être utilisées pour représenter la balise. Aucune autre couleur n'est supportée.

## PROPRIETES DE SECURITE

Se référer au chapitre Sécurité pour plus de détails sur les descripteurs de Sécurité.

## TAGS DE BASE

Les tags de base sont utilisés pour représenter des constantes ou des expressions...



### VALEUR DE DONNEE

- La propriété *Valeur Donnée* définit la valeur du tag. Elle doit être une expression. La balise elle-même adoptera le type de données de l'expression qui est utilisée.

### SIMULATION DE DONNEES

- La propriété *Simulersous* définit une valeur à utiliser par défaut pour le tag lors de l'édition des Pages d'Affichage. La saisie d'une valeur sensible permet une meilleure représentation de l'apparence de la page. Cette valeur est également utilisée comme valeur par défaut de la balise par le dispositif cible, si la communication est globalement réduite.

### ETIQUETTES DONNEES

- La propriété *Etiquette* a été discutée ci-dessus sous attributs du tag.
- La propriété *Description* a été discutée ci-dessus sous attributs du tag.
- La propriété *Classe* est réservée à une expansion future

## TAGS FLUX DE DONNEES

Comme vous l'aurez remarqué, un certain nombre de changement se produisent au niveau des données des tags numériques, entre les données de com. et les valeurs utilisées par Crimson. Ceux-ci peuvent être configurés afin de gérer les données de la manière que vous souhaitez, mais la manière exacte dont ils opèrent pour les tags numériques mérite davantage d'attention.

## LECTURE TAGS NUMERIQUES

Lorsque les données sont lues à partir d'un périphérique, les étapes suivantes se suivent...

- Le conducteur comm lit une valeur fondée sur le réglage de l'adresse qui a été défini pour la source de la balise. Selon le type de l'adresse, le pilote peut combiner plus d'un registre afin de créer la valeur des données. Par exemple, la lecture d'un seul mot comme valeur se traduira par deux registres lus et combinés par le pilote qui utilisera sa connaissance de la manière dont le périphérique organise les mots.
- Les données comm. sont ensuite modifiées selon la propriété de manipulation de la balise en question. Ces processus effectuent des changements au niveau de l'octet des données, généralement pour tenir compte des incompatibilités de conducteur ou d'autres situations où les données ne sont pas dans la forme que le conducteur comm s'attend à rencontrer.
- Les données manipulées sont ensuite interprétées en liaison avec la propriété Traiter Sous du tag.
- Si l'échelle est définie, les données sont ensuite interprétées et échelonnées selon le domaine défini pour la balise. Le résultat de la mise à l'échelle peut être d'un type différent de celui des données interprétées (ainsi la valeur en virgule flottante peut être réduite à un nombre entier ou *vice versa*). En supposant que l'échelle est définie, le résultat de cette étape permet alors de définir la valeur finale et le type de données de la balise.

## ECRITURE TAGS NUMERIQUES

Lorsque les données sont écrites sur un périphérique, les étapes suivantes se produisent...

- Si l'échelle est définie, le domaine et l'image sont inversées, la conversion des données à une valeur non mise à l'échelle est définie selon la propriété Traiter Sous.
- Si les données non ajustées sont plus grandes que les données comm., les bits les plus hauts placés sont supprimés, produisant une version dépouillée des données, appropriée pour l'étape suivante.
- Les données dépouillées sont ensuite modifiées selon la propriété de manipulation, qui a infirmé la transformation appliquée ci-dessus.
- Le conducteur prend en charge les données de communication et les inscrit dans une ou plusieurs entrées du dispositif cible en fonction du type de l'adresse.

## UTILISATION SUR ECRITURE

Une balise apposée sur la propriété Ecrire contient une action qui est exécutée quand un changement lui est apporté. Alors que l'action est exécutée, un système appelé Données est défini à la nouvelle valeur, ce qui permet aux nouvelles données d'être examinées. Il existe trois utilisations typiques de cette fonction...

- Les tags normaux peuvent avoir une propriété Sur lecture définie permettant à certaines actions d'être prises à la demande. Par exemple, une base de données

peut avoir besoin de stocker la valeur d'une balise dans deux formats, l'un étant le format tag original et l'autre étant une version transformée. Bien qu'il existe d'autres façons de faire cela, une méthode consiste à utiliser la propriété Ecrire puis à exécuter un programme pour calculer et stocker la version transformée.

- Les tags en lecture seule peuvent être placés en écriture via la propriété Ecrire. Bien que cela semble bizarre, imaginez, par exemple, que d'une boucle PID a une propriété en lecture seule indiquant la puissance de sortie en cours, et une propriété en lecture-écriture définissant la puissance de sortie manuelle. Vous pouvez définir des champs d'affichage pour permettre la saisie des données en mode manuel, et les capturer à l'aide de la propriété Sur Ecriture, enregistrant ainsi les valeurs de la puissance de sortie manuelle.
- Des transformations complexes peuvent être mises en œuvre par la définition d'un tag d'expression pour effectuer la transformation de l'action Sur Ecriture pour l'inverse. Par exemple, une balise pourrait être mise à `Sqrt ([40001])` afin de prendre la racine carrée d'une valeur dans un Modbus API. Comme il s'agit d'une balise d'expression, cette dernière est par définition en lecture seule, mais toutes les écritures peuvent être autorisées par la définition d'une égalité Sur Ecriture égale à `[40001] = Data * Data`, renversant ainsi le calcul de racine carrée.

## PROPRIETES TAGS FLECHES

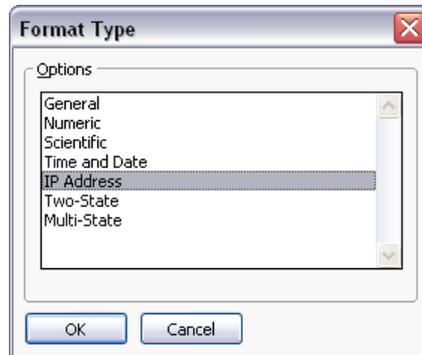
[A compléter]

# UTILISATION DES FORMATS

Les tags numériques peuvent être de différents formats de données, les tags drapeaux et lignes de tags possédant des formats fixés à deux États et Général, respectivement. Chaque type de format aura une valeur de données et les convertira vers ou depuis une chaîne de texte

## TYPES DE FORMATS

Les formats suivants sont supportés...



- Le format *Général* prévoit la mise en forme simple des valeurs et la conversion des valeurs numériques pour les valeurs décimales signées. Le format général n'a pas de propriétés de configuration, et est le format par défaut pour les lignes de tags. Il est également implicitement utilisé par les balises de base.
- Le format *Numérique* prend un point flottant ou entier et le convertit en chaîne, en utilisant une base de nombre spécifique et en sélectionnant le nombre requis de chiffres avant et après la virgule. Il peut également ajouter un préfixe et une chaîne d'unités à la valeur, et manipuler des valeurs signées ou non signées.
- Le format *Scientifique* prend un point flottant ou entier et le convertit en format exponentiel, en sélectionnant le nombre requis de chiffres après la virgule. Il peut également ajouter un préfixe et une chaîne d'unités à la valeur.
- Le format *D'heure et date* prend une valeur entière et le traite comme un certain nombre de secondes écoulées depuis le 1<sup>er</sup> Janvier 1997. Il peut afficher le résultat en valeur de date, en valeur d'heure ou de deux heures, ou considérer la valeur comme des durées qui peuvent contenir plus de 24 heures. La date de mise en forme et les options de formatage de temps sont prises en charge pour permettre l'adaptation à diverses normes internationales.
- Le format *D'adresse IP* prend une valeur entière et l'affiche sous forme de quatre octets décimaux séparés par des points. Cela permet à un nombre de 32 bits pour être affiché comme une adresse IP sans configuration supplémentaire.
- Le format *Deux États* a une valeur numérique et affiche l'une des deux chaînes, selon que la valeur est nulle ou non nulle. C'est le type de format défini de façon permanente pour les tags drapeaux.

- Le format *Multi États* prend une valeur numérique et la compare à un tableau contenant des valeurs et des lignes. Soit la chaîne associée à une valeur de couplage de données est affichée, soit le format peut être configuré pour afficher la dernière chaîne d'une valeur ne dépassant pas les données associées.

## FORMAT GENERAL

Le format général n'a pas de propriétés.

## FORMAT NUMERIQUE

Le format numérique possède les propriétés suivantes...

**Data Format**

Number Base:  Sign Mode:

Digits Before DP:  Digits After DP:

Lead Character:  Group Digits:

---

**Format Units**

Prefix:

Units:

## FORMAT DE DONNEES

- La propriété *Nombre de base* définit la racine de la valeur affichée. Le code d'authentification paramètre le travail en base décimale, mais masque les chiffres en utilisant un astérisque. Beaucoup d'autres options seront désactivées lorsque le mode non-décimal est utilisé.
- La propriété *Mode signe* définit comment les données sont traitées, et la façon dont le panneau est affiché. Une valeur Non Signée affichera un nombre de 32 bits non signés. Une valeur de Signe mou affichera un signe moins pour les nombres négatifs et un espace pour les nombres positifs, alors qu'une valeur de Signe dur vous affichera un signe + initial.
- La propriété *Chiffres avant DP* définit le nombre de chiffres à afficher avant le point décimal. Pour les valeurs sans décimales, c'est le nombre total de chiffres et donc la taille du champ de données.
- La propriété *Chiffres après DP* définit le nombre de chiffres à être indiqués après la virgule. Pour les valeurs entières, le point décimal est inséré dans la représentation entière, de sorte que 1234 serait affichés et inscrits sous la forme 12.34 si cette propriété a été définie. Une valeur nulle supprime la virgule.
- La propriété *Caractère principal* définit comment les valeurs des zéros sont formatées. Les zéros placés en tête peuvent être conservés, remplacés par des espaces ou enlevés complètement. La suppression de certains d'entre eux peut parfois poser problème surtout si la valeur est centrée dans un champ.
- La propriété *Grouper Chiffres* permet l'insertion de séparateurs virgule tous les trois chiffres pour les nombres décimaux, avec des comportements analogues pour les bases de plusieurs autres.

### UNITES DE FORMAT

- La propriété *Préfixe* définit une chaîne à afficher avant la valeur numérique.
- La propriété *Unités* définit une chaîne à afficher après la valeur numérique.

### FORMAT SCIENTIFIQUE

Le format scientifique possède les propriétés suivantes...

The screenshot shows a dialog box titled "Data Format" with two sections: "Data Format" and "Format Units".

**Data Format**

- Mantissa Sign Mode: Soft Sign (dropdown menu)
- Exponent Sign Mode: Hard Sign (dropdown menu)
- Digits After DP: 5 (spin box)

**Format Units**

- Prefix: [text input] [Translate... button]
- Units: [text input] [Translate... button]

### FORMAT DE DONNEES

- La propriété *Mode signe* définit la manière dont le panneau est affiché sur la mantisse. Une valeur de Sign mou affichera un signe moins pour les nombres négatifs et un espace pour les nombres positifs, alors qu'une valeur de Signe dur vous affichera un signe + initial.
- La propriété *Mode exposant* définit la manière dont le panneau est affiché sur l'exposant. Une valeur de Chiffre mou affichera un signe moins pour les valeurs négatives et rien pour les valeurs positives, tandis qu'une valeur de Chiffre dur affichera un signe plus pour les valeurs positives.
- La propriété *Chiffres après DP* définit le nombre de chiffres à être indiquée après la virgule. Par définition, il ya toujours un chiffre avant la virgule au format scientifique. Une valeur nulle supprime la virgule.

### FORMAT UNITES

- La propriété *Préfixe* définit une chaîne à afficher avant la valeur numérique.
- La propriété *Unités* définit une chaîne à afficher après la valeur numérique.

## FORMAT HEURE ET DATE

Le format d'heure et de date possède les propriétés suivantes...

The screenshot shows a configuration window titled "Format Mode". It is divided into three sections: "Format Mode", "Time Format", and "Date Format".

- Format Mode:** A dropdown menu labeled "Field Contents:" is set to "Time Then Date".
- Time Format:**
  - "Time Format:" is set to "12 Hour (Civil)".
  - "Show Seconds:" is set to "No".
  - "AM Suffix:" is set to "AM" with a "Translate..." button.
  - "PM Suffix:" is set to "PM" with a "Translate..." button.
- Date Format:**
  - "Date Format:" is set to "Locale Default".
  - "Show Year:" is set to "As 2 Digits".
  - "Show Month:" is set to "As Digits".

### FORMAT MODE

- La propriété *Mode format* est utilisée pour indiquer si le champ devrait afficher l'heure, la date ou les deux. Dans ce dernier cas, il indique également l'ordre dans lequel deux éléments devraient être indiqués. Des options sont également fournies pour permettre aux valeurs de temps d'être traitées comme des périodes écoulées plutôt que comme une heure liée à une date. Par exemple, une valeur de 25,5 heures sera affichée sous la forme 25:30 dans un mode écoulé. Dans un mode temps conventionnel, elle affichera 00:30.

### FORMAT HEURE

- La propriété *Format de l'heure* est utilisée pour choisir entre le format à 12 heures (civil) ou 24 heures (militaire). Comme avec d'autres propriétés, laisser cet ensemble de paramètres régionaux par défaut permettra à Crimson de choisir un format adapté en fonction de la langue sélectionnée dans le panneau de commandes.
- Les propriétés de *Suffixe AM* et *PM* sont utilisées avec le mode 12 heures pour indiquer si l'heure indiquée est celle du matin ou de l'après-midi. Si vous laissez la propriété non définie, Crimson utilisera une valeur par défaut.
- La propriété *Afficher Secondes* est utilisée pour indiquer si le champ de l'heure devrait inclure les secondes, ou s'il doit simplement comprendre heures et minutes.

### FORMAT DATE

- La propriété *Format de Date* est utilisée pour indiquer l'ordre dans lequel les différents éléments d'une date (à savoir jour, mois et année) doivent être affichés.
- La propriété *Afficher année* est utilisée pour indiquer si le champ date doit comprendre l'année, et si oui, combien de caractères doivent être inscrits pour cet élément.

- La propriété *Afficher mois* est utilisée pour indiquer si le mois doit être affiché sous forme de chiffres (soit de 01 à 12) ou comme son nom court (soit de Jan à Déc).

## FORMAT ADRESSE IP

Le format d'adresse IP n'a pas de propriétés.

## FORMAT DEUX-ETATS

Le format Deux États possède les propriétés suivantes...

**Data Format**

ON State:  [Translate...](#)

OFF State:  [Translate...](#)

## DATE FORMAT

- La propriété *ON* définit le texte qui sera affiché si la valeur est non nulle.
- La propriété *OFF* définit le texte qui sera affiché si la valeur est nulle.

## FORMAT MULTI-ETATS

L'état multi-format possède les propriétés suivantes...

**Format Control**

States:  [Edit...](#)

Limit:

Default:  [Translate...](#)

Match Type:

**Format States**

	Data	Text	
1:	<input type="text" value="1"/>	<input type="text" value="ONE"/>	<a href="#">Translate...</a>
2:	<input type="text" value="2"/>	<input type="text" value="TWO"/>	<a href="#">Translate...</a>
3:	<input type="text" value="3"/>	<input type="text" value="THREE"/>	<a href="#">Translate...</a>
4:	<input type="text" value="4"/>	<input type="text" value="FOUR"/>	<a href="#">Translate...</a>

## CONTROLE FORMAT

- La propriété *États* définit combien d'États multi-formats seront contenus, jusqu'à un maximum de 500 entrées. La fenêtre affichant le format sera mise à jour pour afficher le nombre requis de données et propriétés de texte.
- La propriété *Limite* définit combien d'États seront utilisés lors des recherches de données selon ce format. Elle peut être ajustée de façon dynamique, tandis que le nombre absolu des États est statiquement défini. Cette propriété est utile lorsque les champs sont remplis à l'exécution, car elle permet d'ignorer les champs inutilisés pendant le processus de saisie de données.

- La propriété *Défaut* définit une chaîne à afficher si les données ne peuvent pas être comparées avec les Etats définis. Si aucune valeur n'est fournie, la représentation numérique de l'Etat non suivi est affichée entre parenthèses.
- La propriété *Type de ciblage* définit comment les données sont comparées à différents Etats. Si l'option *Discret* est sélectionnée, les données du tag doivent correspondre à la valeur de données d'un état donné, afin que ce dernier soit utilisé. Si l'option *Par ordre* est sélectionnée, Crimson supposera que les valeurs des données d'état sont dans l'ordre numérique croissant, et utilisera une valeur d'état.

#### ETATS FORMAT

- Les propriétés *Données* et *Texte* définissent la valeur des données et du texte d'affichage de chaque Etat dans ce format. Les états avec des champs de texte vides sont invalides et sont ignorées.

#### COMMANDES FORMAT



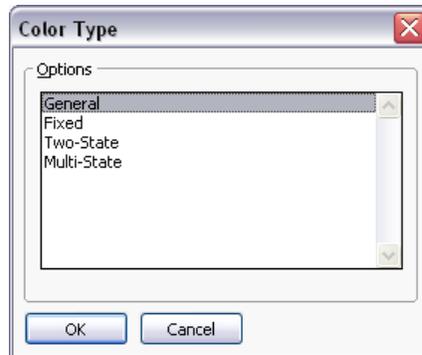
Les objets au format multi-format peuvent également fournir des boutons pour permettre à leurs différents états et aux propriétés associées d'être exportés ou importés depuis des fichiers texte Unicode. Ces fichiers peuvent ensuite être modifiés par une application telle que Microsoft Excel.

# UTILISATION DES COULEURS

Les tags numériques peuvent avoir l'une des couleurs sélectionnées, les tags drapeaux et lignes de tags ayant leurs couleurs respectives. Chaque couleur aura une valeur de données et sera convertie à une paire de premier plan et couleur de fond.

## TYPES DE COULEURS

Les couleurs suivantes sont supportées...



- La coloration *Générale* renvoie toujours blanc sur noir.
- La coloration *Fixe* retourne toujours une paire fixe de couleurs.
- La coloration *Deux-Etats* prend une valeur numérique et choisit l'une des deux paires de couleur selon que la valeur est nulle ou non nulle. Il s'agit du sens de coloration permanente pour les tags drapeaux.
- La coloration *Multi-Etats* prend une valeur numérique et la compare à un tableau contenant les valeurs des données et des paires de couleurs. Soit une paire de couleurs associée à une valeur de couplage de données est sélectionné, soit le sélecteur peut être configuré pour utiliser la paire de couleurs avec un ensemble de données associées à une valeur ne dépassant pas les données.

## OPERATIONS GENERALES

La coloration générale n'a pas de propriétés.

## COULEURS FIXES

La mise en couleur fixe possède les propriétés suivantes...



- La propriété *Couleurs* définit les couleurs à utiliser tout le temps.

## COLORATION DEUX-ETATS

La coloration Deux-Etats possède les propriétés suivantes...

- La propriété *ON* définit les couleurs à utiliser lorsque la balise est non nulle.
- La propriété *OFF* définit les couleurs à utiliser lorsque la balise est nulle.

## COLORATION MULTI-ETATS

La coloration Multi-Etats possède les propriétés suivantes...

### CONTROLE FORMAT

- La propriété *États* définit combien de sélecteurs multi-États seront contenus, jusqu'à un maximum de 500 entrées. La fenêtre affichant le sélecteur sera mise à jour pour afficher le nombre requis de données et les propriétés de texte.
- La propriété *Couleurs par défaut* définit les couleurs à utiliser si les données ne peuvent pas être comparées avec les États définis.
- La propriété *Type de ciblage* définit comment les données sont comparées à différents États. Si l'option Discrets est sélectionnée, les données du tag doivent correspondre à la valeur de données d'un état donné, afin que cet état soit utilisé. Si l'option Par ordre est sélectionnée, Crimson suppose que les valeurs des données d'état sont dans l'ordre numérique croissant et utilisera une valeur d'état.

### ÉTATS FORMAT

- Les propriétés *Données* et *Couleurs* définissent les données et les valeurs de couleur pour chaque état.

## COMMANDE COULEURS



Les objets de coloration multi-états possèdent également des boutons pour permettre à leurs Etats et aux propriétés associées d'être exportés ou importés depuis des fichiers texte Unicode. Ces fichiers peuvent ensuite être modifiés par une application telle que Microsoft Excel. Un bouton supplémentaire permet au champ de données de coloration d'être synchronisé avec le champ de données d'un Etat à plusieurs format d'objet configuré pour la même balise, en évitant que vous ayez à entrer deux fois les mêmes valeurs

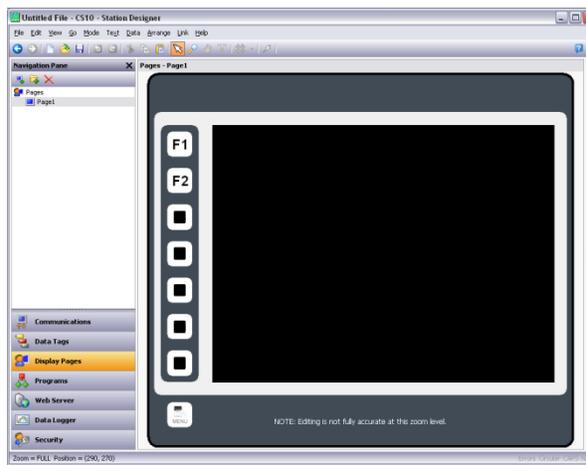


# CREATION DE PAGES D’AFFICHAGE

Pour accéder au nouvel éditeur graphique Crimson, sélectionnez la catégorie des pages d’affichage dans le panneau de navigation. Cet éditeur a été conçu dans le but de permettre la création rapide et efficace de pages d’affichage attractives, tout en gardant une flexibilité maximale.

## PRINCIPES DE BASE DE L’ÉDITEUR

L’éditeur graphique est représenté ci-dessous dans son état initial...



Le panneau d’édition affiche une représentation du périphérique cible, intégrant à la fois les touches et la zone de l’écran. Lorsque le zoom est à son niveau minimal, le panneau est affiché intégralement, même si pour cela il est nécessaire d’allouer moins d’un pixel de l’affichage de votre ordinateur pour chaque pixel de l’affichage du périphérique cible. Dans de telles situations, il est encore possible de visionner les pages et la plupart des fonctionnalités d’édition peuvent être exécutées, mais la précision peut s’en trouver réduite. Un message d’avertissement concernant cet effet sera affiché.

## TRAVAILLER AVEC LES PAGES

La manipulation des pages d’affichage à travers la liste de navigation est intuitive, elle s’opère de la même manière que pour tout autre objet dans la base de données de Crimson. Ceci étant dit, n’oubliez pas que les pages peuvent être copiées d’une base de données à une autre. Il suffit de sélectionner une page dans le panneau de navigation d’une base de données et de la faire glisser dans la catégorie correspondante de la base de données cible. Il est ainsi très simple de construire de nouvelles bases de données en combinant des styles de pages déjà utilisés.

## MODIFIER LE NIVEAU DU ZOOM

Pour effectuer un zoom avant ou un zoom arrière, le moyen le plus simple est d’utiliser la molette de la souris. Si vous ne disposez pas d’une souris à molette, vous pouvez utiliser le mode zoom de l’éditeur en sélectionnant la loupe dans la barre d’outils. Dans ce mode, un clic gauche produira un zoom avant, un clic droit ou un clic gauche en maintenant enfoncé le

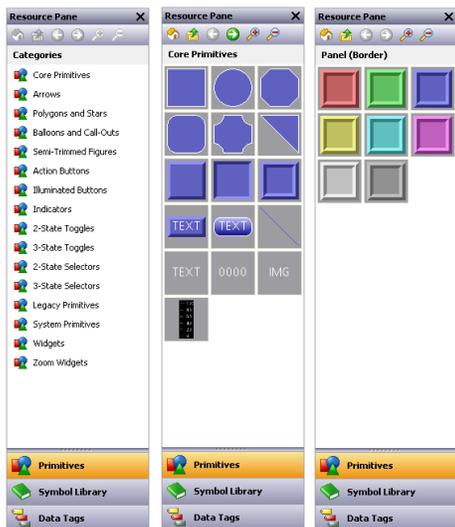
bouton **CTRL** produira un zoom arrière. Vous pouvez aussi utiliser les commandes de zoom situées dans le menu Affichage.

Le premier niveau du zoom a pour effet de passer d'une vue du panneau en entier à un affichage 1:1, avec l'écran du périphérique cible au centre de votre fenêtre d'édition. Les autres niveaux du zoom sont effectués de manière à maintenir visibles les données situées sous le pointeur de votre souris. Il est ainsi plus facile de choisir la zone de l'écran que vous voulez examiner plus en détails.

## LE PANNEAU DE RESSOURCES

La manière classique de construire des pages d'affichage est de faire glisser des objets à partir du panneau de ressources. Vous pouvez choisir de faire dérouler le panneau de ressources en cliquant sur la flèche située sur le côté droit de la fenêtre, ou de le laisser fermé afin de maximiser l'espace de travail disponible dans la fenêtre. Le panneau de ressources possède trois catégories...

### PRIMITIVES



La catégorie Primitives permet d'accéder aux composants principaux nécessaires pour assembler les pages d'affichage. Elle est située à gauche de l'écran dans ses diverses formes. Vous remarquerez que le niveau supérieur contient un certain nombre de sous-catégories, chacune d'entre-elles donnant accès à un certain nombre de primitives. En cliquant sur un icône vous affichez une sous-catégorie et ses primitives. En cliquant sur une primitive donnée vous affichez des versions de cette primitive avec des couleurs prédéfinies. Vous pouvez utiliser les icônes situés dans la barre d'outils pour vous déplacer parmi les sous-catégories, pour accéder au niveau supérieur ou pour modifier le nombre de primitives affichées

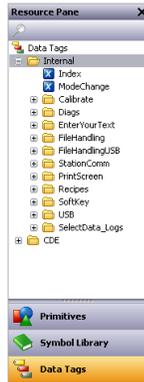
par rang. Une description des primitives est donnée dans le chapitre suivant.

### BIBLIOTHEQUE SYMBOLES



La catégorie Bibliothèque Symboles fonctionne de manière très semblable à la catégorie des primitives. Elle permet d'accéder à un certain nombre de sous-catégories, chacune d'entre-elles contenant un certain nombre de symboles prédéfinis. En cliquant sur un symbole donné vous accédez à des versions de ce symbole avec des couleurs prédéfinies, même si cette option est moins utilisée qu'elle ne l'est pour les primitives. Prenez le temps d'explorer la Bibliothèque Symboles – elle contient plusieurs milliers d'images, son utilisation adéquate vous permettra de produire des bases de données plus attractives et maniables.

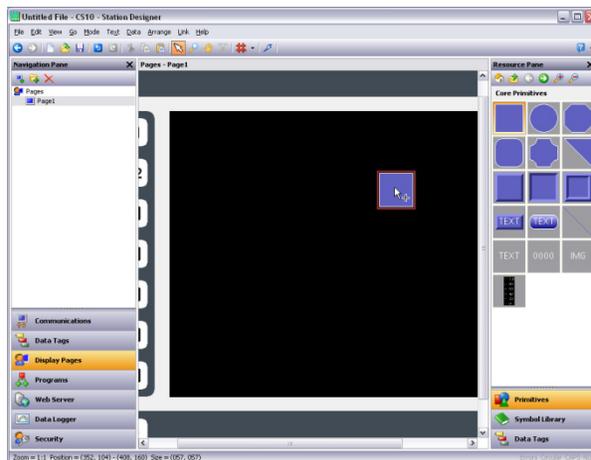
## TAGS DE DONNEES



La catégorie Tags de Données contient une arborescence de tous les tags de la base de données active. Elle est utilisée à la fois pour faire glisser des tags directement dans la page d’affichage et aussi pour permettre d’accéder à des tags pendant la configuration des propriétés des primitives. En faisant glisser un tag dans une page vous créez une boîte de données qui est liée à ce tag, avec toutes les propriétés de formatage basées sur celles définies par le tag. Vous pouvez aussi sélectionner et faire glisser plusieurs tags en utilisant les touches **SHIFT** et **CTRL** de la manière habituelle. Ces options permettent d’ajouter de manière simple et rapide des données à une page.

## AJOUTER DES OBJETS A UNE PAGE

Comme il a été dit précédemment, vous pouvez faire glisser divers objets du panneau de ressources vers l’éditeur, les ajoutant ainsi à la page d’affichage. Les primitives appropriées seront créées pour les tags et les images. L’exemple ci-dessous illustre comment faire glisser une primitive rectangle vers une page en cliquant sur Primitives de base dans la catégorie Primitives...



## TRAVAILLER AVEC LES PRIMITIVES

Une description des opérations courantes effectuées sur les primitives est donnée dans les paragraphes suivants.

### SELECTION DES PRIMITIVES

Pour sélectionner une primitive d’affichage, il suffit de positionner le pointeur de votre souris sur celle-ci et de faire un clic gauche. Vous remarquerez que lorsque le pointeur est positionné sur une primitive, un contour rectangulaire bleu apparaît afin de vous indiquer la zone qui sera sélectionnée. Lorsque la sélection est effectuée, le rectangle devient rouge et des poignées de redimensionnement de la primitive s’affichent. Si la primitive que vous voulez sélectionner est cachée derrière une autre, appuyez sur la touche **CTRL** afin de pouvoir effectuer la sélection.

Pour sélectionner plusieurs primitives à la fois, vous pouvez soit créer un rectangle de sélection autour des primitives choisies, soit sélectionner les primitives une par une en

maintenant la touche **SHIFT** enfoncée. Si plusieurs primitives sont sélectionnées, un rectangle rouge encadrera toutes ces primitives et vous pourrez utiliser les poignées pour redimensionner d'un seul coup l'ensemble des primitives. Du moment qu'un espacement minimal est respecté, la taille et la position relative des primitives restent inchangées lors de cette opération.

### UTILISATION DE LA BARRE RAPIDE

La barre Accès rapide est une barre d'outils flottante qui apparaît en haut à droite de la sélection active...



La barre apparaît tout d'abord floutée, elle devient plus nette lorsque vous approchez votre souris. Lorsque vous éloignez la souris la barre disparaît, elle ne réapparaîtra que si vous répétez la sélection ou si vous appuyez sur la molette de la souris. La barre Accès rapide permet l'accès rapide à un certain nombre de fonctionnalités courantes en minimisant le déplacement de la souris. La barre peut être activée ou désactivée en utilisant une commande du menu Affichage.

### DEPLACER LES PRIMITIVES ENTRE LES PAGES

Vous pouvez déplacer les primitives à l'intérieur d'une page d'affichage en les faisant glisser de la manière habituelle, mais vous pouvez aussi copier les primitives d'une page à une autre. Pour cela, sélectionnez la primitive que vous voulez copier et faites-la glisser vers le panneau de navigation. Si le panneau est masqué, faites-le défiler pour le rendre visible en positionnant votre souris sur la flèche. Positionnez votre souris sur la page cible choisie pour la sélectionner. Faites maintenant glisser la primitive de retour vers l'éditeur et déposez-la sur la page sélectionnée. Si vous voulez déplacer la primitive d'une page à une autre plutôt que de faire un copier-coller, vous devez maintenir la touche **CTRL** enfoncée, contrairement à ce qui se passe pour les opérations de déplacement à l'intérieur d'une même page.

### DEPLACER LES PRIMITIVES ENTRE LES BASES DE DONNEES

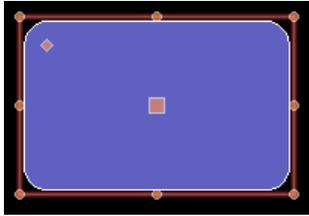
Il est tout aussi simple de déplacer les primitives d'une base de données à une autre. Il suffit de sélectionner les objets que vous voulez copier et de les faire glisser vers l'autre session de Crimson contenant la nouvelle base de données. Cette opération est possible pour des pages entières, des groupes de primitives ou pour un seul objet.

### MODIFIER LA TAILLE DES PRIMITIVES

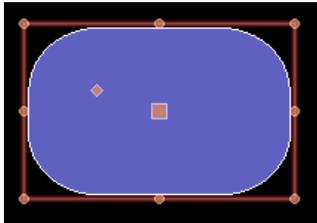
Le redimensionnement des primitives se fait de manière intuitive, en cliquant sur les poignées situées sur les bords de la primitive et en les déplaçant dans la direction voulue. En maintenant la touche **CTRL** enfoncée le redimensionnement agit en même temps sur la hauteur et sur la largeur de la primitive. En maintenant la touche **SHIFT** enfoncée le redimensionnement se fait à partir du centre plutôt qu'à partir d'un bord.

### UTILISATION DES POIGNEES DE MISE EN PAGE

Certaines primitives possèdent des poignées internes que vous pouvez déplacer afin de changer leur apparence. Par exemple le rectangle à bords arrondis illustré ci-dessous possède une poignée unique située dans le coin supérieur gauche. La poignée est représentée par un diamant lorsque la primitive est sélectionnée...



Dans cet exemple, en déplaçant la poignée vous changez le rayon des cercles formant les bords arrondis du rectangle...

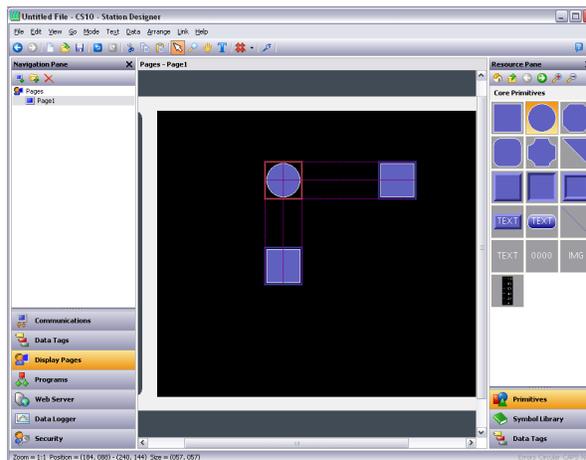


La fonction particulière de chaque poignée dépend de la primitive, mais son utilisation est normalement intuitive.

### ALIGNEMENT INTELLIGENT

Si la fonctionnalité d’alignement intelligent est activée dans le menu Affichage, des lignes directrices d’alignement s’affichent pendant les opérations de déplacement ou de redimensionnement. Elles vous aideront à aligner une primitive avec celles déjà présentes ou avec le centre de l’écran. Avec un peu de pratique, il devient très simple grâce à cette fonctionnalité d’aligner directement les primitives que vous créez, sans avoir besoin de régler l’alignement général de toutes les figures de votre page d’affichage.

Dans l’exemple ci-dessous, un cercle est aligné avec deux carrés...

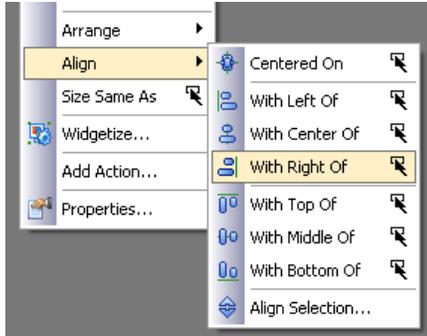


Des lignes d’alignement sont présentes aux deux extrémités des figures et en leur centre, indiquant qu’à la fois les extrémités et les centres sont alignés. Le rectangle rouge est positionné autour de la primitive que vous manipulez, les rectangles bleus sont positionnés autour des primitives vers lesquelles les lignes d’alignement ont été tracées.

### ALIGNEMENT RAPIDE

La fonctionnalité d’alignement rapide de l’éditeur graphique Crimson permet d’aligner des primitives à d’autres primitives sans avoir recours à une boîte de dialogue. Pour utiliser cette

fonctionnalité, il suffit de sélectionner la primitive que vous voulez déplacer et de faire un clic droit pour faire apparaître le menu contextuel. Choisissez le sous-menu Aligner et ensuite sélectionnez les diverses options “Avec...de” indiquées par le symbole rectangle-et-curseur. Le pointeur de la souris change et vous indique que vous devez cliquer sur la primitive sur laquelle vous voulez effectuer l’alignement.



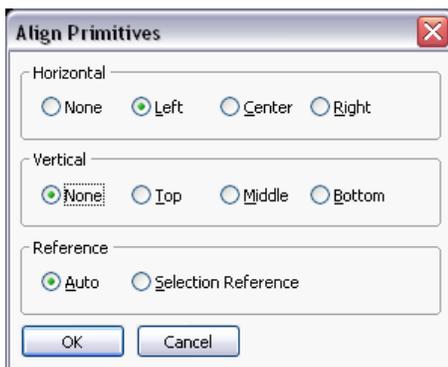
Dès que vous cliquez, l’alignement s’effectuera.

### UTILISATION DE LA GRILLE

Vous pouvez utiliser le bouton Grille de la barre d’outils afin de contrôler le comportement et l’affichage de la grille d’alignement. Cliquer sur la partie gauche du bouton permet d’afficher ou de cacher la grille. Cliquer sur la partie inférieure permet de configurer les opérations pour lesquelles la grille est utilisée. Vous pouvez activer ou désactiver la grille séparément pour les opérations de création, de redimensionnement et de déplacement, ou vous pouvez utiliser les options Tout ou Aucun pour les activer ou désactiver globalement. Vous pouvez aussi contrôler l’utilisation de la grille lorsque vous éditez à l’intérieur de groupes.

### ALIGNEMENT DES PRIMITIVES

Bien que les options d’alignement rapide et d’alignement intelligent examinées ci-dessus vous permettent d’effectuer sans problèmes des opérations d’alignement, il est aussi possible d’utiliser une approche plus courante. Pour cela, sélectionnez un certain nombre de primitives et utilisez la commande Aligner la Sélection dans le menu Arranger. La boîte de dialogue suivante s’affiche...



Utilisez les paramètres Horizontal et Vertical pour indiquer quel type d’alignement vous souhaitez effectuer, le paramètre Référence définissant quant à lui quelle primitive est utilisée comme référence pour les opérations d’alignement. Dans l’exemple ci-dessus, le mode Auto utilisera comme référence la primitive la plus à gauche puisqu’un alignement à gauche est effectué. Les autres modes d’alignement fonctionnent de la même manière. Le mode alternatif

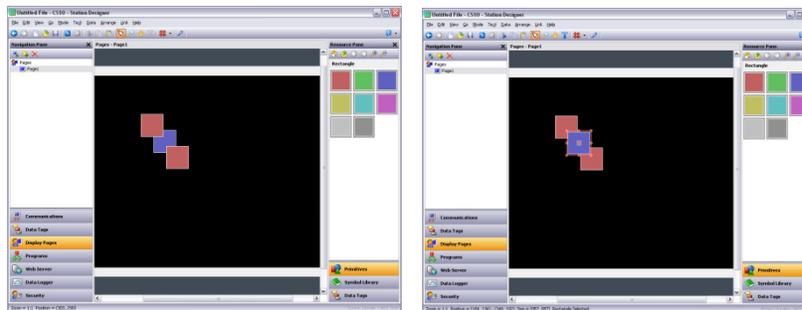
utilise le premier objet sélectionné comme référence. Cet objet est identifié par la présence d’un carré plus grand en son centre.

### ESPACEMENT DES PRIMITIVES

Si vous voulez espacer de manière équidistante un certain nombre de primitives sur la page, vous pouvez utiliser les commandes Espacement vertical équidistant ou Espacement horizontal équidistant du menu Arranger. Ces commandes agissent sur les primitives sélectionnées en essayant de redistribuer l’espace libre entre les objets de manière à obtenir un espacement équidistant. Les deux primitives des extrémités ne bougeront pas de position. Notez que cette commande peut échouer si un ensemble inadéquat de primitives est choisi, et que l’espacement obtenu peut ne pas être parfaitement équidistant si l’espace disponible est insuffisant.

### REORGANISATION DES PRIMITIVES

Les primitives d’une page d’affichage sont stockées selon un ordre appelé ordre-z. Cet ordre définit la séquence suivant laquelle sont dessinées les primitives, et indique donc si une primitive donnée apparaît devant ou derrière une autre primitive. Dans le premier exemple ci-dessous, le carré bleu est affiché derrière les carrés rouges, elle est donc placée après les autres dans l’ordre-z. Dans le deuxième exemple, cette primitive a été déplacée avant les autres dans l’ordre-z, elle apparaît devant les autres figures sur la page.



Pour modifier la position des objets dans l’ordre-z, sélectionnez-les et utilisez les diverses commandes du menu Arranger. Les commandes Déplacer vers l’avant et Déplacer vers l’arrière déplacent l’objet sélectionné d’un cran dans la direction indiquée, alors que les commandes Déplacer vers le début et Déplacer vers la fin déplacent l’objet sélectionné à l’une ou l’autre extrémité de l’ordre-z. Si vous disposez d’une souris à molette, vous pouvez déplacer l’objet sélectionné en faisant rouler la molette tout en maintenant la touche **CTRL** enfoncée. Faire rouler la molette vers le haut déplace l’objet sélectionné vers la fin de l’ordre-z, la faire rouler vers le bas le déplace vers le début.

### DUPLICATION DES PRIMITIVES

Vous pouvez utiliser la combinaison de touches **CTRL + D** ou la commande Duplication intelligente du menu Edition pour faire une copie de la primitive active de manière à ce que ses propriétés soient contrôlées par les données de l’objet suivant. Ici la définition de « suivant » dépend du type exact de donnée, l’éditeur Crimson pouvant sélectionner le registre suivant dans le périphérique comms, l’élément suivant dans un vecteur ou le tag suivant dans une séquence. Par exemple, l’utilisation répétée de la commande Duplication intelligente avec un bouton pointant vers `Vecteur[0]` produira une séquence de boutons pointant vers `Vecteur[1]`, `Vecteur[2]`, etc. jusqu’au remplissage total de l’écran.

## EDITION DE PRIMITIVES MULTIPLES

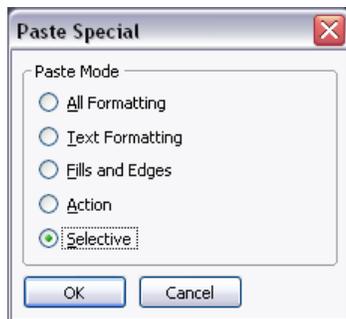
Il peut vous arriver de vouloir éditer les propriétés de plusieurs primitives. L'éditeur Crimson vous offre cette possibilité : vous devez éditer une primitive et ensuite configurer les propriétés d'un certain nombre d'autres primitives pour qu'elles soient identiques aux propriétés de la première primitive. L'éditeur Crimson vous offre deux méthodes reposant sur le même mécanisme sous-jacent pour le faire.

### UTILISATION DE LA COMMANDE COPIER DEPUIS

Vous pouvez utiliser la commande Copier depuis pour copier les propriétés choisies d'une certaine primitive vers une ou plusieurs autres primitives. Pour utiliser cette commande, sélectionner les primitives cibles et faites un clic droit pour pouvoir accéder au menu contextuel associé. Sélectionnez une des commandes Copier depuis. Le curseur va changer de forme afin de vous permettre de sélectionner la primitive à partir de laquelle vous effectuerez la copie. En fonction de la commande que vous avez choisie, une ou plusieurs propriétés de la primitive source seront appliquées aux primitives cibles.

### UTILISATION DE LA COMMANDE COLLAGE SPECIAL

Vous pouvez utiliser la commande Collage spécial pour obtenir un résultat identique par une méthode qui permet en plus de copier les propriétés d'une base de données à une autre, ou entre plusieurs sessions de Crimson. Sélectionnez tout d'abord la primitive source et utilisez la commande Copier pour la déplacer vers le Presse-papiers. Sélectionnez ensuite les primitives cibles, faites un clic droit sur la sélection et utilisez la commande Collage spécial. La boîte de dialogue suivante apparaîtra...



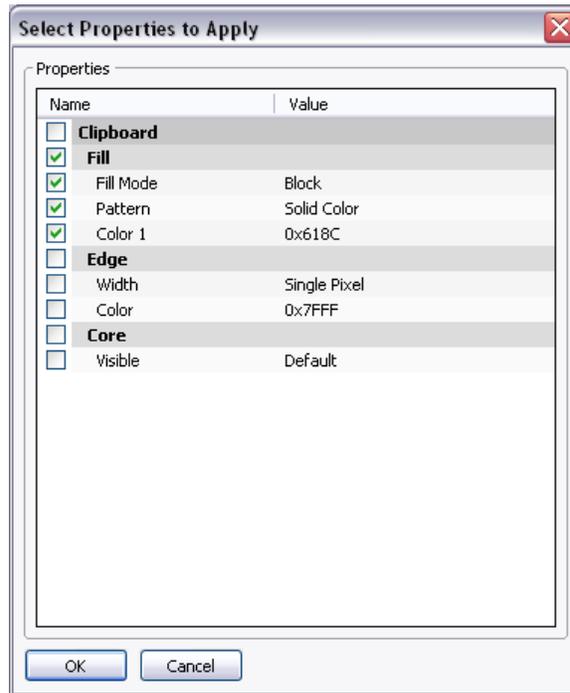
Les propriétés sélectionnées de la primitive source seront appliquées aux primitives cibles.

### SELECTIONS DE PROPRIETE

Les deux méthodes décrites ci-dessus vous permettent de définir quelles propriétés vous voulez copier...

- *Formatage entier* copie tout, exception faite de tout texte, donnée ou action.
- *Formatage du texte* copie la police, l'alignement et les marges du texte ou des objets.
- *Remplissages et bordures* copie les attributs remplissage et bordure de l'onglet Figure.
- *Action* copie toute action assignée à la primitive.

De plus, vous pouvez utiliser l’option *Sélection* pour sélectionner les propriétés qui seront copiées...



La liste contient une présentation hiérarchisée des propriétés de la primitive source, organisée en fonction de la mise en page utilisée lors de l’édition de la primitive. Cette liste affiche les valeurs assignées à chaque propriété. Chaque propriété ou groupe de propriétés peut être sélectionné ou désélectionné en cochant la case associée. Seules les propriétés cochées seront appliquées, ce qui vous donne un niveau de contrôle détaillé de ce qui sera copié d’une primitive à une autre.

### PASSAGE VERS D’AUTRES ELEMENTS

Si une primitive fait référence à des tags, à des pages d’affichage ou à d’autres objets, un sous-menu Aller vers apparaîtra dans le menu contextuel. Sélectionnez ce menu afin de voir une liste des objets référencés. Sélectionnez un de ces objets pour aller directement à la section correspondante de la base de données. L’exemple suivant montre une primitive qui fait référence à deux tags...



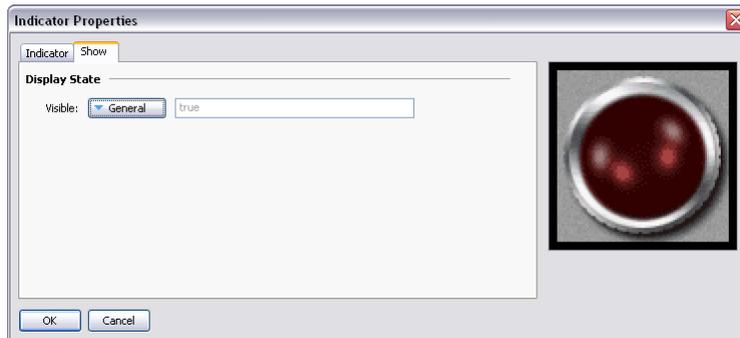
Après avoir effectué les modifications souhaitées au tag, vous pouvez utiliser le bouton Retour de la barre d’outils ou la combinaison de touches **ALT + GAUCHE** pour revenir à la page d’affichage en cours d’édition. Notez que la sélection est conservée durant la navigation, ce qui rend plus simple de voir ou d’éditer l’objet référencé et de reprendre par la suite le processus de création de l’affichage.

## PROPRIETES DES PRIMITIVES

Vous pouvez éditer les propriétés d'une primitive en faisant un double clic sur la primitive ou en utilisant la commande Propriétés du menu contextuel de la primitive. Vous pouvez aussi sélectionner la primitive et faire la combinaison de touches **ALT + ENTER**. La fenêtre de propriétés d'une primitive contient plusieurs onglets, certains ne s'affichant que lorsque des objets supplémentaires — tels que du texte, des données ou une action — ont été ajoutés à la primitive. La fenêtre Propriétés affiche un aperçu en temps réel de la primitive active, ce qui vous permet de voir les effets des modifications avant de les appliquer.

### AFFICHER OU CACHER LES PRIMITIVES

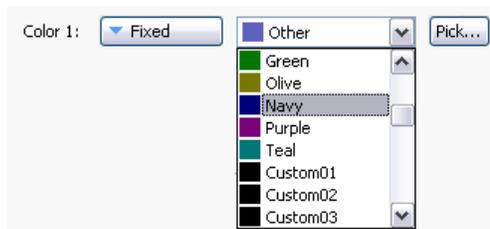
Toutes les primitives possèdent un onglet Afficher dans leur fenêtre de propriétés...



Vous pouvez assigner un nombre entier à la propriété *Visible* pour montrer ou masquer la primitive associée lors de l'exécution du programme. La valeur zéro a pour effet de masquer la primitive, une valeur non nulle permet l'affichage de la primitive. Par défaut, toutes les primitives sont visibles.

### DEFINIR LES COULEURS DES PRIMITIVES

L'édition des couleurs dans une primitive s'effectue comme dans l'exemple ci-dessous...



La présentation des propriétés de couleur est donnée sous forme d'un menu déroulant à boutons, d'une liste déroulante et d'un bouton Choisir. Le menu déroulant permet de choisir le mode d'animation des couleurs parmi les modes suivants...

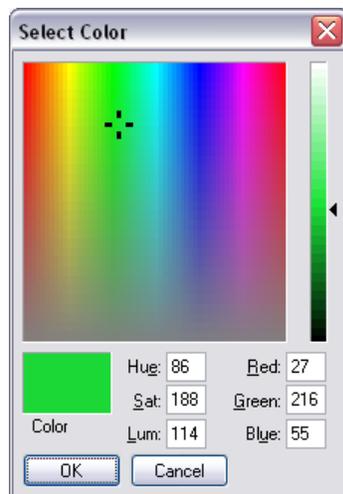
- En mode *Fixe*, la couleur ne change pas, elle est choisie soit à partir de la liste déroulante soit en cliquant sur le bouton Choisir qui ouvrira la fenêtre de sélection des couleurs.
- En mode *Tag Texte*, l'animation couleur est définie de manière à ce que la couleur corresponde à celle du premier plan définie par un tag spécifique. Ce tag peut être sélectionné en cliquant sur le bouton Choisir.

- En mode *Tag Arrière*, l’animation couleur est définie de manière à ce que la couleur corresponde à celle de l’arrière plan définie par un tag spécifique. Ce tag peut être sélectionné en cliquant sur le bouton Choisir.
- En mode *Clignotant*, l’animation couleur oscille entre deux couleurs à une fréquence donnée, une autre couleur étant affichée si le clignotement est désactivé.
- En mode *2-États*, l’animation couleur change entre deux couleurs en fonction de la valeur d’un tag ou d’une autre donnée.
- En mode *4-États*, l’animation couleur change entre quatre couleurs en fonction de la valeur de deux tags ou d’autres données.
- En mode *Mélange*, l’animation couleur est définie de manière à passer progressivement d’une couleur à une autre en fonction de la valeur d’un tag ou d’une autre donnée par rapport à des valeurs minimales et maximales particulières.
- En mode *Expression*, vous pouvez saisir une expression numérique qui sera utilisée pour définir la couleur d’affichage. Voir ci-dessous pour plus de détails.
- En mode *Complexe*, vous pouvez écrire un programme local dont le résultat est un nombre entier pour définir la couleur d’affichage. Voir ci-dessous pour plus de détails.

Le menu-déroulant contient les couleurs suivantes...

- La palette des seize couleurs standards VGA.
- Trente-deux niveaux de gris allant du noir au blanc.
- Toute autre couleur utilisée dans la base de donnée, jusqu’à une limite de vingt-quatre.

Vous pouvez utiliser l’option Autres Couleurs située au bas de la liste pour faire apparaître la boîte de dialogue de sélection des couleurs...

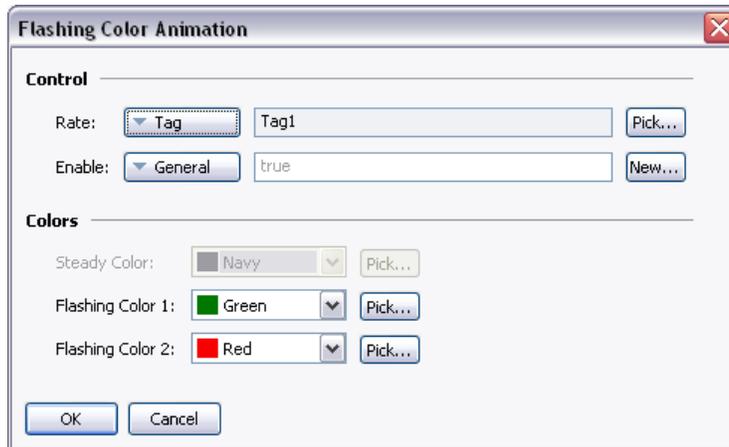


Cette fenêtre offre diverses possibilités de définition d’une couleur. Vous pouvez en choisir une dans la palette des couleurs, dans la fenêtre « arc-en-ciel » ou bien saisir les paramètres HSL ou RGB explicites. Si la couleur choisie n’a pas été utilisée précédemment dans la base

de données et si elle n'est pas identique à l'une des couleurs ou l'un des niveaux de gris standards, elle sera ajoutée aux couleurs personnalisées affichées dans le menu déroulant.

### DEFINIR LES COULEURS DE CLIGNOTEMENT

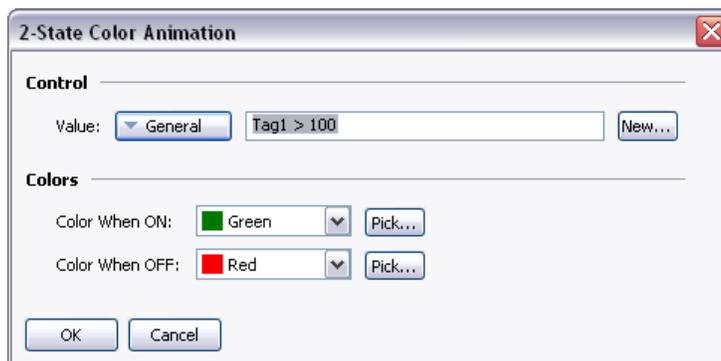
Le mode clignotant de couleurs est défini à travers la boîte de dialogue suivante...



- La propriété *Fréquence* définit la fréquence du clignotement. La valeur 1 donne une fréquence de 1 Hz, chaque couleur étant affichée durant 500 ms. Il n'est pas recommandé d'utiliser des fréquences supérieures à 4 Hz car la mise à jour de la fréquence de l'affichage du périphérique cible peut produire des effets de « résonance ».
- La propriété *Activer* définit une expression optionnelle qui peut être utilisée pour activer ou désactiver le clignotement. Lorsque le clignotement est désactivé, une couleur fixe est affichée.
- La propriété *Couleurs* vous permet de définir les couleurs utilisées.

### DEFINIR LES COULEURS 2-ÉTATS

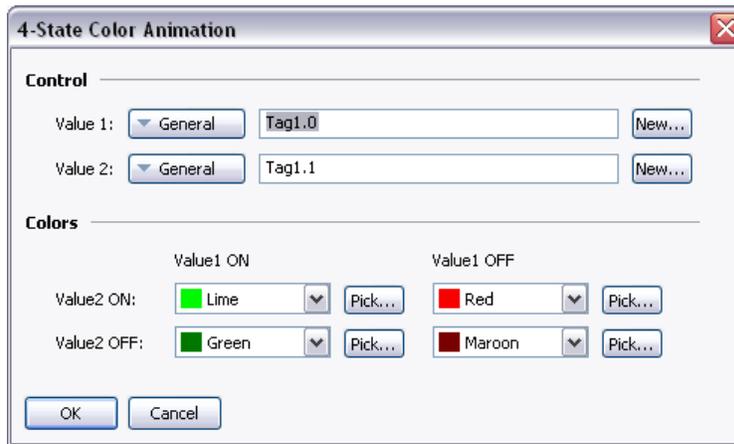
Le mode à deux états est défini à travers la boîte de dialogue suivante...



- La propriété *Valeur* est utilisée pour sélectionner la couleur qui sera affichée.
- La propriété *Couleurs* vous permet de définir les couleurs utilisées.

## DEFINIR LES COULEURS 4-ETATS

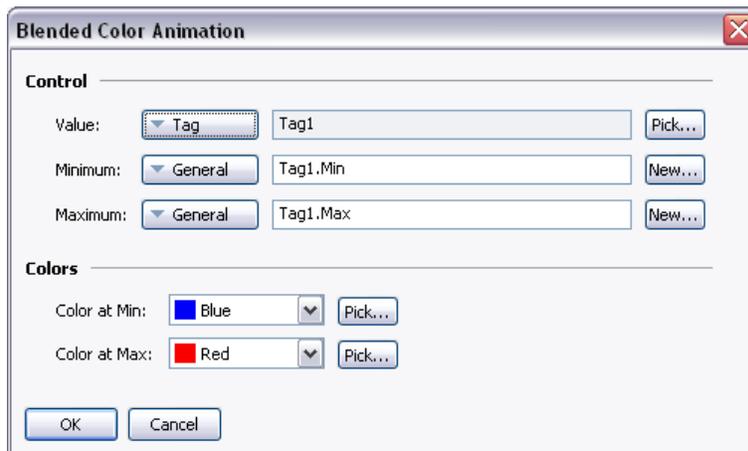
Le mode à quatre couleurs est défini à travers la boîte de dialogue suivante...



- La propriété *Valeur* est utilisée pour sélectionner la couleur qui sera affichée.
- La propriété *Couleurs* vous permet de définir les couleurs utilisées.

## DEFINIR LE MODE MELANGE

Le mode mélange de couleurs est défini à travers la boîte de dialogue suivante...



- Les propriétés *Valeur*, *Minimum* et *Maximum* sont utilisées pour définir les couleurs qui seront affichées. Dans l'exemple ci-dessus, la couleur affichée est le bleu si le tag est inférieur ou égal à sa valeur minimale, le rouge si le tag est supérieur ou égal à sa valeur maximale, et elle passe progressivement du bleu au rouge lorsque la valeur du tag évolue entre ses limites.
- La propriété *Couleurs* vous permet de définir les couleurs qui seront utilisées.

## DEFINIR LES EXPRESSIONS DE COULEURS

Comme il a été mentionné ci-dessus, les propriétés des couleurs peuvent être définies à travers des valeurs entières ou des programmes locaux dont les résultats sont des valeurs entières. Ces mécanismes ne sont utilisés que lorsque les méthodes standards d'animation en

couleur sont insuffisantes. Comme il ne vous arrivera pas souvent d'utiliser ces fonctionnalités, n'hésitez pas à passer cette section si vous la trouvez trop complexe.

Crimson utilise un encodage à 15 bits pour la valeur des couleurs, les cinq premiers bits représentant le rouge, les cinq suivants représentant le vert et les cinq derniers le bleu. Vous pouvez modifier les valeurs des couleurs comme vous le feriez pour une valeur entière.

#### COMPOSER DES COULEURS

Vous pouvez utiliser la fonction `ColGetRGB(r,g,b)` pour créer une valeur de couleur à partir de ses composantes rouge, verte et bleue. Même si Crimson utilise un encodage à 15 bits pour la valeur des couleurs, contenant trois valeurs à 5 bits, les arguments transmis à la fonction sont réduits d'un facteur 8 et doivent donc être compris entre 0 et 255. `ColGetRGB(128, 0, 64)` donne comme résultat une valeur rouge de 16, pas de composante verte et une valeur bleue de 8.

#### SEPARER DES COULEURS

Vous pouvez utiliser les fonctions `ColGetRed(rgb)`, `ColGetGreen(rgb)` et `ColGetBlue(rgb)` pour accéder aux composantes individuelles d'une couleur donnée. Pour respecter la convention adoptée par `ColGetRGB()`, les valeurs de ces fonctions sont comprises entre 0 et 255.

#### CHOIX DE COULEURS

Vous pouvez utiliser la fonction `ColPick2()` pour choisir entre deux couleurs en fonction de la valeur d'une expression. Par exemple, l'expression `ColPick2(Flag1, Col1, Col2)` donne comme résultat `Col1` si `Flag1` est différent de zéro, ou `Col2` si `Flag1` vaut zéro. Les deux arguments de couleur peuvent être remplacés par des appels à la fonction `ColGetRGB()` si nécessaire.

#### MELANGE DE COULEURS

Vous pouvez utiliser la fonction `ColBlend()` pour construire une couleur personnalisée à partir de deux autres couleurs. Par exemple, l'expression `ColBlend(Data, 0, 100, Col1, Col2)` donne comme résultat `Col1` si `Data` vaut 0 et `Col2` si `Data` vaut 100. Des valeurs intermédiaires produiront les mélanges appropriés de ces deux couleurs, avec une transition progressive d'une couleur à l'autre. Ici aussi, les arguments des couleurs peuvent être remplacés par des appels à la fonction `ColGetRGB()`.

#### REPONSE A UNE STIMULATION

La variable système `IsPressed` a la valeur `True` lorsque la primitive active a été stimulée, la valeur `False` dans le cas contraire. Cette variable peut être utilisée avec les fonctions de sélection de couleur afin d'animer une primitive en fonction de sa stimulation. Notez que les primitives ne permettent pas la stimulation à moins qu'une action lui ait été assignée ou qu'elle possède une action inhérente.

#### DEFINIR DES JAUGES DE NIVEAU

Beaucoup de primitives géométriques possèdent une option de type « jauge de niveau » dans laquelle la figure est remplie jusqu'à un certain niveau en fonction du contenu d'un tag. Vous pouvez utiliser cette fonctionnalité dans des histogrammes ou pour remplir des formes plus complexes.

L’exemple ci-dessous montre une étoile à six branches avec un niveau de jauge rempli à 60%...



Les jauges de niveau sont définies en utilisant les propriétés Style de remplissage d’une primitive...

**Fill Behavior**

Fill Mode:  ▼

Value:  Tag1

Minimum:  Tag1.Min

Maximum:  Tag1.Max

- La propriété *Mode remplissage* indique si une jauge de niveau doit être dessinée, et dans quelle direction le remplissage doit s’effectuer. Les remplissages peuvent se faire à partir de tous les bords de la primitive permettant ainsi la création d’animations complexes. En mode Bloqué, la figure est remplie avec un motif unique et la jauge de niveau est désactivée.
- La propriété *Valeur* sélectionne la valeur qui sera utilisée pour calculer le niveau de la jauge. Si un tag est choisi, les limites Minimum et Maximum seront automatiquement définies comme étant celles de la donnée d’entrée en utilisant la propriété de syntaxe d’expression de ce tag. La propriété *Valeur* peut être un entier ou un nombre à virgule flottante. Les calculs du niveau de jauge sont toujours effectués en virgule flottante.
- Les valeurs *Minimum* et *Maximum* fixent les limites de la propriété Valeur qui seront utilisées pour le calcul du niveau de la jauge.

## DEFINIR DES FORMATS DE JAUGE

La propriété Styles de Remplissage d’une primitive indique la manière de remplir l’intérieur d’une primitive...

**Fill Format**

Pattern:  Graduated Fill 2 ▼

Color 1:   ▼

Color 2:   ▼

Color 3:   ▼

- La propriété *Modèle* sélectionne parmi les différents styles de remplissage. L’option usuelle est Couleur Unie, mais vous pouvez choisir parmi différents styles hachurés ou à tramage. Des remplissages gradués sont aussi disponibles....

STYLE	DESCRIPTION
Remplissage gradué 1	Couleur 1 en haut et en bas de la primitive, passant

STYLE	DESCRIPTION
	progressivement vers la Couleur 2 au centre.
Remplissage gradué 2	Couleur 1 en haut de la primitive, passant progressivement vers la Couleur 2 en bas.
Remplissage gradué 3	Couleur 1 à gauche et à droite de la primitive, passant progressivement vers la Couleur 2 au centre.
Remplissage gradué 4	Couleur 1 à gauche de la primitive, passant progressivement vers la Couleur 2 à droite.

- La propriété *Couleur 1* définit la première couleur utilisée lors du remplissage.
- La propriété *Couleur 2* définit une deuxième couleur optionnelle utilisée lors du remplissage.
- La propriété *Couleur 3* définit la couleur de l'arrière plan de la jauge de niveau. Il n'est pas nécessaire de la définir si le mode de remplissage bloqué est activé. Cette propriété peut ne pas être affichée si la primitive active ne possède pas l'option jauge de niveau.

### DEFINIR LES STYLES DES BORDURES

Les propriétés Styles de bordures d'une primitive définissent la manière de dessiner les bordures de la primitive...



- La propriété *Epaisseur* indique l'épaisseur de la bordure. Actuellement Crimson ne permet que des valeurs impaires pour l'épaisseur de la bordure, jusqu'à neuf pixels.
- La propriété *Couleur* définit la couleur de la bordure.
- La propriété *Coins*, qui n'existe que pour les rectangles, indique si les coins de la bordure doivent être dessinés carrés ou arrondis. Toutes les autres primitives utilisent par défaut des coins arrondis.

### UTILISATION DES GROUPES UTILISATION DE GROUPES

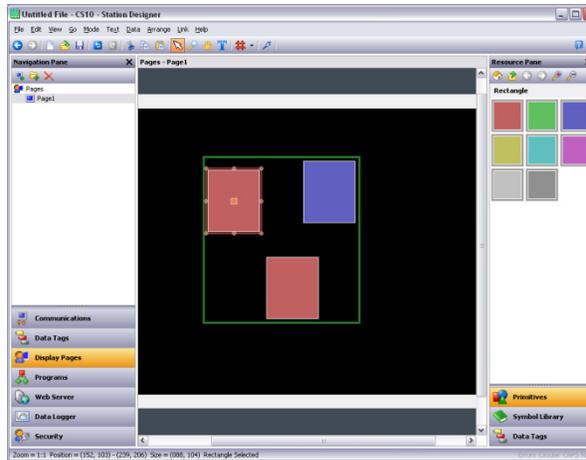
Un groupe est constitué d'un ensemble de primitives traitées comme un objet unique.

#### FAIRE ET DEFAIRE DES GROUPES

Si vous voulez regrouper des primitives, vous pouvez les sélectionner comme décrit plus haut et utiliser la commande Grouper du menu Arranger. La même opération peut s'effectuer avec la combinaison de touches **CTRL + G**. Une fois le groupe créé, vous pouvez le déplacer, le redimensionner et le copier de la même manière que pour un objet unique. Vous pouvez défaire un groupe en ses constituants que sont les primitives en sélectionnant le groupe et en cliquant sur la commande Dégroupier ou avec la combinaison de touches **CTRL + U**. Notez que des groupes peuvent être composés de primitives et d'autres groupes, et que les groupes peuvent être imbriqués les uns dans les autres dans une limite raisonnable.

## EDITION A L’INTERIEUR DES GROUPES

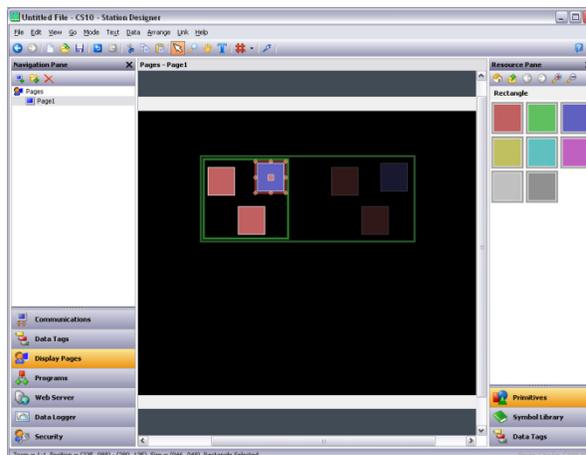
Une fois le groupe créé, il peut vous arriver de vouloir éditer ses éléments sans devoir le défaire. Ceci est d’autant plus utile si vous avez créé des groupes imbriqués, car alors le processus de regroupement peut s’avérer particulièrement difficile. Pour éditer à l’intérieur d’un groupe, sélectionnez tout d’abord ce groupe et cliquez ensuite sur un élément du groupe (ne cliquez pas sur la poignée située au centre du groupe car celle-ci est utilisée pour déplacer ou sélectionner le groupe dans son ensemble). Une fois l’élément du groupe sélectionné, l’éditeur Crimson va passer en mode édition de groupe, comme indiqué ci-dessous...



Observez le rectangle vert affiché autour du groupe en cours d’édition. L’édition à l’intérieur d’un groupe s’effectue de la même manière que l’édition d’une page, à l’exception faite que les objets ne peuvent pas être déplacés à l’extérieur du groupe. Ils peuvent être copiés, collés, redimensionnés et supprimés. En fait, toutes les opérations usuelles peuvent être effectuées. Vous pouvez même faire glisser et déposer de nouveaux objets depuis le panneau de ressources vers le groupe. Pour quitter le mode édition de groupe, il suffit de cliquer sur une zone à l’extérieur du groupe ou appuyez sur la touche Esc.

## EDITION DE GROUPES IMBRIQUES

L’éditeur Crimson vous permet aussi d’éditer à l’intérieur de groupes qui sont eux-mêmes à l’intérieur de groupes...

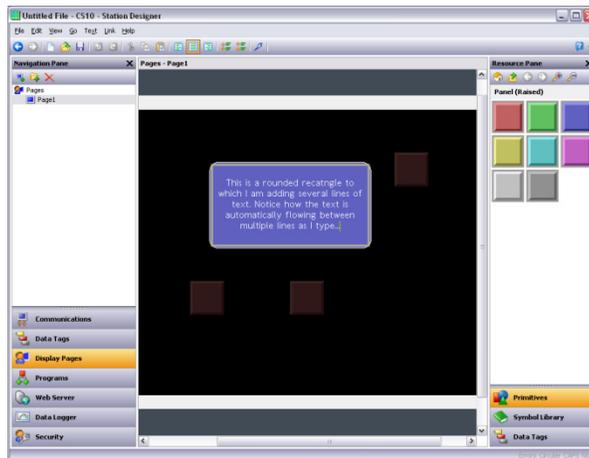


Pour activer cette fonctionnalité, commencez l’édition dans le groupe externe et cliquez sur un élément du groupe imbriqué dans le premier. Observez dans l’exemple ci-dessous

comment la hiérarchie dans un groupe est représentée par une série de rectangles. Observez aussi que les objets à l'extérieur des groupes sont affichés avec des couleurs pour mieux visualiser les limites des groupes. Appuyez sur la touche **Esc** pour quitter l'édition dans un groupe imbriqué, chaque clic faisant quitter un cran dans la hiérarchie du groupe.

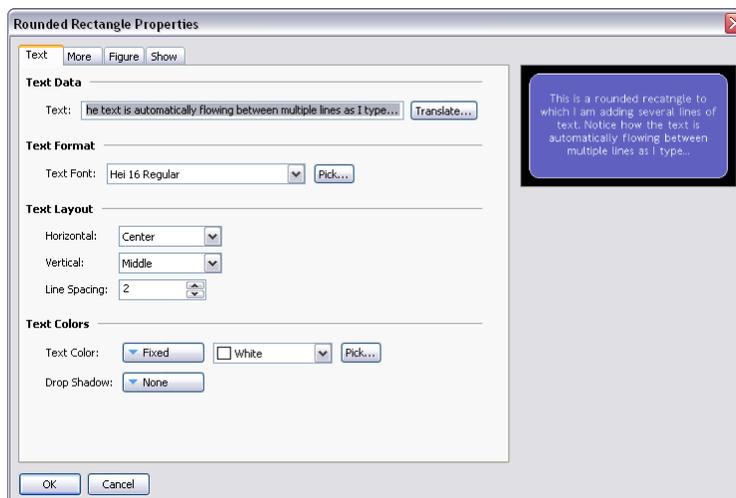
## AJOUTER DU TEXTE A DES PRIMITIVES

La plupart des primitives dans l'éditeur Crimson permettent l'ajout de texte. Pour ajouter du texte à une primitive, il suffit de sélectionner la primitive, d'appuyer sur la touche **F2** et de saisir le texte. Vous pouvez également faire un clic droit sur la primitive et sélectionner la commande Ajouter du Texte du menu qui s'affiche. L'exemple ci-dessous montre l'ajout de texte dans un rectangle à bords arrondis...



Observez tout d'abord que le rectangle de contour de la primitive est affiché en jaune et que toutes les autres primitives de la page sont grisées. Observez également comment l'éditeur répartit automatiquement le texte entre les lignes. Si vous essayez de redimensionner une primitive contenant du texte, vous verrez que l'éditeur Crimson répartit le texte de manière à ce qu'il s'adapte au mieux à la nouvelle forme.

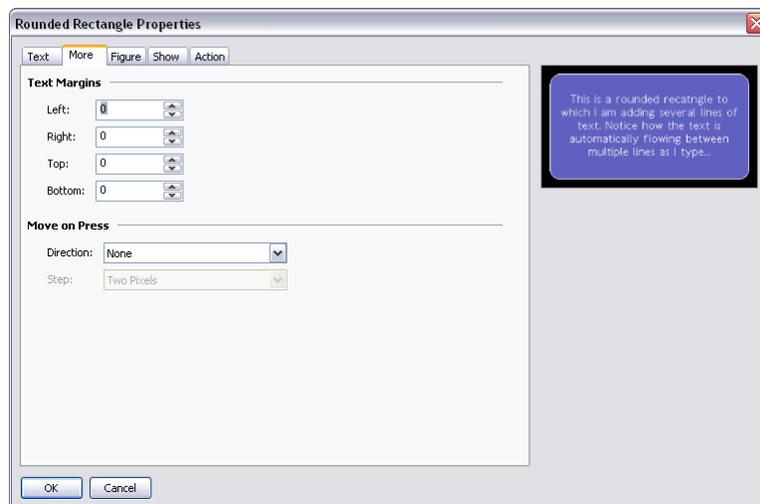
En mode édition de texte, des commandes permettant de modifier l'alignement du texte et d'augmenter ou de réduire l'espacement entre les lignes apparaissent dans la barre d'outils. Vous pouvez éditer d'autres propriétés du texte soit en sélectionnant les Propriétés Texte du menu contextuel de la primitive, soit en appuyant sur **ALT + ENTER** en mode édition...



## PROPRIETES DU TEXTE

- La propriété *Texte* contient le texte qui sera affiché. Le caractère barre verticale est utilisé pour coder les retours à la ligne imposés. Comme ce champ est une chaîne de caractères qui peut être traduite, des versions dans plusieurs langues peuvent être éditées. Cela implique également que l’on peut assigner une expression à la propriété, ce qui permet de modifier son contenu de manière dynamique. Crimson supporte une recomposition dynamique complète offrant des options de présentation complexe et attractive.
- La propriété *Police du texte* permet de sélectionner la police. La nouvelle police par défaut utilisée dans l’éditeur Crimson est Hei — une police Unicode compatible avec le chinois simplifié et plusieurs autres langues. Vous pouvez utiliser le bouton Choisir pour faire apparaître la boîte de dialogue de la police, permettant à toute police installée dans votre système d’être mise sous une forme qui puisse être utilisée par le périphérique cible. Notez qu’il est de votre responsabilité de s’assurer que vous avez la licence requise pour ce type d’utilisation.
- La propriété *Horizontal* définit l’alignement horizontal du texte.
- La propriété *Vertical* définit l’alignement vertical du texte.
- La propriété *Interligne* définit un espacement supplémentaire entre les lignes donné en pixels.
- La propriété *Couleur du texte* sélectionne la couleur du texte.
- La propriété *Ombre portée* est utilisée pour activer un effet d’ombre optionnel qui s’applique sur la partie droite et le bas du texte. Cet effet est utile pour faire ressortir le texte de l’arrière-plan, en particulier lorsque ce dernier est une image contenant plusieurs couleurs.

## MORE PROPERTIES AUTRES PROPRIETES



- Les propriétés *Marges du texte* sont utilisées pour contrôler les marges autour du texte par rapport à la boîte de contour du texte définie par la primitive. Elles sont utiles pour que visuellement le texte apparaisse mieux centré lorsque vous

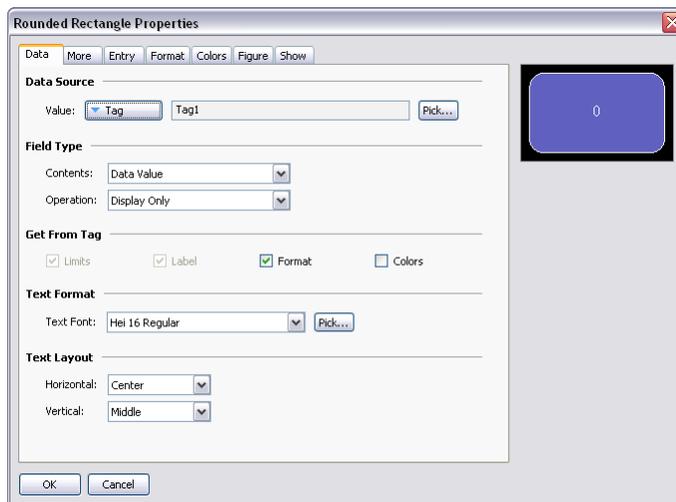
utilisez des polices ayant beaucoup d'espacement au dessus ou en-dessous des caractères, aussi bien les diacritiques que les jambages.

- La propriété *Direction* définit la direction dans laquelle le texte sera déplacé lorsque la primitive associée est actionnée. Cette propriété n'est activée que si une action a été assignée à une primitive, ou lorsque la primitive est semblable à un bouton qui possède une action inhérente associée. Cette option est utile pour créer des boutons personnalisés devant fournir un retour d'information lorsqu'il est actionné.
- La propriété *Cran* indique de combien le texte doit être déplacé lorsque la primitive est actionnée. Vous pouvez choisir un cran de un à trois pixels en fonction de l'effet désiré.

## AJOUTER DES DONNEES AUX PRIMITIVES

Les primitives qui possèdent l'option ajout de texte permettent aussi l'affichage de données en temps réel. Il est possible de les configurer pour permettre la saisie de données. Pour ajouter des données à une primitive, faites un clic droit sur la primitive et dans le menu qui apparaît sélectionnez la commande Ajout de données. Vous pouvez également sélectionner la primitive et appuyez sur la touche F3. La boîte de dialogue des propriétés de la primitive s'affichera, avec des onglets supplémentaires disponibles pour définir la donnée requise et son comportement.

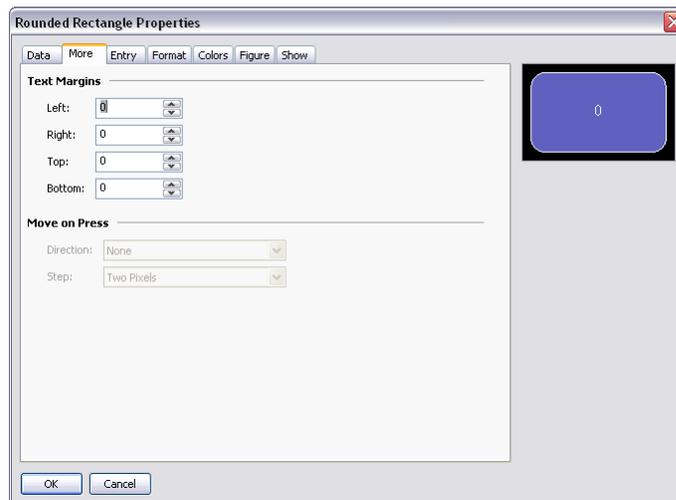
### PROPRIETES DONNEES



- La propriété *Valeur* définit la valeur de la donnée qui s'affichera.
- La propriété *Contenu* indique si le champ doit afficher la valeur de la donnée, la valeur de la donnée ainsi que le label associé ou seulement le label.
- La propriété *Opération* indique si le champ doit seulement afficher la valeur ou s'il doit aussi fournir la fonctionnalité de la donnée saisie. La saisie de donnée n'est évidemment possible que si la valeur de la donnée sélectionnée n'est pas protégée en écriture.

- Les propriétés *Prendre depuis* indique si certaines propriétés du champ sont définies localement ou si elles pointent vers les propriétés du tag affiché. Les options ne sont disponibles que si un tag est spécifié dans Valeur.
- La propriété *Police du texte* permet de sélectionner la police. La nouvelle police par défaut utilisée dans l’éditeur Crimson est Hei — une police Unicode compatible avec le chinois simplifié et la plupart des autres langues. Vous pouvez utiliser le bouton Choisir pour faire apparaître la boîte de dialogue de la police, permettant à toute police installée dans votre système d’être mise sous une forme qui puisse être utilisée par le périphérique cible. Notez qu’il est de votre responsabilité de s’assurer que vous avez la licence requise pour ce type d’utilisation.
- La propriété *Horizontal* définit l’alignement horizontal du texte.
- La propriété *Vertical* définit l’alignement vertical du texte.

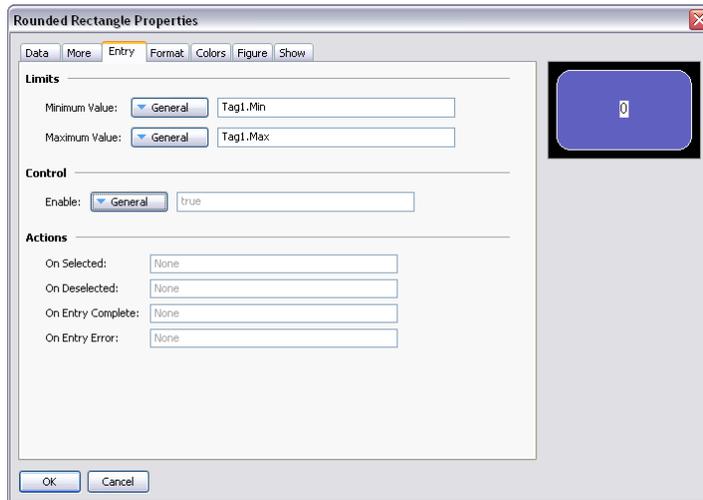
#### AUTRES PROPRIETES



- Les propriétés *Marges du texte* permettent de contrôler les marges autour du texte par rapport à la boîte de contour du texte définie par la primitive. Elles sont utiles pour que visuellement le texte apparaisse mieux centré lorsque vous utilisez des polices ayant beaucoup d’espace au dessus ou en-dessous des caractères, aussi bien les diacritiques que les jambages.
- La propriété *Direction* définit la direction dans laquelle le texte sera déplacé lorsque la primitive associée est actionnée. Cette propriété n’est activée que si une action a été assignée à une primitive, ou lorsque la primitive est semblable à un bouton qui possède une action inhérente associée. Cette option est utile pour créer des boutons personnalisés devant fournir un retour d’information lorsqu’il est actionné.
- La propriété *Cran* indique de combien le texte doit être déplacé lorsque la primitive est actionnée. Vous pouvez choisir un cran de un à trois pixels en fonction de l’effet désiré.

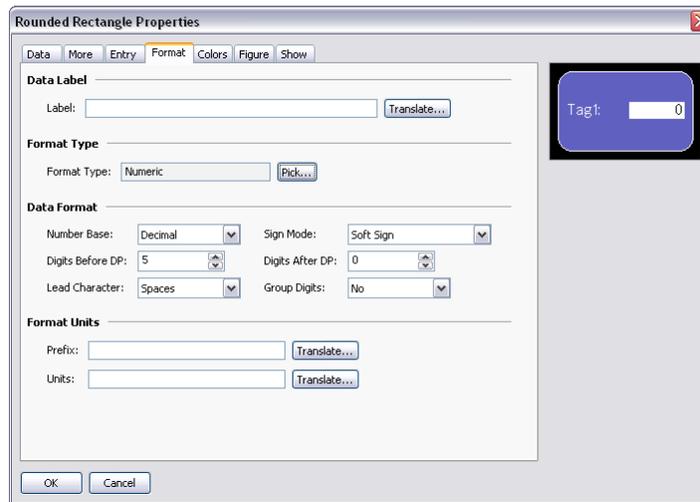
## PROPRIETES ENTREE

Ces propriétés ne sont disponibles que si la saisie de donnée est activée ...



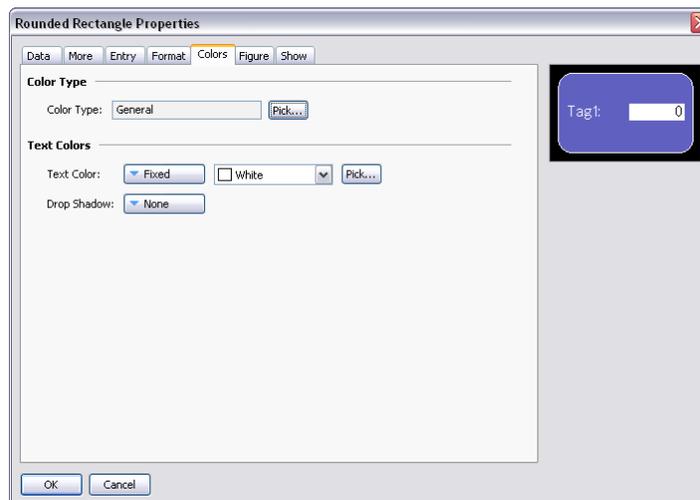
- Les propriétés *Valeur maximale* et *Valeur minimale* définissent les limites de la donnée entrée. Elles ne sont pas disponibles si le champ a été configuré pour obtenir les limites de la donnée entrée à partir du tag de contrôle. Il n'est pas possible de fixer ces paramètres pour tout type de format, en particulier pour ceux dont les limites sont définies de manière implicite.
- La propriété *Activer* permet de définir une expression qui active ou désactive la saisie de donnée. Les champs dont la saisie de donnée a été désactivée se comportent comme des champs d'affichage uniquement
- La propriété *Sur sélectionné* définit une action qui sera exécutée lorsque l'utilisateur appuie sur la donnée entrée du champ, juste avant le début de la saisie de donnée.
- La propriété *Sur désélectionné* définit une action qui sera exécutée lorsque la saisie de donnée se termine, à la suite soit de l'écriture d'une valeur, d'un changement de page ou de l'activation d'un bouton par l'utilisateur qui annule le processus de saisie de données.
- La propriété *Sur entrée terminée* définit une action qui sera exécutée lorsque la saisie de donnée a été achevée avec succès.
- La propriété *Sur erreur entrée* définit une action qui sera exécutée lorsque l'utilisateur entre une valeur invalide.

## PROPRIETES FORMAT



- La propriété *Label* définit le label qui sera appliqué à ce champ. Cette propriété peut ne pas être disponible si le label n’est pas affichable, ou si le champ est configuré de manière à obtenir son label à partir du tag de contrôle.
- Le champ *Type de format* définit le format qui sera utilisé lors de l’affichage ou éventuellement lors de l’édition de la valeur de la donnée. Cette sélection peut ne pas être disponible si le format est obtenu à partir du tag de contrôle.
- D’autres propriétés dépendent du format de la donnée sélectionnée. Reportez-vous au chapitre Utilisation des formats pour le détail des propriétés de chaque format.

## PROPRIETES COULEUR

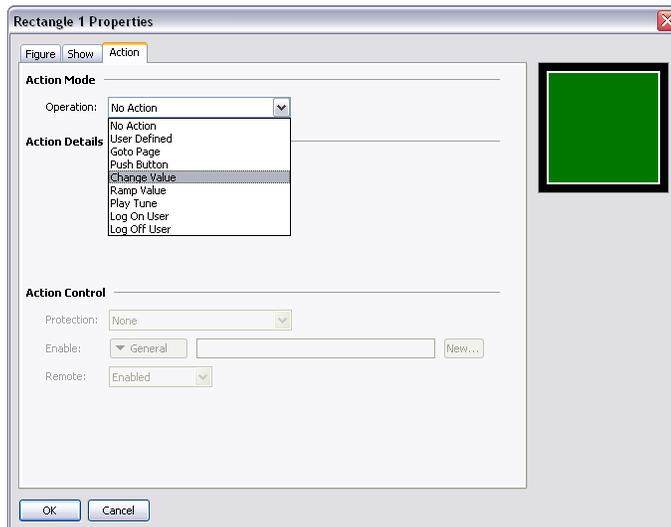


- Le champ *Type de couleur* définit l’animation en couleur qui sera utilisée lors de l’affichage de la valeur de la donnée. Cette sélection peut ne pas être disponible si l’animation en couleur est obtenue à travers un tag de contrôle.
- La propriété *Couleur du texte* permet de remplacer la couleur du texte si l’option Coloriage général est activée.

- La propriété *Ombre portée* est utilisée pour activer un effet d'ombre optionnel qui s'applique sur la partie droite et le bas du texte. Cet effet est utile pour faire ressortir le texte de l'arrière-plan, en particulier lorsque ce dernier est une image contenant plusieurs couleurs. Cette propriété n'est disponible que pour l'option Coloriage général.
- D'autres propriétés dépendent de l'animation en couleur sélectionnée. Reportez-vous au chapitre Utilisation des colorations pour obtenir le détail des propriétés de chaque animation en couleur.

## AJOUTER DES ACTIONS A DES PRIMITIVES

Les primitives qui n'exécutent pas leur propre action implicite permettent l'ajout d'actions qui seront exécutées lorsque l'utilisateur touche ou relâche l'écran tactile. Vous pouvez ajouter une action soit en utilisant la commande Ajouter une action du menu contextuel de la primitive, soit en sélectionnant la primitive et en faisant la combinaison de touches **CTRL + I**. Un onglet Action sera ajouté aux propriétés des primitives et la boîte de dialogue suivante s'affichera...



## PROTEGER DES ACTIONS

Vous pouvez prévenir l'appel accidentel à une action en utilisant la propriété Protection de l'action. Cette fonctionnalité vient s'ajouter à toute protection fournie par la Sécurité Système, elle est appelée avant la mise en exécution des actions associées. Les modes de protection suivants sont disponibles...

- En mode *Confirmé* un pop-up s'affiche et demande de confirmer l'exécution de l'action. Celle-ci est effectuée aussitôt que l'utilisateur confirme l'exécution de l'action.
- En mode *Verrouillage* un pop-up s'affiche indiquant que l'action a été verrouillée. Si l'utilisateur confirme l'exécution de l'action, celle-ci est déverrouillée, et il doit activer l'action à nouveau pour l'exécuter. Sélectionner une autre action ou après un délai d'inactivité global aura pour effet de verrouiller l'action précédente.

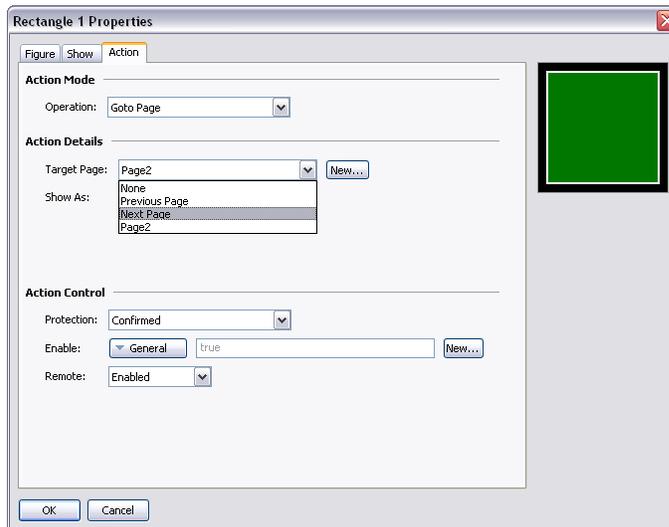
- Le mode *Double verrouillage* fonctionne de la même manière que le mode Verrouillage, sauf que l’action sera verrouillée après exécution, il est nécessaire de la déverrouiller à chaque utilisation.

## ACTIVER LES ACTIONS

Si vous voulez configurer une action afin qu’elle dépende d’une certaine condition ayant la valeur vrai, saisissez l’expression de cette condition dans le champ *Activer*. Cette expression peut dépendre directement d’un tag drapeau, ou peut être composée d’opérateurs logiques ou de comparaison définis dans la section Ecriture d’expressions. Si vous avez besoin d’une structure logique plus complexe, par exemple celle où une ou plusieurs actions sont exécutées en fonction de prises de décision plus complexes, configurez l’entrée correspondante sur mode Défini par l’utilisateur, et dans ce mode faites appel à un programme qui implémente la structure logique voulue. Vous pouvez aussi utiliser la propriété *A distance* pour bloquer l’accès à cette action à partir du Serveur web.

## L’ACTION ALLER A LA PAGE

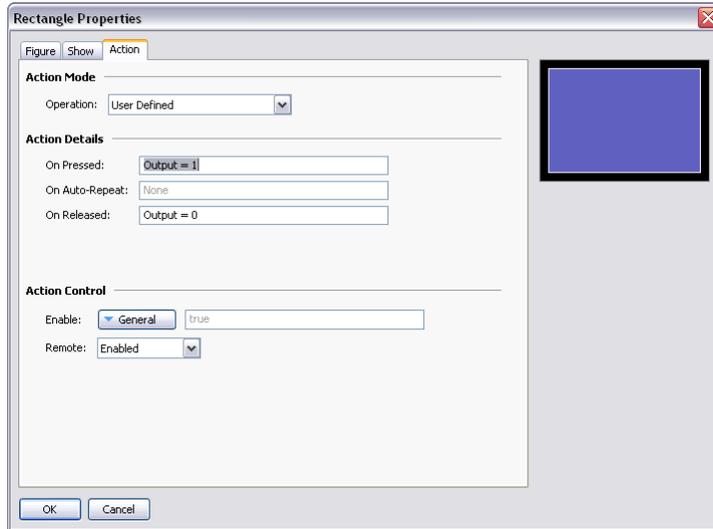
Cette action commande au périphérique cible l’affichage d’une nouvelle page...



- La propriété *Page Cible* indique quelle page doit être affichée. En plus des pages contenues dans la base de données, vous pouvez sélectionner Page Précédente ou Page Suivante pour naviguer dans l’historique. Vous pouvez utiliser le bouton Nouveau pour créer une nouvelle page sans quitter la boîte de dialogue.
- La propriété *Afficher sous* indique comment la page doit s’afficher. Avec l’option Page Normale la page est sélectionnée de la manière habituelle, alors qu’avec l’option Fenêtre Pop-up les primitives dans la nouvelle page seront affichées dans un pop-up rectangulaire en haut de la page active. Vous pouvez fermer un pop-up en exécutant la fonction `HidePopup()`.

## L'ACTION DEFINIE PAR L'UTILISATEUR

Cette action est utilisée pour exécuter des actions définies par l'utilisateur...

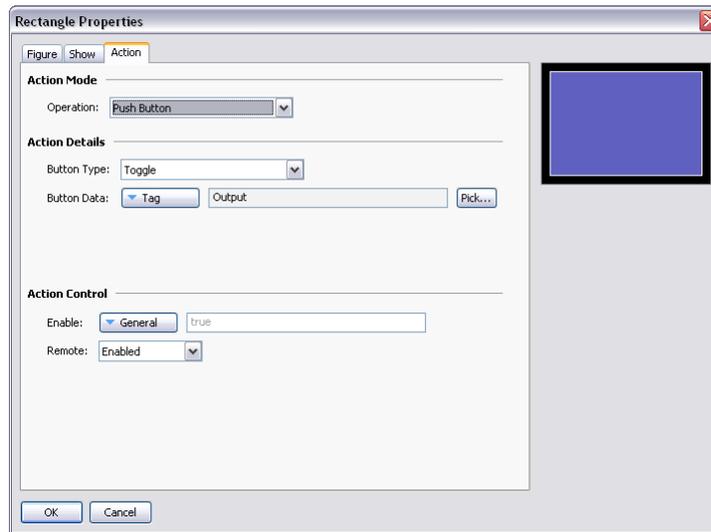


- La propriété *Sur appuyé* définit l'action qui sera exécutée lorsque la primitive est actionnée. Cette action peut faire appel à toute fonction de la fonction référence ou à tout opérateur de modification de données décrits dans le chapitre Ecriture d'actions. Elle peut aussi exécuter un programme pour effectuer des actions plus complexes.
- La propriété *Sur répétition auto* définit l'action qui sera exécutée lorsque la primitive est actionnée et maintenue enfoncée. L'action se produit à la fois lors de la pression initiale sur la primitive et à chaque répétition automatique ultérieure, il est donc inutile de définir à la fois cette propriété et la propriété *Sur appuyé*. Cette action peut faire appel à toute fonction de la fonction référence ou à tout opérateur de modification de données décrit dans le chapitre Ecriture d'actions, ou elle peut exécuter un programme.
- La propriété *Sur relâché* définit l'action qui sera exécutée lorsque la primitive est relâchée. Cette action peut faire appel à toute fonction de la fonction référence ou à tout opérateur de modification de données décrit dans le chapitre Ecriture d'actions, ou elle peut exécuter un programme.

Dans l'exemple au-dessus, une action définie par l'utilisateur est utilisée afin d'implémenter un bouton poussoir momentané.

## L’ACTION APPUYER SUR LE BOUTON

Cette action permet d’émuler un bouton poussoir...



- La propriété *Type de bouton* sélectionne la fonction recherchée...

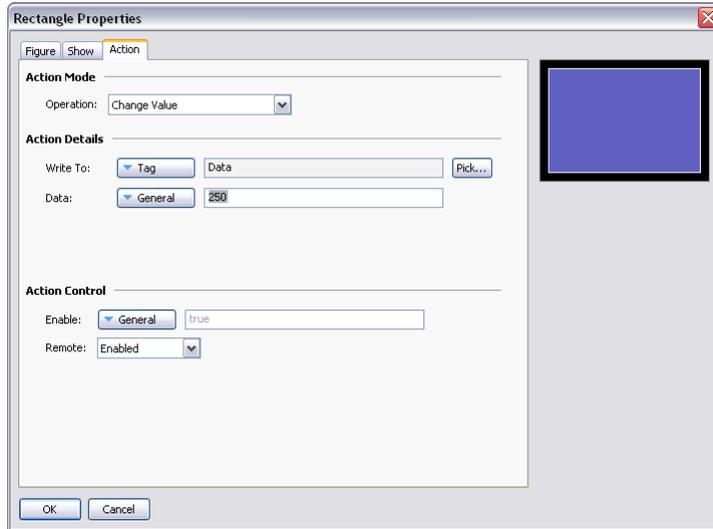
TYPE BOUTON	FONCTION DE LA PRIMITIVE
Basculer	Modifier l’état de la donnée lorsque la primitive est actionnée.
Momentané NO	Mettre la donnée à 1 lorsque la primitive est actionnée. Mettre la donnée à 0 lorsque la primitive est relâchée.
Momentané NF	Mettre la donnée à 0 lorsque la primitive est actionnée. Mettre la donnée à 1 lorsque la primitive est relâchée.
Allumer	Mettre la donnée à 1 lorsque la primitive est actionnée.
Eteindre	Mettre la donnée à 0 lorsque la primitive est actionnée.

- Le *Bouton Donnée* définit la donnée qui sera modifiée par la touche.

Dans l’exemple ci-dessus, appuyer sur la primitive aura comme effet de modifier la valeur du tag de sortie.

## L'ACTION MODIFIER LA VALEUR

Cette action permet de modifier la valeur numérique d'une donnée...

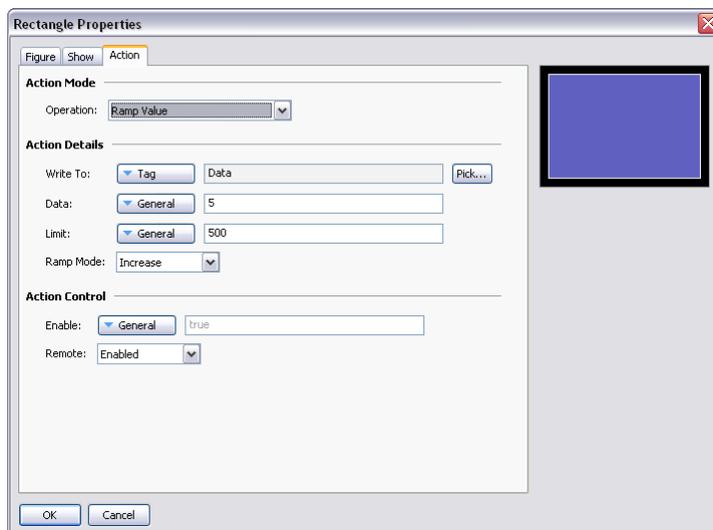


- La propriété *Ecrire sur* définit l'élément qui sera modifié.
- La propriété *Donnée* définit la donnée qui sera écrite.

Dans l'exemple ci-dessus, appuyer sur la primitive aura pour effet de mettre à 250 le tag de la donnée. Cette action est possible pour des valeurs entières ou à virgules flottantes. La propriété *Donnée* doit être appropriée à l'objet défini dans la propriété *Ecrire Sur*.

## L'ACTION MODIFIER VALEUR RAMPE

Cette action permet d'incrémenter ou de décrémenter la valeur d'une donnée. Les options possibles sont indiquées ci-dessous...



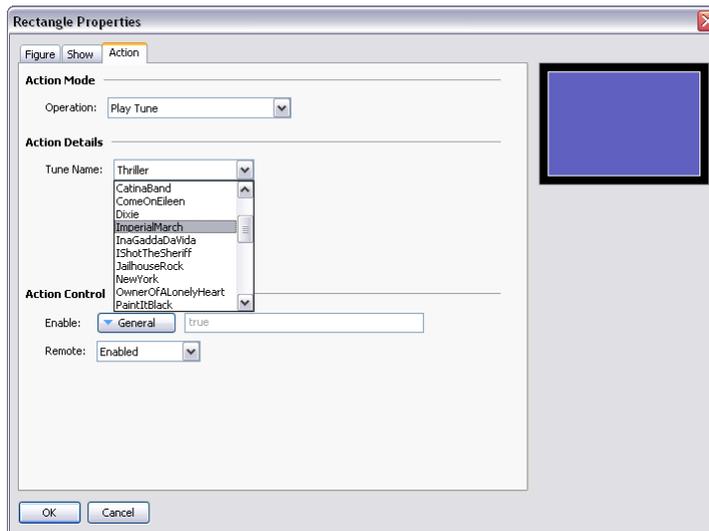
- La propriété *Ecrire Sur* définit l'élément qui sera modifié.
- La propriété *Donnée* définit le pas avec lequel incrémenter ou décrémenter l'élément.

- La propriété *Limite* définit les valeurs minimum et maximum de la donnée.
- La propriété *Mode rampe* indique s’il faut incrémenter ou décrémenter l’objet.

Dans l’exemple ci-dessus, appuyer sur la primitive et la maintenir enfoncée incrémente le tag de la Donnée par pas de 5 jusqu’à atteindre la valeur de 500. Cette action est possible pour des valeurs entières ou à virgules flottantes. Les propriétés Donnée et Limite doivent être appropriées à l’objet défini dans la propriété Ecrire Sur.

## L’ACTION JOUER UNE MELODIE

Cette action joue la mélodie sélectionnée en utilisant le système de son interne du périphérique cible.



- *Titre de la mélodie* sélectionne la mélodie qui sera jouée.

Vous pouvez jouer une mélodie personnalisée en utilisant la fonction `PlayRTTTL()`.

## L’ACTION CONNECTER UTILISATEUR

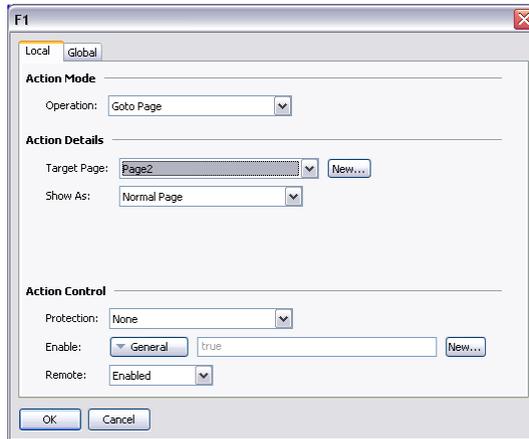
Cette action active l’écran de connexion sur le périphérique cible. Elle ne possède pas d’options.

## L’ACTION DECONNECTER UTILISATEUR

Cette action déconnecte l’utilisateur du périphérique cible. Elle ne possède pas d’options.

## AJOUTER DES ACTIONS A DES TOUCHES

Vous pouvez aussi ajouter des actions à des touches du périphérique cible. Faites un zoom arrière jusqu'à pouvoir visualiser les touches et faites un double clic sur une touche pour faire apparaître ses propriétés...



La boîte de dialogue affichée contient deux onglets, chacun d'entre eux définissant une action. Le premier onglet définit l'action qui sera exécutée par la touche lorsque la page active est affichée, le second définit une action qui sera exécutée dans toutes les pages. Ces deux types d'action sont appelées locale et globale.

La couleur d'affichage de la touche est différente en fonction du type d'action défini...



Si la touche est affichée en VIOLET, une action locale est définie pour cette page.



Si la touche est affichée en VERT, une action globale est définie.



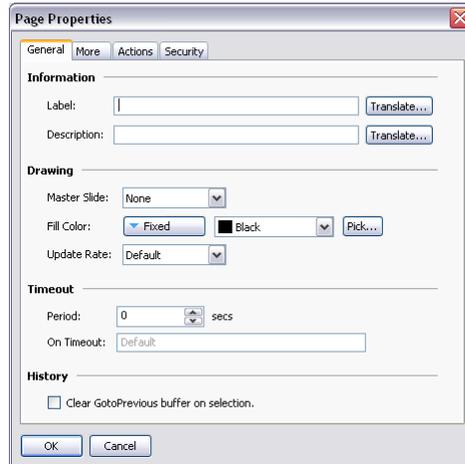
Si la touche est affichée en BLEU, des actions locales et globales sont définies.

Après avoir défini le type d'action, vous pouvez faire un clic droit sur la touche et sélectionner les commandes Rendre Locale ou Rendre Globale du menu qui s'affiche pour modifier le type d'action. Ces options ne sont pas disponibles si les deux types d'action ont été définis.

## EDITER LES PROPRIETES PAGES

Faites un clic-droit dans le panneau d’édition sans être proche d’une primitive pour activer le menu contextuel. Ce menu vous permet de sélectionner les commandes Propriétés afin d’éditer les propriétés d’une page d’affichage...

### PROPRIETES GENERALES



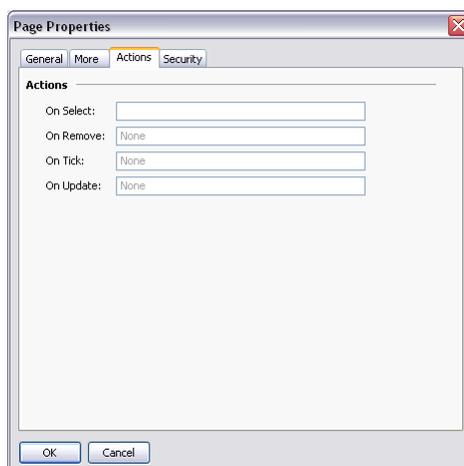
- Les propriétés *Label* et *Description* définissent les chaînes de caractères universelles traduisibles qui peuvent être accessibles ailleurs en utilisant la syntaxe d’extraction de Crimson. Reportez-vous au chapitre *Ecriture d’expressions* pour plus de détails.
- La propriété *Master Slide* (*Masque de diapositive*) permet la sélection d’une autre page qui sera utilisée comme arrière plan pour la page active. Elle permet l’utilisation d’éléments communs d’interface utilisateur comme par exemple des horloges, des indicateurs d’état ou autres qui seront dessinés sur une même page et intégrés par la suite à plusieurs autres pages.
- La propriété *Couleur de remplissage* définit la couleur de l’arrière plan, en supposant qu’un masque de diapositive n’ait pas été utilisé. Evitez d’animer la couleur de l’arrière plan, car les modifications apportées impliquent que le hardware redessine tous les objets de la page, ce qui peut causer une baisse de la performance.
- La propriété *Fréquence de mise à jour* définit la fréquence de mise à jour de la page. Dans des circonstances normales, le paramètre d’overdrive ne doit pas être utilisé. Le paramètre par défaut est actuellement équivalent au paramètre standard.
- Les propriétés *Inactivité* indique ce qui se passe après un délai d’inactivité. Si un intervalle de temps égal à *Période* se passe sans action de l’utilisateur, l’action *Sur inactivité* sera exécutée. Reportez-vous au chapitre *Ecriture d’actions* pour les détails des actions possibles.
- La propriété *Effacer / aller au tampon précédent* indique que l’historique du tampon conservé dans `GotoPrevious()` et `GotoNext()` doit être effacé lorsque la page est sélectionnée. Vous définissez typiquement cette propriété dans la page du menu principal de votre base de données, supprimant la possibilité de pouvoir revenir en arrière à partir de ce point.

## AUTRES PROPRIETES



- La propriété Liens permet de sélectionner un certain nombre de pages par des actions standards sur une page d'affichage. La propriété *Page parent* définit une page qui sera sélectionnée si le délai est dépassé et qu'aucune action n'a été définie. La propriété *Page suivante* définit la page qui sera sélectionnée si la navigation est activée et que vous déplacez le curseur au-delà du dernier champ de la page. La propriété *Page Précédente* définit la page qui sera sélectionnée si vous déplacez le curseur au-dessus du premier champ de la page.
- La propriété *Position* permet de remplacer pour cette page les paramètres de position des fenêtres pop-up définies globalement. Si les paramètres locaux sont activés, les propriétés *Horizontal* et *Vertical* sont utilisées pour définir la position.
- La propriété *Master Slide* indique si le master slide doit rester actif pendant l'affichage d'un pop-up. Par défaut le paramètre défini est Activé, ce qui permet le fonctionnement des boutons du master slide, alors que les boutons de la page active sont désactivés en présence d'un pop-up. Cette possibilité est utile si vous voulez que les options globales de navigation sur le master slide soient toujours disponibles.

## PROPRIETES ACTIONS



- La propriété *Sur sélectionné* définit l'action qui sera exécutée lors de l'affichage de la page.

- La propriété *Sur désélectionnée* définit l’action qui sera exécutée lorsque la page est désélectionnée.
- La propriété *Sur marque temporelle* définit l’action qui sera exécutée une fois par seconde.
- L’action *Sur actualisation* définit l’action qui sera exécutée à chaque actualisation de l’écran.

#### PROPRIETES SECURITE

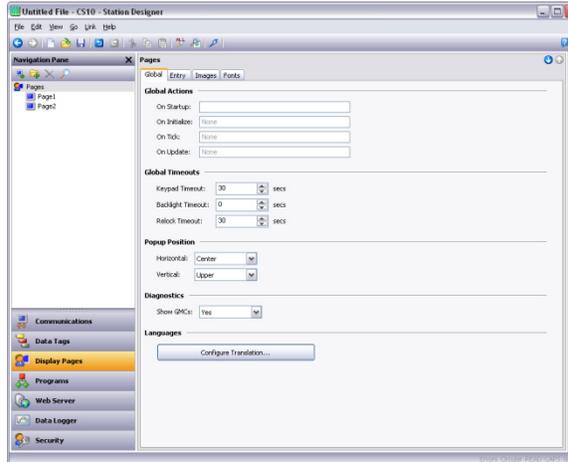
Reportez-vous au chapitre Sécurité pour plus de détails

## PARAMETRES DE L’INTERFACE UTILISATEUR

Sélectionnez la racine de la Liste de Navigation pour accéder aux paramètres de l’interface utilisateur.

### PROPRIETES GLOBALES

L’onglet Global contient divers paramètres généraux qui s’appliquent à toute la base de données...



### ACTIONS GLOBALES

- La propriété *Au démarrage* définit une action qui sera exécutée au démarrage du système.
- La propriété *Sur initialisation* définit une action qui sera exécutée un court instant plus tard<sup>1</sup>.
- La propriété *Sur marque temporelle* définit une action qui sera exécutée une fois par seconde.
- La propriété *Sur actualisation* définit une action qui sera exécutée à chaque actualisation de l’affichage.

### DELAIS GLOBAUX

- La propriété *Délai Clavier* définit l’intervalle de temps sans action de l’utilisateur après lequel toute opération de saisie de données sera annulée et le pop-up clavier associé sera supprimé de l’écran.

<sup>1</sup> La différence entre ces deux propriétés est assez subtile, la plupart des utilisateurs ne sont pas concernés.

- La propriété *Délai Eclairage* définit l'intervalle de temps sans action de l'utilisateur après lequel l'éclairage de l'écran sera éteint pour réduire la consommation d'énergie et prolonger la durée de vie de l'écran. La valeur par défaut zéro désactive cette fonction.
- La propriété *Délai Reverrouillage* définit l'intervalle de temps après lequel toutes les actions protégées par les méthodes Verrouillage et Double Verrouillage seront automatiquement verrouillées. Les utilisateurs devront les déverrouiller à nouveau avant de pouvoir les utiliser.

#### POSITION DU POPUP

- Les propriétés *Horizontal* et *Vertical* définissent la position par défaut des pop-up Page d'affichage et Claviers. Vous pouvez entrer de nouvelles valeurs pour une page donnée en utilisant les propriétés spécifiques de cette page.

#### DIAGNOSTIQUES

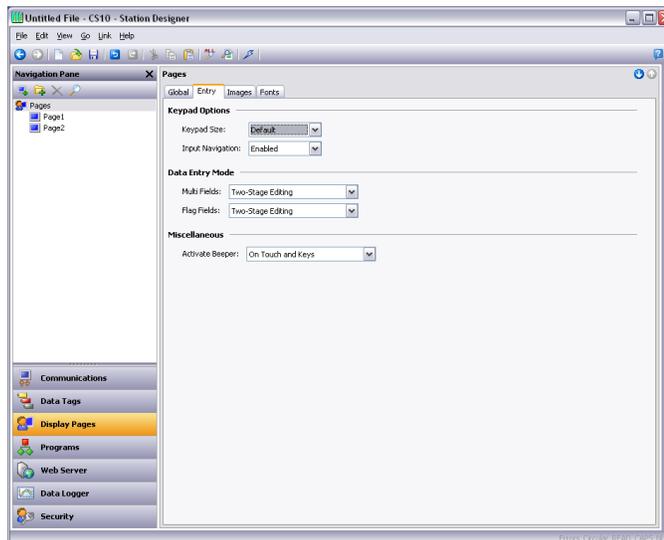
- La propriété *Afficher GMCs* permet d'activer ou de désactiver l'affichage de certaines informations de diagnostic après une défaillance du système lors de l'exécution. Cette information est utile à Red Lion pour lui permettre de corriger des problèmes de logiciel, mais est sans grand intérêt pour les utilisateurs communs.

#### LANGUES

- Le bouton *Configurer Traduction* permet de configurer les langues utilisées dans le système. Reportez-vous au chapitre Localisation pour plus d'informations.

#### PROPRIETES ENTREE

L'onglet Entrée contient des paramètres globaux qui s'appliquent aux données d'entrée...



#### OPTIONS CLAVIER

- La propriété *Taille Clavier* permet de sélectionner la taille de la donnée d'entrée clavier. Les différents paramètres permettent d'augmenter progressivement la taille du clavier, avec une valeur Maximum pour laquelle le clavier prend

presque tout l’espace de l’écran. Cette option est utilisée par exemple dans des situations où l’opérateur porte des gants encombrants.

- La propriété *Navigation d’entrée* permet d’afficher ou de masquer les touches **NEXT** et **PREVIOUS** des divers claviers. Ces touches peuvent être utilisées pour se déplacer parmi les champs d’entrée sans devoir préalablement désactiver le clavier.

#### MODE SAISIE DE DONNEES

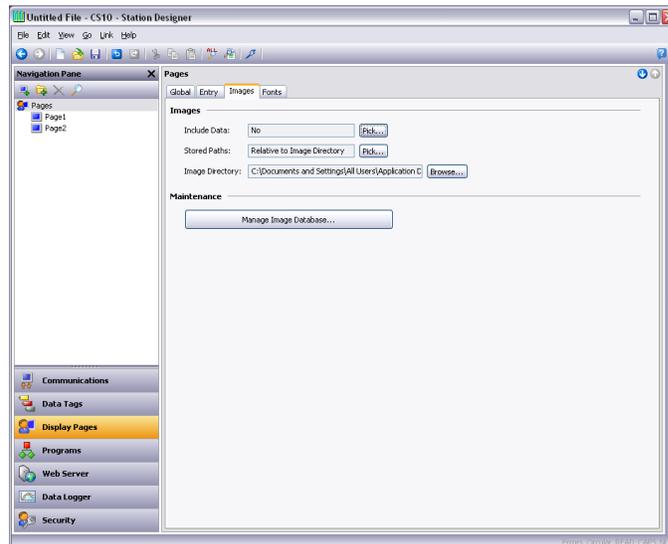
- La propriété *Saisie de données multiples* permet de contrôler le mode de saisie de données utilisé pour des objets à formats multiples. Dans l’édition en mode à deux-états, la touche **ENTER** doit être pressée pour effectuer un changement, alors qu’en mode simple la nouvelle donnée est écrite directement sur l’objet associé lorsque vous appuyez sur les touches **RAISE** ou **LOWER**. Le mode de saisie de donnée simple est plus rapide, mais peut conduire à l’écriture de valeurs intermédiaires lors d’une modification d’un paramètre d’état multiple.
- La propriété *Saisie de donnée drapeau* permet de contrôler le mode de saisie de données utilisé pour des objets à deux états. Elle fonctionne de la même manière que la propriété ci-dessus.

#### DIVERS

- La propriété *Activer le bipeur* permet d’activer ou de désactiver le bipeur. Celui-ci indique lorsqu’il y a une activité du clavier ou de l’écran tactile, mais il peut devenir gênant lors du cycle de développement.

#### PROPRIETES IMAGES

L’onglet Images permet de gérer les images dans une base de données...



#### IMAGES

- La propriété *Inclure Donnée* indique si les images externes qui ont été déposées dans la page d’affichage doivent être enregistrées en tant que pointeur vers l’emplacement source, ou bien incluses dans le fichier de la base de donnée.

L'inclusion d'images aura comme effet d'augmenter considérablement la taille de la base de données, ce qui peut rendre impossible l'utilisation de la fonctionnalité Support envoi sans occuper la mémoire du périphérique cible.

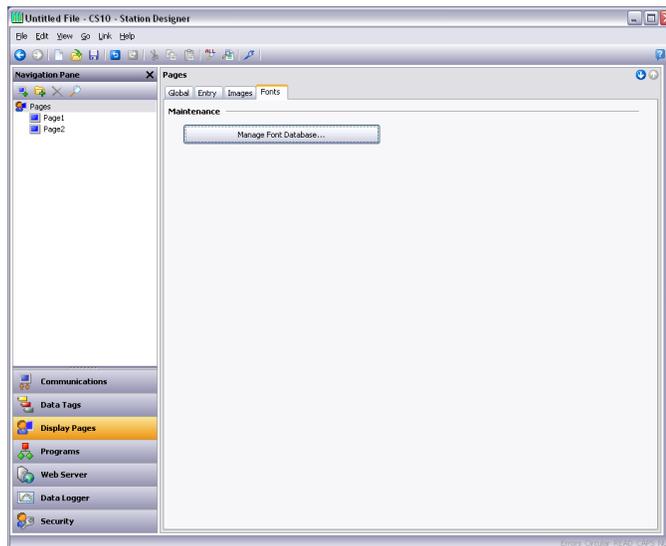
- La propriété *Chemins d'accès stockés* définit comment les liens vers les images sont gardés en mémoire. Le mode absolu garde en mémoire le chemin d'accès en entier, y compris la lettre du lecteur. Les deux modes relatifs gardent en mémoire et interprètent les chemins d'accès relatifs soit à la base de données soit au répertoire image de Crimson, ce qui permet de déplacer d'une machine à une autre la base de données et les fichiers images sans s'occuper des emplacements des chemins d'accès absolus.
- La propriété *Répertoire Image* définit le chemin d'accès aux images indiqués ci-dessus.

#### MAINTENANCE

- Le bouton *Gestionnaire Base de données Images* permet de faire appel au Gestionnaire Images afin de visualiser et de manipuler les images d'une base de données. Reportez-vous à la section ci-dessous pour plus d'informations sur cette fonctionnalité.

#### PROPRIETES DE POLICE

L'onglet Polices permet de gérer les polices dans une base de données...



#### MAINTENANCE

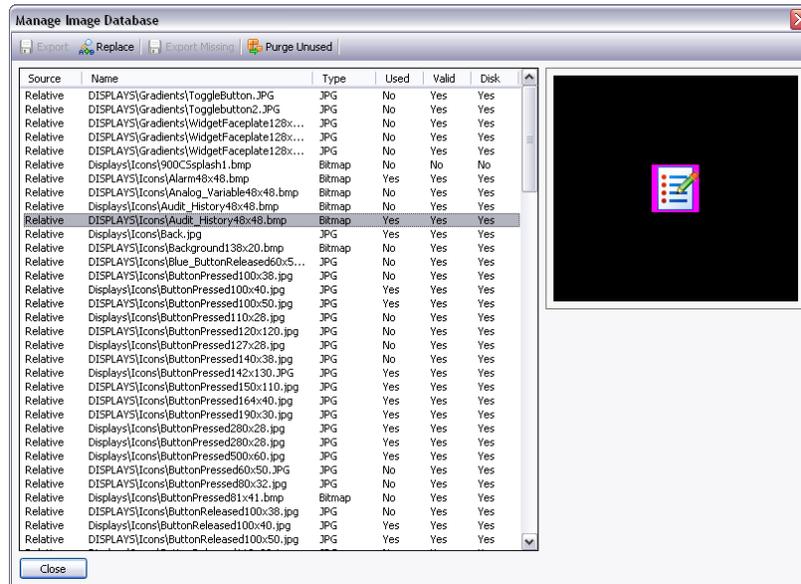
- Le bouton *Gestionnaire Base de données Polices* permet de faire appel au Gestionnaire Polices afin de visualiser et de manipuler les polices utilisées dans une base de données. Reportez-vous à la section ci-dessous pour plus d'informations sur cette fonctionnalité.

#### GESTION DES IMAGES

Vous pouvez accéder au Gestionnaire d'Images à partir de l'onglet Images des paramètres de l'interface utilisateur. Ce gestionnaire contient une liste de toutes les images référencées dans

la base de données et leurs propriétés. Il vous permet de visionner les images et d’effectuer certaines modifications dans la manière de les stocker et de les utiliser.

L’exemple ci-dessous présente le Gestionnaire d’Images d’une base de données complexe...



La liste principale affiche les propriétés des différentes images...

- La colonne *Source* indique si l’image d’un fichier provient d’un chemin d’accès fixe ou relatif, de la Bibliothèque Symboles ou de données internes stockées lorsqu’une image a été effacée ou déposée à partir d’une autre source.
- La colonne *Nom* affiche le nom du fichier pour les images stockées dans des fichiers et le symbole d’information approprié pour des images provenant de la Bibliothèque Symboles.
- La colonne *Type* affiche le type de fichier d’une image.
- La colonne *Utilisée* indique si une image est utilisée dans la base de données.
- La colonne *Valide* indique si une image valide est disponible. Vous pouvez mettre la valeur Non si l’image provient d’un fichier sur disque qui n’existe plus, ou si la base de données n’a pas été configurée pour conserver ses propres images via la propriété Inclure Données décrite ci-dessus.
- La colonne *Disque* indique si l’image est sauvegardée sur le disque. Des images qui ont été collées ou directement déposées dans l’éditeur peuvent ne jamais avoir été sauvegardées sur le disque, et des images provenant d’un fichier mais ayant été sauvegardées dans la base de données peuvent ne plus exister si le fichier n’est plus accessible.

La barre d’outils située sur la partie supérieure de la fenêtre vous permet d’utiliser certaines commandes...

- La commande *Exporter* sauvegarde une image qui est disponible mais qui n’est pas stockée sur un fichier du disque. Si un nom de fichier a déjà été défini pour

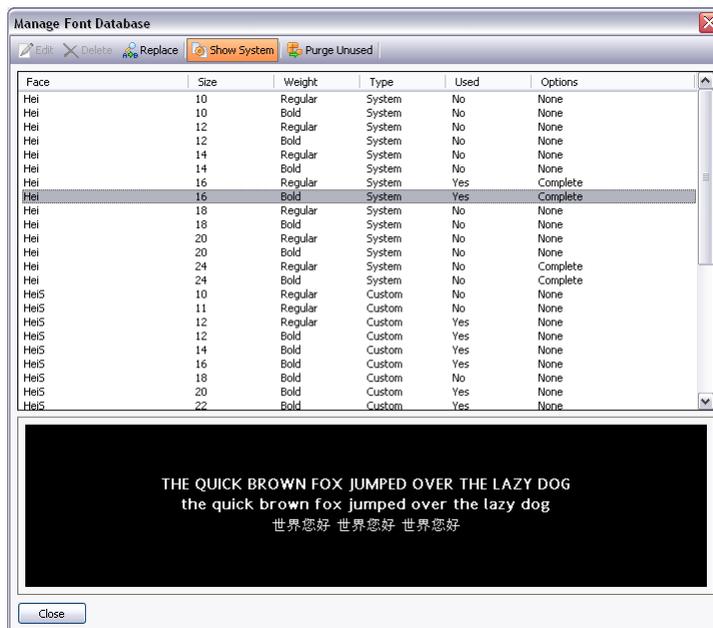
l'image sélectionnée, c'est ce nom qui sera utilisé. Dans le cas contraire, il vous sera demandé de choisir un nom de fichier.

- La commande *Remplacer* permet de remplacer une image par une autre. Toutes les références à cette image seront actualisées afin de tenir compte de ce changement.
- La commande *Exporter Tout* sauvegarde toutes les images disponibles qui ont un nom de fichier défini et qui n'ont pas encore été stockées sur le disque. Elle permet de s'assurer que toutes les images sont bien sauvegardées sur des fichiers externes avant de fermer la propriété Inclure Données.
- La commande *Purger non utilisées* d'effacer toutes les images de la base de données qui ne pas utilisées, permettant ainsi d'économiser de l'espace avant de sauvegarder la base de données sur le disque. L'utilisation de cette commande permet aussi de réduire l'espace mémoire du périphérique cible.

## GESTION DES POLICES

Vous pouvez accéder au Gestionnaire de Polices à partir de l'onglet Polices des paramètres de l'interface utilisateur. Ce gestionnaire contient une liste de toutes les polices référencées dans la base de données et leurs propriétés. Il vous permet de visionner les polices et d'effectuer certaines modifications dans la manière de les stocker et de les utiliser.

L'exemple ci-dessous présente le Gestionnaire de Polices d'une base de données complexe...



La liste principale affiche les propriétés des différentes polices...

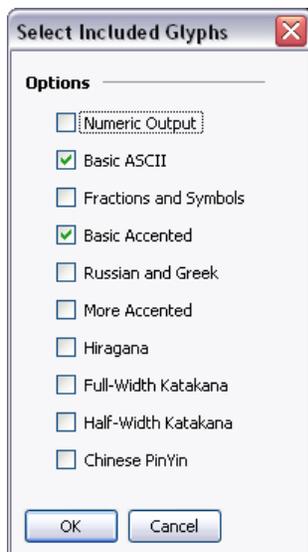
- La propriété *Face* affiche le nom de la police.
- La propriété *Taille* affiche la hauteur donnée en pixels de la police.
- La propriété *Gras* indique si le texte est affiché en gras ou pas.
- La propriété *Type* indique si la police est une police du système ou personnalisée.

- La propriété *Utilisée* indique si la police est utilisée dans la base de données.
- La propriété *Options* affiche les options sélectionnées pour la police.

La barre d’outils située sur la partie supérieure de la fenêtre vous permet d’effectuer certaines commandes...

- Le bouton *Editer* permet d’éditer les propriétés des polices personnalisées.
- Le bouton *Supprimer* permet de supprimer les polices non utilisées. Lorsque qu’une police est supprimée, elle n’apparaît plus dans la liste déroulante de sélection des polices, mais il est possible de l’y faire figurer à nouveau en utilisant le bouton Choisir associé.
- Le bouton *Remplacer* permet de remplacer une police par une autre. Toutes les références à cette police dans la base de données seront mises à jour afin de prendre en compte cette modification.
- Le bouton *Afficher Système* permet de définir si la police du système est affichée dans la liste.
- Le bouton *Purger non utilisées* supprime toutes les polices non utilisées de la base de données, permettant ainsi de réduire l’espace mémoire utilisé dans le périphérique cible. De même que pour les polices supprimées, les polices qui ont été purgées n’apparaissent plus dans la liste déroulante de sélection des polices, mais il est possible de l’y faire figurer à nouveau en utilisant le bouton Choisir associé.

Lorsque vous éditer une police personnalisée, la boîte de dialogue suivante s’affiche...



Les différentes options permettent d’inclure des ensembles particuliers de caractères à la représentation graphique de la police créée et transférée vers le périphérique cible. Afin d’économiser de la mémoire, en particulier pour les grandes polices, limitez-vous aux caractères nécessaires à votre application. Notez que vous pouvez utiliser uniquement l’option Sortie numérique pour restreindre la police à des chiffres, à des nombres décimaux ou à tout autre caractère utilisé pour des écritures scientifiques ou hexadécimales.

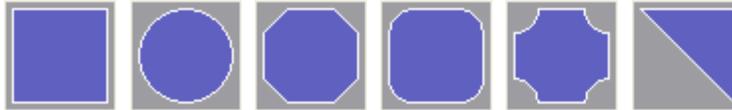


# TYPES DE PRIMITIVES

Ce chapitre décrit chacune des primitives fournies par Crimson.

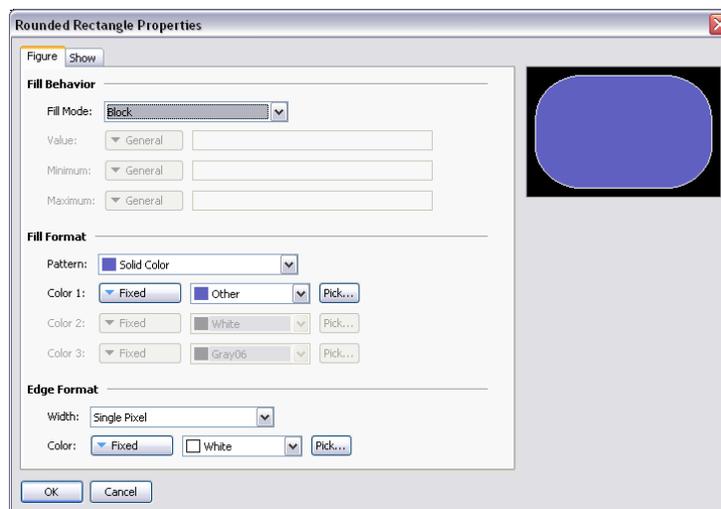
## PRIMITIVES NOYAU

### PRIMITIVES GEOMETRIQUES



Les Primitifs représentent des formes géométriques simples : un rectangle, un cercle, un rectangle coupé, un rectangle à coins arrondis, une plaque et un coin. Toutes ces primitives peuvent être utilisées pour appliquer des effets tels que des graphiques à barres. Elles supportent également l'ajout de texte ou de données. Enfin, elles supportent l'addition d'actions, et peuvent donc être utilisées pour mettre en œuvre les éléments d'affichage interactif.

L'onglet propriété spécifique des primitives est montré ci-dessous...

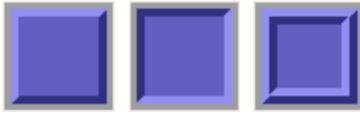


Reportez-vous au chapitre précédent pour les détails de remplissage standard et les paramètres de pointe. Notez que le coin a une propriété supplémentaire, à savoir une propriété *Position* utilisée pour spécifier l'orientation du prisme triangulaire dans le rectangle de délimitation.

Le rectangle coupé, le rectangle aux coins arrondis et la plaque ont tous une poignée de redimensionnement qui peut être utilisée pour spécifier le rayon de l'effet de coin. Dans leur forme dégénérée avec un rayon de coin nul, ils deviennent équivalents à un simple rectangle.

Alors que les Primitives géométriques sont très simples, leur support de données, textes et actions signifie qu'une grande partie de la plupart des bases de données peuvent en effet être créés en utilisant seulement le rectangle ou le rectangle arrondi.

## PRIMITIVES 3D



Les Primitives 3D représentent des rectangles avec une bordure en trois dimensions. Bien que les trois versions sont présentées dans Le panneau de ressources, tous ne sont réellement que des variations d'une seule primitive. Elles peuvent être utilisées pour appliquer des effets tels que des graphiques à barres. Elles supportent également l'ajout de texte ou de données, et peuvent donc être utilisées pour créer du texte ou des affichages de données, ou de fournir des données d'entrée. Enfin, elles supportent l'ajout d'actions, et peuvent donc être utilisées pour mettre en œuvre des éléments d'affichage interactif.

L'onglet spécifique à ces primitives est montré ci-dessous...



Reportez-vous au chapitre précédent pour les détails de remplissage standard et les paramètres de pointe. L'option *Style Bordure* sélectionne le type de bordure à tirer, en choisissant de manière efficace entre les trois versions pré-définies ci-dessus. La propriété *Largeur Bordure* définit le nombre de pixels qui seront alloués à chaque élément de pointe.

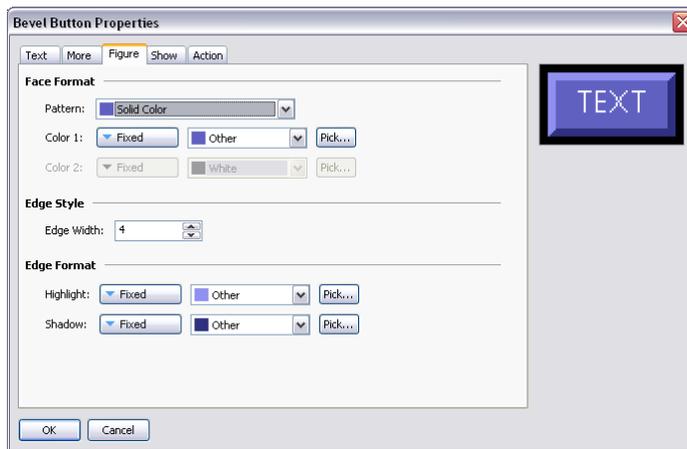
Comme pour les Primitives géométriques, les Primitive 3D peuvent être utilisées pour créer une large part d'une base de données standard du fait de leur support des données, textes et actions.

## PRIMITIVES BOUTON



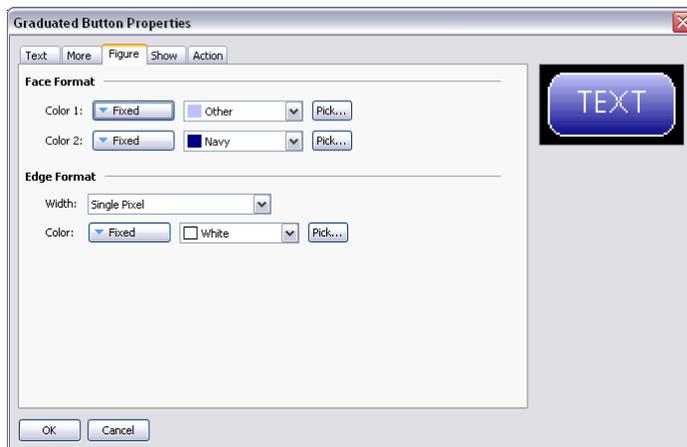
Les Primitives bouton peuvent être biseautés ou gradués. Les textes sont préconfigurés pour permettre au bouton d'être étiqueté, mais l'étiquette peut être enlevée pour permettre l'ajout de données en direct. Un onglet est en outre prévu par défaut, mais sera désactivé si les données en direct sont ajoutées et configurées. Les boutons avec champs de saisie de données utilisent la presse bouton pour activer l'édition.

L'onglet spécifique à un bouton biseauté est montré ci-dessous...



Reportez-vous au chapitre précédent pour les détails des réglages standard. La propriété *Largeur Bordure* définit le nombre de pixels qui seront alloués à chaque élément de pointe. Etant donné que cette primitive utilise toujours un style de bordure de la frontière, le bord sera de deux fois la largeur définie de pointe.

L'onglet spécifique d'un bouton gradué est montré ci-dessous...



Reportez-vous au chapitre précédent pour les détails des réglages standard.

## PRIMITIVES TEXTE ET DONNEES



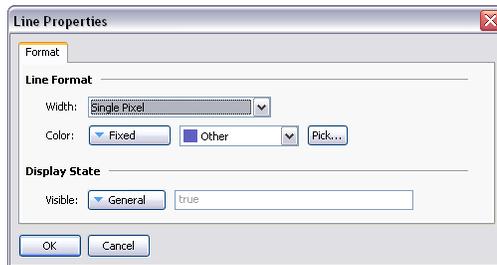
Les zones de texte et de boîte de données sont en fait des rectangles avec des données prédéfinies et des éléments de texte, possédant des couleurs aucun remplissage ou une bordure définie. Ils existent pour rendre plus facile l'ajout de données et d'éléments de texte, et pour fournir plus de confort aux utilisateurs qui ne sont pas habitués à construire une base de données entière à partir de primitives géométriques simples ! Ils peuvent également être utilisés pour ajouter une deuxième donnée ou élément de texte à une primitive ou lors de la construction d'un groupe.

Se référer à la section précédente pour les détails des réglages standard.

### PRIMITIVE LIGNE



La ligne primitive crée une ligne simple. La boîte de dialogue de propriété est indiquée ci-dessous...



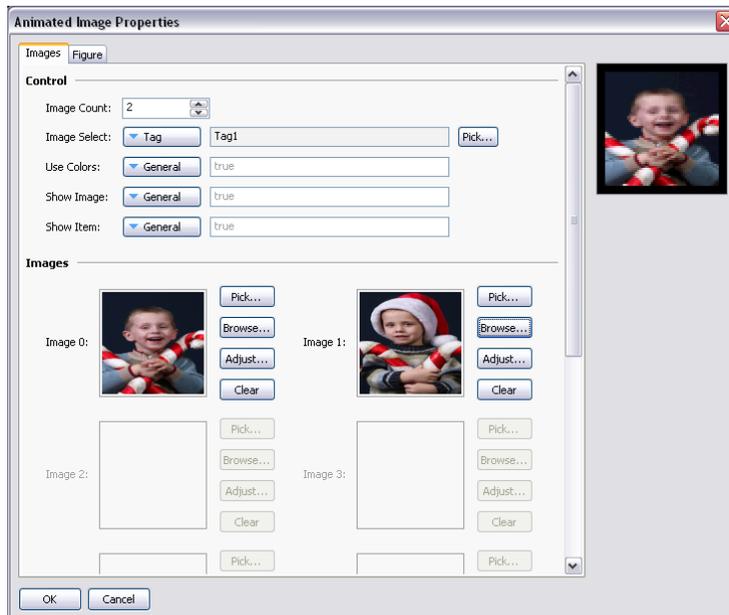
Reportez-vous au chapitre précédent pour les détails des réglages standard.

### PRIMITIVE IMAGE



La Primitive image est utilisée pour afficher une image qui peut-être choisie parmi de simages basées sur une valeur numérique. La primitive supporte l'affichage des images bitmap, JPEG ainsi que de nombreux autres types d'images. Elle peut fonctionner avec un fond transparent ou rempli et peut éventuellement définir un bordure. Elle supporte également l'ajout de données, de textes ou d'actions, permettant ainsi la construction d'éléments plus complexes.

L'onglet correspondant à cette fonction est montré ci-dessous...



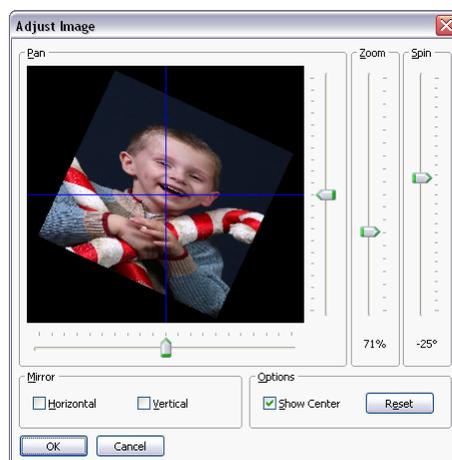
- La propriété *Décompte Images* définit le nombre de créneaux d'image définies pour cette primitive. L'une des images sera choisie pour l'affichage à un moment donné, en fonction de la valeur de la propriété Sélectionner une image.
- La propriété *Sélection d'Image* sélectionne l'image désirée. Cette dernière est traitée comme une base zéro de la valeur. En d'autres termes, si quatre images sont définies, les valeurs de 0, 4, 8, etc afficheront la première image, les valeurs de 1, 5, 9, etc afficheront la deuxième image et ainsi de suite.
- La propriété *Utilisation Couleur* permet de réduire une image en noir et blanc ou de préserver sa couleur. Une expression qui renvoie une valeur non nulle ou une expression vide amènera une image couleur. Une valeur de zéro va réduire l'image en niveaux de gris en utilisant la norme de coefficients de luminosité RGB. Cette option est utile lorsque l'on montre l'état désactivé d'une image sur un bouton.
- La propriété *Afficher l'image* est utilisée pour afficher ou masquer les images. Si la primitive ne possède pas de bordure elle fonctionnera comme la propriété *Afficher point*.
- La propriété *Afficher l'objet* est utilisé pour afficher ou masquer la primitive en entier.

#### DEFINITION DES IMAGES

La section *Images* de la boîte de dialogue permet de définir les images pour chaque emplacement. Le bouton choisir placé à côté de chaque image va afficher une boîte de dialogue vous rappelant que vous pouvez simplement faire glisser une image sur le champ. Cette image peut être prise dans la catégorie Bibliothèque de symboles du panneau de ressources, d'un dossier de l'Explorateur Windows ou de toute autre application capable d'effectuer un glisser-déposer. Le bouton Parcourir peut être utilisé pour ouvrir un fichier contenant un format d'image convenable et pour charger ce fichier. Comme mentionné ci-dessus, Les formats PEG, métafichiers, bitmaps et de nombreux autres formats de fichiers sont supportés.

#### AJUSTEMENT DES IMAGES

Le bouton Ajuster permet de modifier l'image...



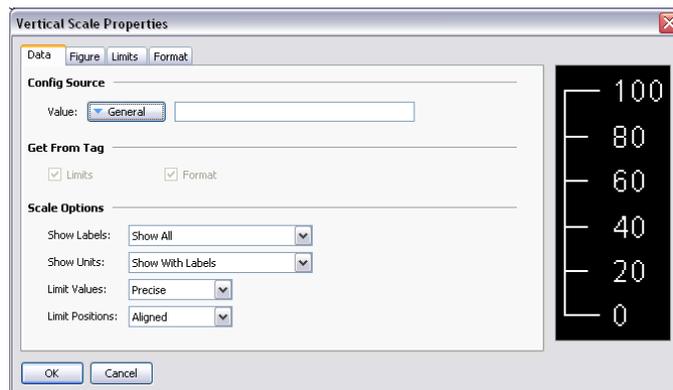
Les différents curseurs peuvent être utilisés pour effectuer un panoramique, un zoom ou une rotation de l'image, tandis que les cases à cocher peuvent être utilisées pour un miroir horizontal ou vertical. Les options de manipulation sont parfois utilisées pour modifier une image de façon à créer divers états différents pour une utilisation dans l'animation.

### MISE A L'ECHELLE PRIMITIVE



La primitive Mise à l'échelle est utilisée pour établir une échelle verticale. Les limites de l'échelle peuvent être définies comme étant des constantes, ou pouvant être modifiées en fonction de la valeur des expressions spécifiques. L'échelle peut être étiquetée ou non étiquetée, toutes les étiquettes étant fondées sur un objet au format spécifié pouvant éventuellement être obtenu à partir d'une balise.

### PROPRIETES DE DONNEES

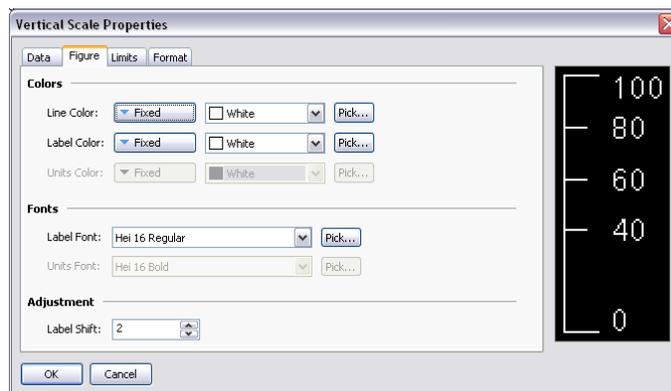


- La propriété *Valeur* définit une balise facultative qui sera utilisée pour obtenir des limites et des informations sur le format de l'échelle. La valeur en soi n'est pas réellement utilisée par la primitive, mais le tag est simplement utilisé comme une source d'informations complémentaires.
- La propriété *Prendre depuis le tag* est utilisée pour indiquer si l'option tag définie dans la propriété valeur doit être utilisée comme source pour les données en question.
- La propriété *Afficher légende* est utilisée pour afficher ou masquer les légendes d'échelle numérique.
- La propriété *Afficher unités* est utilisée pour afficher ou masquer les unités définies par un format de données numériques. Les unités peuvent être annexée à chaque légende ou être tracées verticalement depuis le bord de l'échelle.
- La propriété de *Limite* des valeurs spécifie comment les valeurs haut et bas de l'échelle sont déterminées. Si un réglage de précision n'est pas spécifié, les valeurs limites seront utilisées exactement, même si certaines ne correspondent pas exactement à l'espacement automatiquement sélectionné. Cela peut produire des légendes non régulières. Un réglage de l'arrondi permet une mise à l'échelle

sur la primitive ajustant automatiquement les limites pour atteindre des graduations régulièrement espacées.

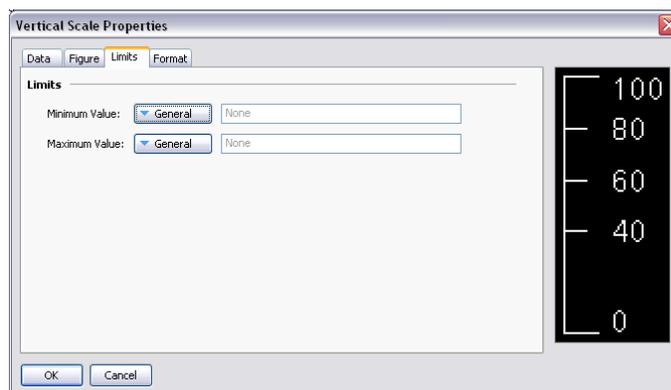
- La propriété *Limites de positions* spécifie comment les limites de l'échelle portent sur les légendes d'unités. Une valeur alignée conserve les marques de graduation et les étiquettes alignées avec précision, au coût de l'enlèvement des marques de graduation extérieure vers l'intérieur du bord du primitif. Vous choisissez un paramètre de mouvements entre les deux étiquettes extérieure par rapport aux marques de graduation, qui permet au minimum et maximum de s'aligner avec le bord de la primitive, ce qui rend plus facile l'alignement avec, par exemple, un réservoir de remplissage.

### PROPRIETES FIGURE



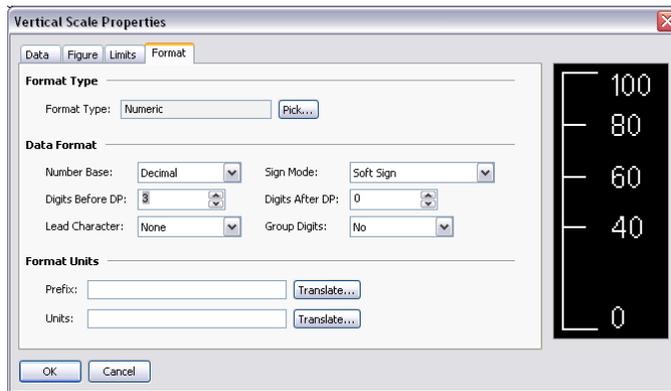
Les propriétés présentes sur cette page définissent les couleurs et les polices utilisées pour l'échelle. Reportez-vous au chapitre précédent pour le détail des propriétés standard. La propriété *Déplacement Etiquette* peut être utilisée pour déplacer les étiquettes vers le haut ou vers le bas par rapport aux marques de graduation, en produisant des résultats plus attractifs lorsque l'on travaille avec des polices avec un espacement supérieur ou inférieur au glyphes caractère.

### PROPRIETES LIMITES



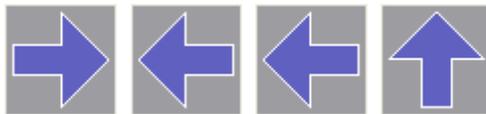
Les propriétés de cette page sont utilisées pour définir les valeurs minimales et maximales indiquées par l'échelle. Les expressions peuvent demander, dans ce cas, à Crimson de mettre à jour dynamiquement l'échelle lors de l'exécution, en choisissant les marques de graduation et de positions de légende appropriées aux nouvelles valeurs. Ces paramètres peuvent ne pas être disponibles si une balise a été choisie comme source des valeurs limites.

## PROPRIETES DE FORMAT



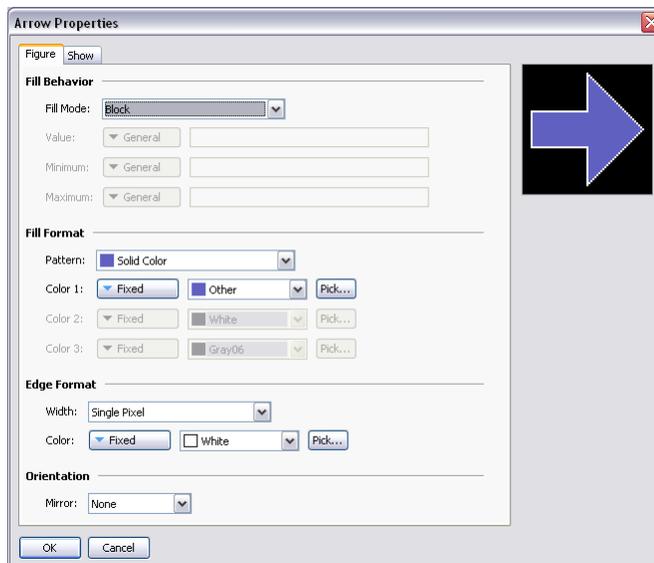
Le champ *Type de format* spécifie le type de format à utiliser lors de l'élaboration des étiquettes échelle. Seul le format Général ou Numérique est pris en charge. La sélection peut ne pas être disponible si le format est obtenu à partir d'un tag. Reportez-vous à la section Utilisation des formats pour plus de détails sur les différentes propriétés affichées lorsque le format de données numériques est sélectionné.

## FLECHES



Les primitives des quatre flèches sont en fait des versions prédéfinies d'une seule primitive. Elles supportent également l'ajout de texte ou de données, et peuvent donc être utilisés pour fournir des données d'entrée. Enfin, elles supportent l'ajout d'actions, et peuvent donc être utilisées pour mettre en œuvre les éléments d'affichage interactif.

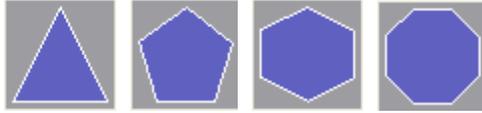
L'onglet spécifique est présenté ci-dessous...



Reportez-vous au chapitre précédent pour les détails des réglages standard. La propriété *Miroir* est utilisée pour contrôler la direction de la flèche. C'est cette propriété qui est utilisée pour produire les quatre versions prédéfinies montrés dans Le panneau de ressources.

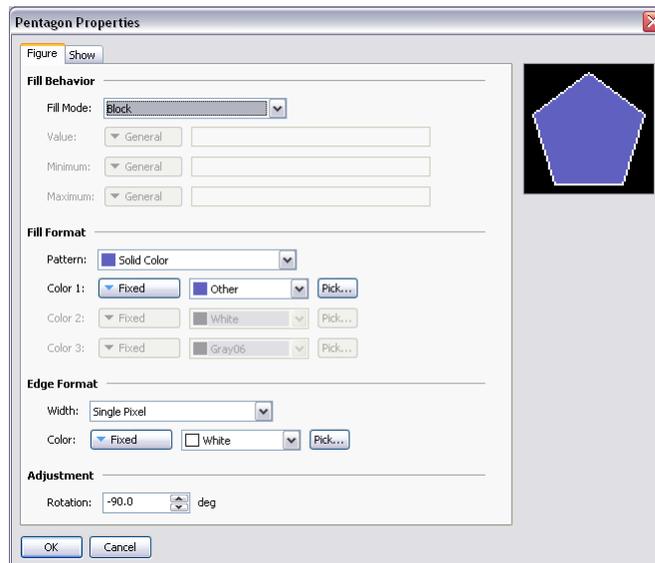
## POLYGONES ET ÉTOILES

### POLYGONES



Ces primitives sont utilisées pour afficher des polygones réguliers : un triangle, un pentagone, un hexagone et un octogone. Elles supportent également l'ajout de texte ou de données, et peuvent donc être utilisées pour fournir des données d'entrée. Enfin, elles soutiennent l'addition d'actions, et peuvent donc être utilisées pour mettre en œuvre les éléments d'affichage interactif.

L'onglet spécifique de ces primitives est montré ci-dessous...



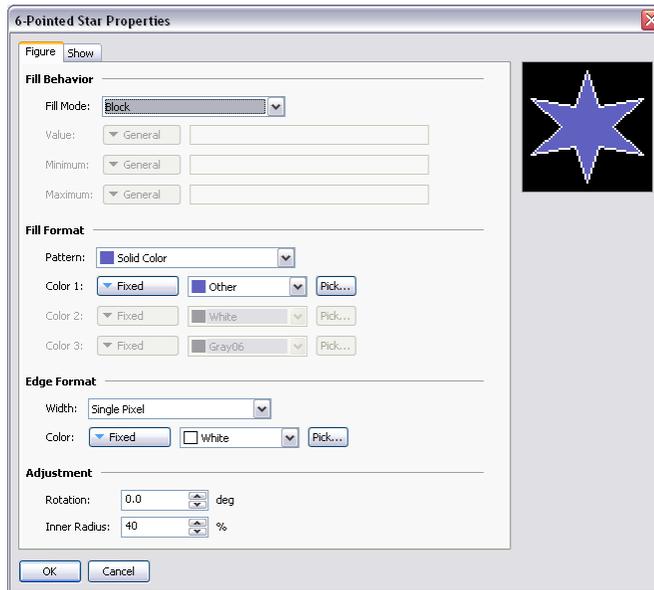
Reportez-vous au chapitre précédent pour les détails du remplissage standard et les paramètres de pointe. La propriété *Rotation* peut être utilisée pour faire pivoter le polygone dans le rectangle de délimitation. Les axes x-et y sont réduits en fonction de la largeur totale et la hauteur du polygone.

### ÉTOILES



Ces primitives représentent des étoiles régulières avec quatre, cinq, six et huit branches. Elles supportent également l'ajout de texte ou de données, et peuvent donc être utilisés pour fournir des données d'entrée. Enfin, elles supportent l'addition d'actions, et peuvent donc être utilisées pour mettre en œuvre les éléments d'affichage interactif.

L'onglet spécifique de ces primitives est montré ci-dessous...



Reportez-vous au chapitre précédent pour les détails du remplissage standard et les paramètres de pointe. La propriété *Rotation* peut être utilisée pour faire pivoter le polygone dans le rectangle de délimitation. Les axes x-et y sont réduits en fonction de la largeur totale et de la hauteur du polygone. La propriété *Angle Interne* est utilisée pour changer les branches de l'étoile. (Cette propriété contrôle le rapport des rayons.)

## INFO-BULLES



La primitive info-bulles fournie peut être utilisée pour marquer des points sur une page ou fournir une aide aux opérateurs. Elle supporte l'ajout de texte et de données, et peut également être configuré pour afficher un réservoir de remplissage. Elle supporte également l'ajout d'actions, et peut donc être utilisés pour mettre en œuvre les éléments d'affichage interactif.

La forme de l'info-bulle est contrôlée par un certain nombre de poignées de mise en page...



La poignée située dans le coin supérieur gauche contrôle le rayon des angles. La poignée centrale contrôle la hauteur du corps de l'info-bulle par rapport à sa pointe. Le texte situé dans la bulle sera redistribué automatiquement quand les poignées seront déplacés.

## FIGURES SEMI-HACHURÉES



Les figures semi- hachurées sont des versions du rectangle arrondi et de la plaque qui n'ont que deux de leurs coins retirés. Elles sont utiles pour créer des barres de titre et d'autres effets sur les bordures de groupes de primitives. Elles sont toutes disponibles en quatre orientations.

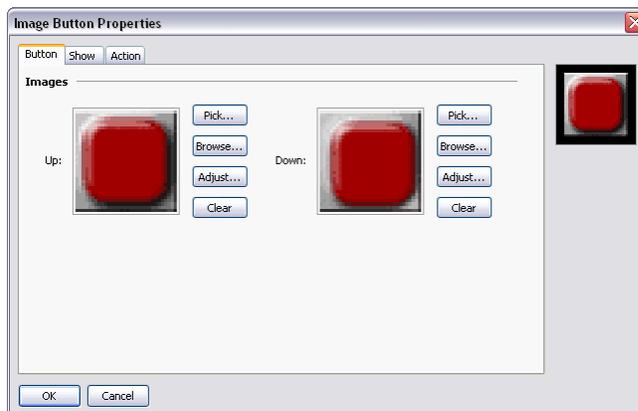
## BOUTONS D'ACTION



Les boutons d'action utilisent des images présélectionnées de la bibliothèque de symboles pour créer un bouton qui permettra d'exécuter une action donnée quand il est enfoncé. De nombreuses versions sont fournies en plus de celles indiquées ci-dessus. En cliquant sur un bouton Données dans le Panneau de Ressources vous verrez les variantes de couleurs différentes qui sont disponibles. Par exemple, le bouton carré est disponible en rouge, vert ou noir...



Lorsque vous utilisez un bouton d'action, vous devrez d'abord utiliser l'onglet action de la boîte de dialogue des propriétés pour définir une action selon la description donnée dans le chapitre précédent. L'onglet bouton peut également être utilisé pour ajuster les images des boutons, ou définir votre propre version...



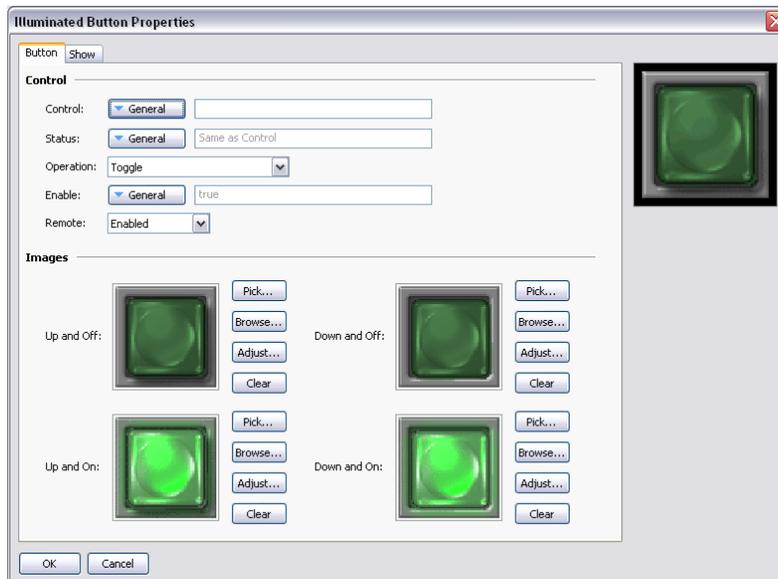
## BOUTONS ILLUMINES



Les boutons Illuminés utilisent des images présélectionnées à partir de la bibliothèque de symboles pour créer un bouton qui permettra de contrôler une balise, et s'allument en fonction de l'état de cette balise ou de l'état d'une autre expression. De nombreuses versions sont fournies au-delà de celles indiquées ci-dessus. En cliquant sur un bouton donnée dans le Panneau de Ressources vous verrez des variantes de couleurs différentes qui sont disponibles. Par exemple, le bouton Rondci-dessus est disponible en rouge, vert, jaune, bleu ou gris...



L'onglet spécifique à ces primitives est montré ci-dessous...



- La propriété *Contrôle* définit la valeur à écrire lorsque le bouton est enfoncé ou relâché. Cette valeur doit être accessible en écriture, et sera fixe ou nulle selon le fonctionnement exact défini pour le bouton.
- La propriété *Status* est utilisée pour contrôler l'éclairage de la touche. Si elle est laissée vide, ce dernier sera par défaut la valeur de contrôle. Une valeur différente peut être utilisée si une logique plus complexe est nécessaire.

- La propriété *Opération* sélectionne le comportement requis...

OPÉRATION	COMPORTEMENT DES BOUTONS
Bascule	Changer l'état des données lorsque le bouton est pressé.
Verrouillage	Si la donnée est 0, mise à 1 lorsque le bouton est pressé. Si les données sont de 1, mise à 0 lorsque le bouton est relâché.
NO Momentary	Définit les données à 1 lorsque le bouton est pressé. Définit les données à 0 lorsque le bouton est relâché.
NC Momentary	Définit les données à 0 lorsque le bouton est pressé. Définit les données à 1 lorsque le bouton est relâché.
Allumage	Définit les données à 1 lorsque le bouton est pressé.
Extinction	Définit les données à 0 lorsque le bouton est pressé.

Notez que de verrouillage est légèrement différent de la bascule par rapport au point à partir duquel une valeur non nulle de contrôle est définie à zéro. En bascule, les changements sont apportés lorsque le bouton est enfoncé alors qu'en verrouillage une valeur OFF est enclenchée lorsqu'il est libéré. Cela produit un résultat plus conforme au comportement d'un véritable bouton poussoir.

Reportez-vous au chapitre précédent pour les détails sur la *Protection*, *Activation* et les propriétés à *Distance*, et plus haut dans ce chapitre pour plus de détails sur la manière de changer ou d'ajuster les images de boutons divers. Comme vous pouvez le voir dans l'exemple ci-dessus, quatre images sont nécessaires pour représenter les états des boutons.

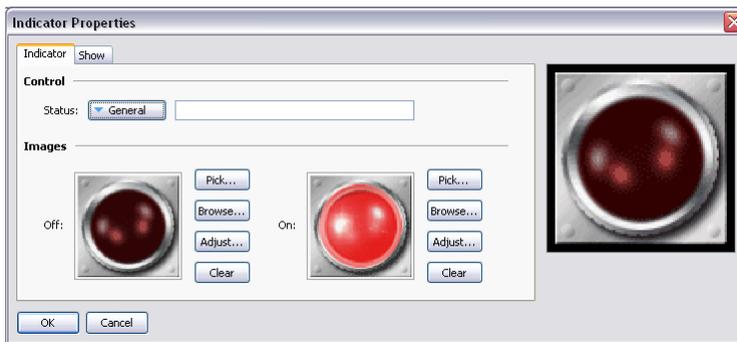
## INDICATEURS



Les indicateurs utilisent des images présélectionnées de la bibliothèque de symboles pour afficher le statut On / Off d'une valeur de données. De nombreuses versions sont fournies en plus de celles indiquées ci-dessus. En cliquant sur le bouton données dans le Panneau de Ressources on affichera les variantes de couleurs différentes qui sont disponibles. Par exemple, l'indicateur pilotes ci-dessus est disponible en rouge, vert, jaune, bleu ou blanc...



Indicateurs avec un jeu très simple de propriétés...



La propriété *Status* contrôle les images pour en être tirées. Toutes les autres propriétés sont standard.

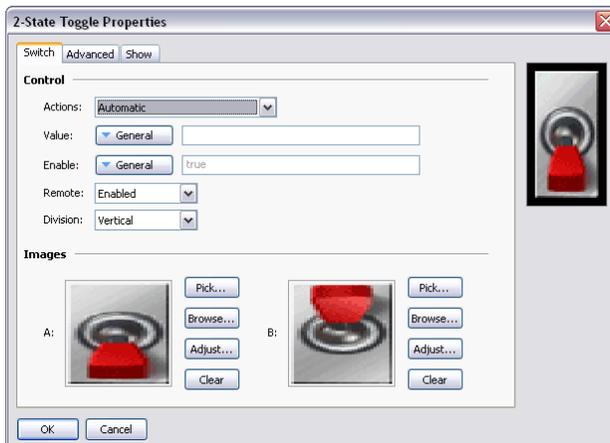
## BASCULES 2-ETATS



Les bascules 2-Etats utilisent des images présélectionnées de la bibliothèque de symboles pour mettre en œuvre des commutateurs à bascule avec positions haute et basse. De nombreuses versions sont fournies au-delà de celles indiquées ci-dessus. En cliquant sur une bascule donnée dans le Panneau de Ressources on affichera les variantes de couleurs différentes qui sont disponibles. Par exemple, le commutateur est disponible en rouge, vert ou noir...



## PROPRIETES DES INTERRUPTEURS

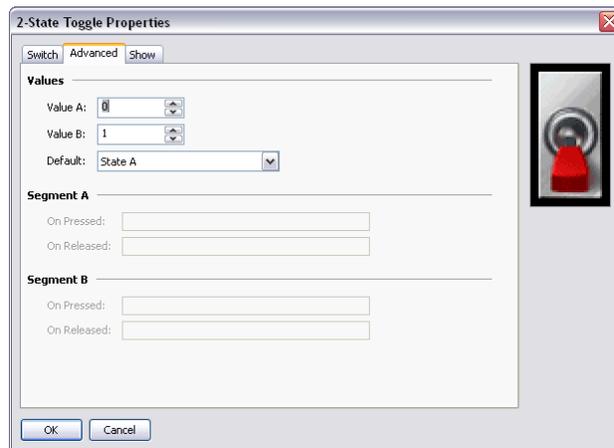


- La propriété *Actions* contrôle le comportement de l'interrupteur. Il existe trois modes automatiques que le mode Défini par l'utilisateur vous permet de spécifier des actions plus complexes qui se produisent lorsque la moitié de l'interrupteur est enfoncé ou relâché.
- La propriété *Valeur* est utilisée dans les modes automatiques et sera inscrite dans les données associées avec les États A et B que l'interrupteur modifie. Par défaut, l'Etat A est représenté par un zéro et l'Etat B par un, mais ces valeurs peuvent être modifiées en utilisant les paramètres avancés de cette primitive.
- La propriété *Divisions* définit si l'interrupteur est lancé verticalement ou horizontalement, et permet donc de savoir comment Crimson devrait diviser la primitive lors de l'interprétation touchée par l'utilisateur.

Reportez-vous au chapitre précédent pour plus de détails.

Reportez-vous aux paragraphes précédents de ce chapitre pour savoir comment changer ou d'ajuster les images commutateur.

#### PROPRIETES AVANCEES



- Les propriétés de *Valeur A* et *Valeur B* définissent les valeurs de données utilisées dans le mode automatique pour représenter les deux états de l'interrupteur. La valeur lue sera comparée à ces deux valeurs pour décider quel état d'affichage doit écrire la valeur appropriée.
- La propriété *Défaut* sélectionne l'Etat à être affiché si les données lues sur la propriété de la valeur ne correspondent à aucune valeur A ou B.
- Les propriétés *Sur appuyé* et *Sur relâché* définissent des comportements personnalisés à lancer lorsque les parties A et B de l'interrupteur sont pressées ou libérées par l'utilisateur. Pour un interrupteur vertical, A représente la moitié inférieure et B la moitié supérieure. Pour un interrupteur horizontal, A représente la moitié gauche et B la moitié droite.

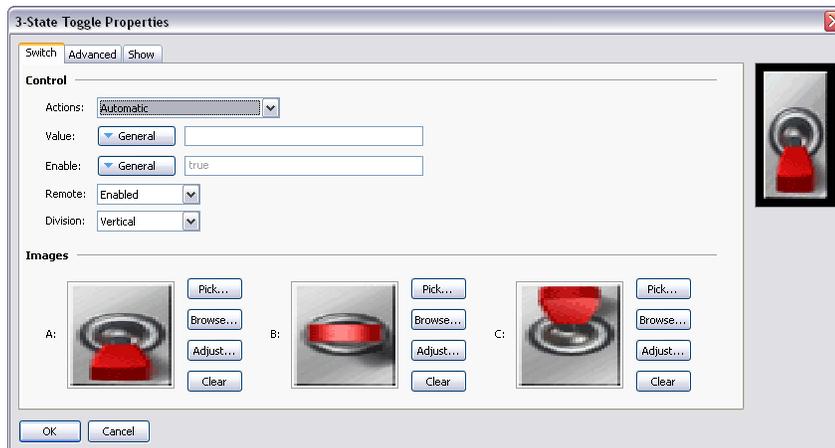
## BASCULES 3-ETATS



Les bascules 3-Etats utilisent des images présélectionnées de la bibliothèque de symboles pour mettre en œuvre les commutateurs à bascule, au centre et vers le bas. D'autres versions sont prévues au-delà de celles indiquées ci-dessus. En cliquant sur une bascule présentée dans le Panneau de Ressources on affichera les variantes de couleurs différentes qui sont disponibles. Par exemple, le commutateur est disponible en trois couleurs...



### PROPRIETES DS INTERRUPTEURS

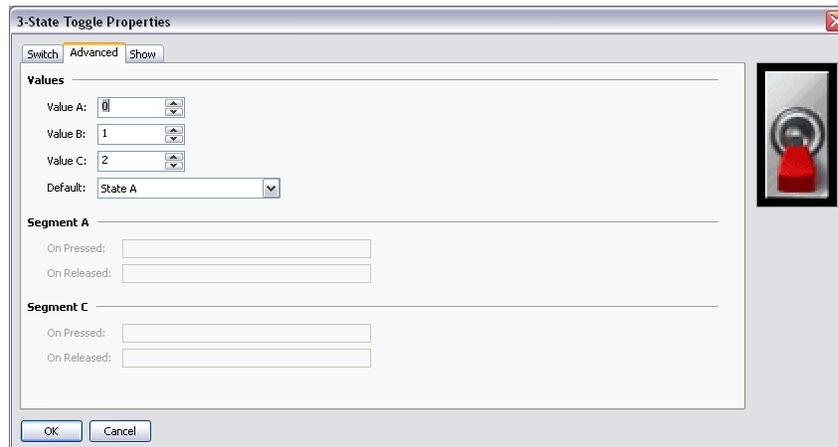


- La propriété *Actions* contrôle le comportement de l'interrupteur. Il existe quatre modes automatiques, tandis que le mode Défini par l'Utilisateur vous permet de spécifier des actions plus complexes qui se produisent lorsque la moitié de l'interrupteur est enfoncée ou relâchée. Notez que le commutateur ne peut être déplacé de plus d'un poste à la fois comme cela se produirait avec un interrupteur à bascule physique.
- La propriété *Valeur* est utilisée dans les modes automatique et sera décrite dans les données de valeurs données associées aux Etats A, B ou C, selon le commutateur changé. Par défaut, l'Etat A est représenté par un zéro, l'Etat B par un et de l'Etat C par un deux, mais ces valeurs peuvent changer en utilisant les paramètres avancés de cette primitive.
- La propriété *Divisions* est utilisée si l'interrupteur est lancée verticalement ou horizontalement, et permet donc de savoir comment Crimson devrait diviser le primitif lors de l'interprétation touche par l'utilisateur.

Reportez-vous au chapitre précédent pour les détails de la *Protection et de la validation à distance* des propriétés.

Reportez-vous aux paragraphes précédents de ce chapitre pour plus de détails sur la manière de changer ou d'ajuster les images commutateur.

### PROPRIETES AVANCEES



- Les propriétés de *Valeur A*, *B* et *C* définissent données utilisées dans le mode automatique pour représenter les trois états de l'interrupteur. La valeur lue sera comparée à ces valeurs pour décider quel est l'état d'affichage, et changer le switch en inscrivant la valeur appropriée.
- La propriété *Défaut* sélectionne l'Etat à être affiché si les données lues sur la propriété de la valeur ne correspondent pas aux valeurs A, B ou C.
- Les propriétés *Sur appuyé* et *Sur relâché* définissent des comportements personnalisés à effectuer lorsque A et une partie C de l'interrupteur sont pressées ou libérées par l'utilisateur. Pour un interrupteur vertical, A représente la moitié inférieure et C est la moitié supérieure. Pour un interrupteur horizontal, A représente la moitié gauche et C est la moitié droite.

### SELECTEURS 2-ETATS



Les sélecteurs 2-Etats utilisent des images présélectionnées de la bibliothèque de symboles pour mettre en œuvre des sélecteurs rotatifs à deux états. Leur comportement est identique à aux bascules 2-états, et elles sont mises en œuvre en utilisant les mêmes primitives.

### SELECTEURS 3-ETATS



Les sélecteurs 3-Etats utilisent des images présélectionnées de la bibliothèque de symboles pour mettre en œuvre des sélecteurs rotatifs à trois états. Leur comportement est identique aux bascules 3-états et elles sont mises en œuvre en utilisant les mêmes primitives.

## PRIMITIVES HERITEES

Ces primitives sont fournies pour gérer la compatibilité avec d'autres logiciels.

### FRAGMENTS A ELLIPSE



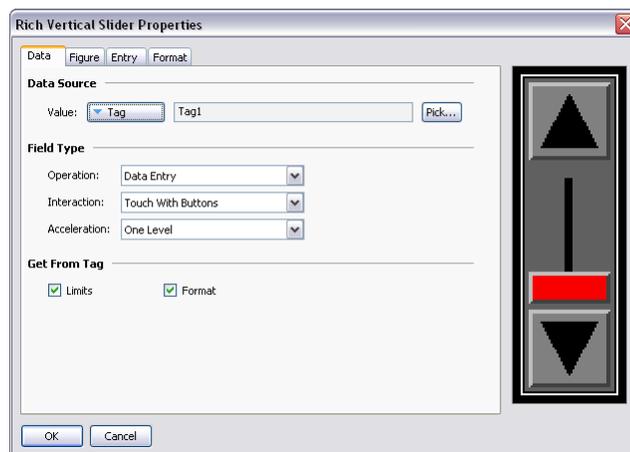
Ces primitives représentent le quart ou la moitié d'une ellipse. Leurs propriétés sont classiques.

### SLIDERS RICHES



Ces primitives permettent à une valeur de balise d'être ajustée au moyen d'un curseur analogique. Même si elles peuvent être utiles, elles sont susceptibles d'être remplacées par des primitives plus puissantes dans une prochaine version de Crimson et sont donc placées dans la sous-catégorie Legacy.

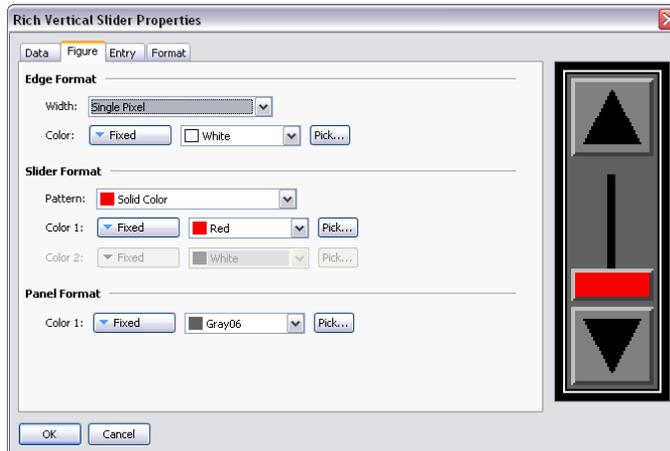
### PROPRIETES DE DONNEES



- La propriété *Valeur* spécifie les données dont la valeur doit être éditée.
- La propriété *Fonctionnement* est utilisée pour indiquer si la saisie de données doit être activée ou non. La valeur par défaut permet l'entrée, en lecture seule, de curseurs et peut induire en erreur.
- La propriété *Interaction* précise comment l'utilisateur interagit avec le primitif, que ce soit via les boutons poussoirs, en manipulant le curseur directement, ou par les deux méthodes.

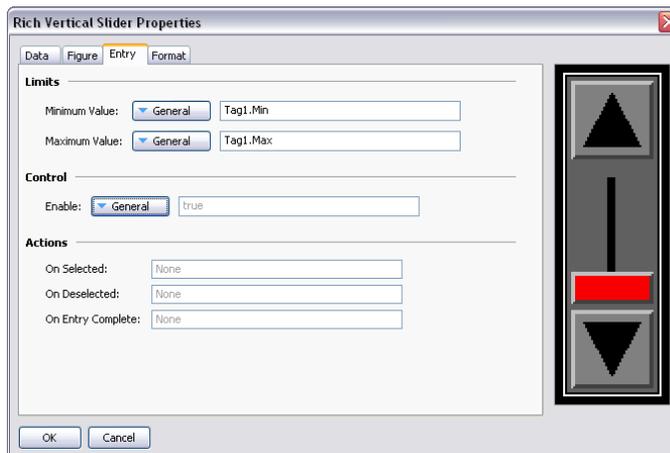
- La propriété *Accélération* spécifie le nombre de niveaux d'accélération fournis lors de la saisie des données. Elle permet de déplacer le curseur dans une accélération progressive plus rapide après un nombre approprié de mesure. Plus d'un niveau d'accélération peuvent entraîner des changements importants déployés par inadvertance.
- Les propriétés *Prendre depuis le tag* sont utilisées pour indiquer si les limites de curseur et de format des données seront obtenues à partir de l'étiquette fournie dans la propriété Valeur ou si elles seront saisies manuellement.

#### PROPRIETES DE FIGURE



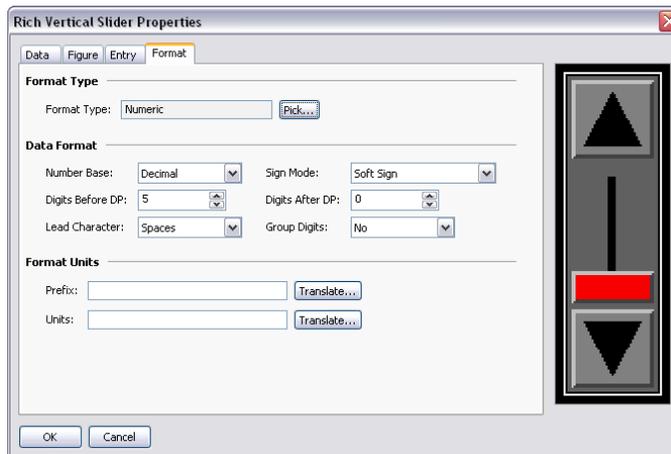
Reportez-vous au chapitre précédent pour les détails du remplissage standard et les paramètres de pointage.

#### PROPRIETES D'ENTREE



Reportez-vous au chapitre précédent pour le détail des Propriétés standard de données d'entrée.

## PROPRIETES DE FORMAT

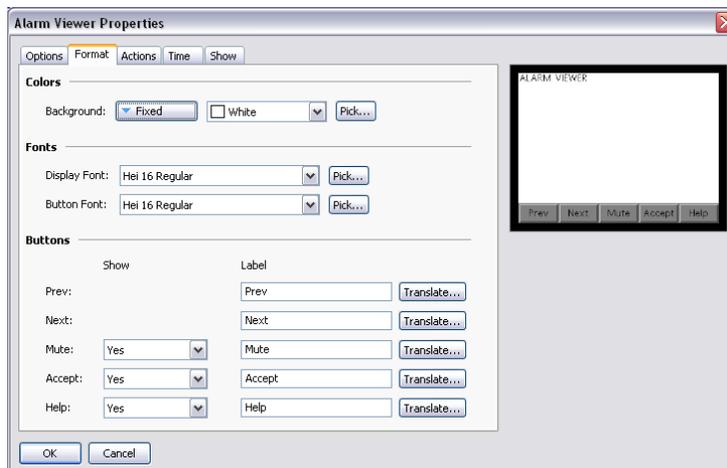


L'onglet Format définit le format de données utilisé par la primitive. Etant donné qu'elle n'affiche pas réellement les données, vous pouvez vous demander pourquoi elle est nécessaire : car elle permet d'aller plus vite. L'accélération de la saisie des données dépend de la connaissance la base de nombre des données en cours d'édition et de la position d'un point décimal. Les autres paramètres sont ignorés. Notez que la sélection du format peut ne pas être disponible si le format est obtenu à partir de la balise de contrôle.

## PRIMITIVES SYSTEME

### VISUALISATION DU FORMAT

La plupart des données Primitives Système d'affichage sont créées ou accessibles par Crimson. Chaque visualiseur se compose d'une surface d'affichage avec un certain nombre de boutons situés en dessous. L'apparition de la liste à la base du visualisateur est contrôlée via l'onglet Format de la boîte de dialogue de propriétés...

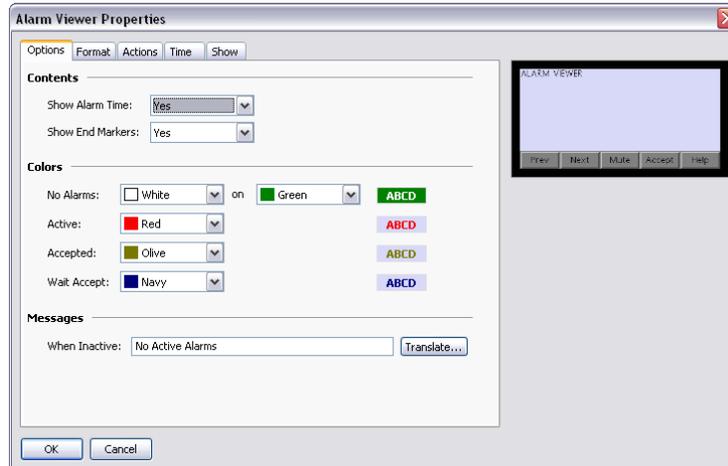


Les couleurs et les polices sont précisées de la façon conventionnelle. Les boutons permettent à certains des boutons situés en bas du visualiseur d'être édités, ou à leurs étiquettes d'être édités ou traduites pour les demandes internationales. N'oubliez pas que les chaînes à traduire peuvent être réglées sur des expressions, ce qui implique que l'étiquette sera placée sur un bouton personnalisé à l'exécution.

## ALARMES DE VISUALISATION

Les Alarmes de visualisation sont utilisées pour afficher et éventuellement accepter les alarmes du le système.

### OPTIONS DES PROPRIETES



- La propriété *Afficher heure alarme* est utilisée pour indiquer si chaque alarme devrait être précédée de la date et l'heure à laquelle elle s'est produite. Le format de l'heure exacte à utiliser est spécifié sur l'onglet Heure.
- La propriété *Afficher arqueur de fin* propriété est utilisée pour indiquer si les marqueurs doivent être inclus dans la liste des premier et dernier éléments, rendant ainsi plus facile la visualisation par l'utilisateur quand ils sont à chaque extrémité de la liste.
- Le groupe de propriétés *Couleurs* spécifie les couleurs du texte pouvant être utilisées lors de l'affichage des alarmes dans des États différents. Le message Aucune alarme permet d'afficher une couleur de fond consacrée, alors que l'état de diverses couleurs spécifiques utilise toujours le fond de la primitive elle-même.
- La propriété *Lorsque inactif* traduit la chaîne qui est affichée par la primitive en l'absence d'alarmes actives.

### PROPRIETES D'ACTION

Si le bouton Aide situé en bas de la visionneuse est activé via l'onglet Format, l'action aide s'exécute exécuter lorsque le bouton est pressé.

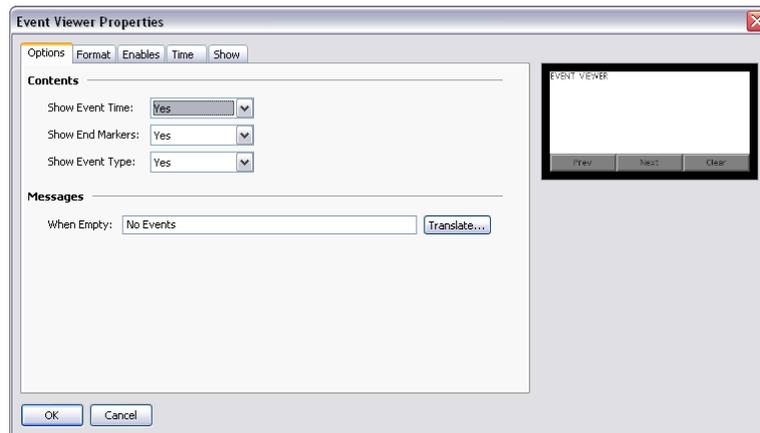
### PROPRIETES HEURE

L'onglet Heure définit le format à utiliser pour indiquer l'heure et la date à laquelle une alarme se produit. Reportez-vous au chapitre sur l'utilisation des formats pour des informations détaillées.

## VISUALISATION EVENEMENTS

La visualisation d'événements est utilisée pour visualiser et éventuellement effacer les événements enregistrés par le système en réponse à des alarmes ou des événements générés par les données de Tags.

## PROPRIETES DE L'OPTION



- La propriété *Afficher heure de l'événement* est utilisée pour indiquer si chaque événement doit être préfixé avec l'heure et la date à laquelle il s'est produit. Le format de l'heure exacte à utiliser est spécifié sur l'onglet Heure.
- La propriété *Afficher marqueurs de fin* est utilisée pour indiquer si les marqueurs doivent être inclus dans la liste pour le pavillon du premier et dernier éléments.
- La propriété *Afficher type d'événement* est utilisée pour indiquer si chaque entrée doit être marquée pour indiquer s'il s'agit d'un événement d'alarme, d'acceptation ou d'autorisation, ou si elle représente un événement simple. Si les alarmes sont en cours d'utilisation, l'activation par défaut de ce paramètre peut entraîner des confusions dans l'affichage.
- La propriété *Lorsue vide* définit ou traduit la chaîne qui est affichée par la primitive en l'absence de manifestations dans le journal.

## PROPRIETES D'ACTIVATION

Si le bouton Effacer situé en bas de la visionneuse est activé via l'onglet Format, la propriété *Activer Effacement* est utilisé pour activer ou désactiver le nettoyage du journal des événements.

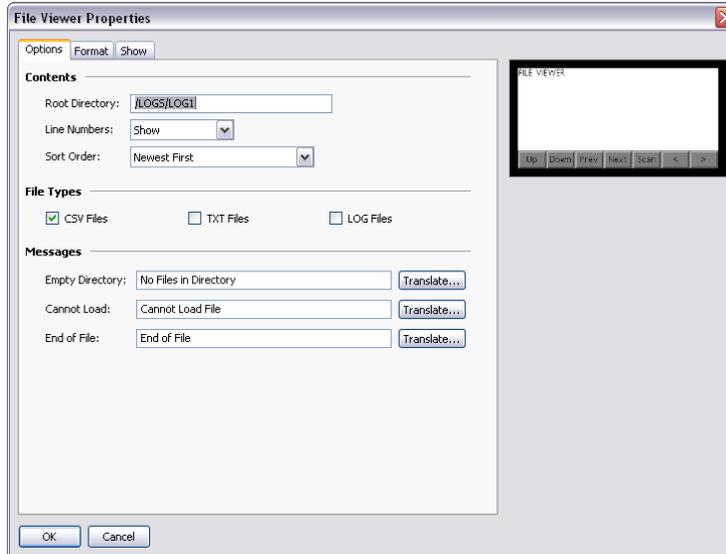
## PROPRIETES D'HEURE

L'onglet Heure spécifie le format à utiliser pour l'indication de la date et l'heure à laquelle un événement s'est produit. Reportez-vous au chapitre sur l'utilisation des formats pour plus de détails.

## FICHER DE VISUALISATION

Le Fichier de visualisation est utilisé pour permettre à l'utilisateur de visualiser des fichiers texte sur la carte CompactFlash.

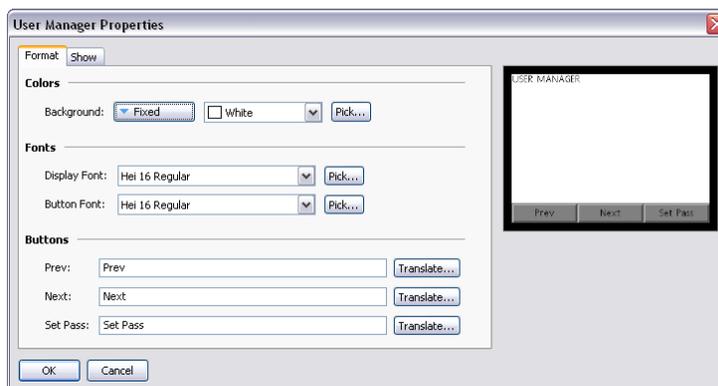
### PROPRIETES DE L'OPTION



- La propriété *Répertoire racine* spécifie le répertoire devant être affiché.
- La propriété *Numéros de ligne* est utilisée pour afficher ou masquer les numéros de ligne dans le fichier.
- La propriété *Ordre de tri* est utilisée pour indiquer comment les fichiers doivent être accessibles.
- La propriété *Types de fichier* groupe de propriétés est utilisée pour indiquer les types de fichier qui devrait être disponibles pour le visionnement. Notez que seuls les fichiers texte peuvent être affichés.
- La propriété *Messages* définit et traduit les divers messages utilisés par le Fichier Visualisation.

## GESTIONNAIRE UTILISATEURS

Gestionnaire des utilisateurs est utilisé pour permettre le changement de mots de passe lors de l'exécution

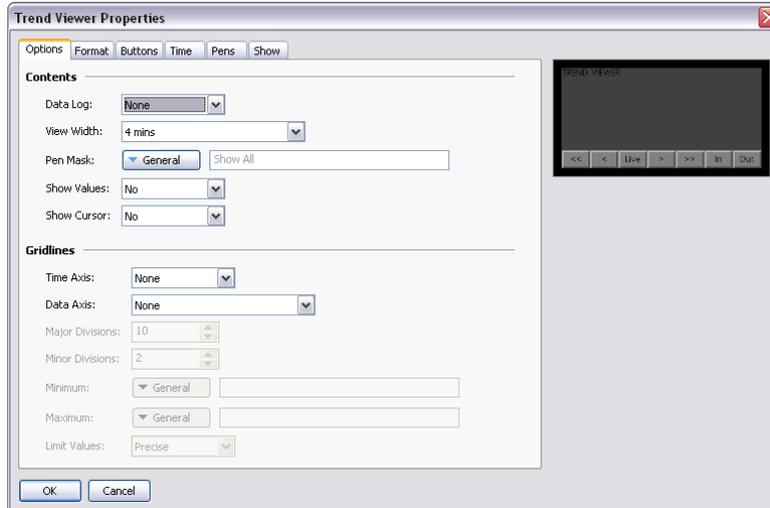


Ses propriétés fondamentales sont contenues sur un seul onglet, et tous sont classiques.

## VISUALISATEUR TENDANCES

Le Visualisateur de tendances permet l'affichage d'informations provenant des données du journal.

### PROPRIETES DE L'OPTION



- La propriété *Journal de données* sélectionne les données du journal pour être affichés.
- La propriété *Voir longueur* est utilisée pour indiquer le montant initial de données qui doivent figurer dans la fenêtre. L'utilisateur peut ensuite zoomer et dézoomer en utilisant les boutons au bas de la visionneuse.
- La propriété *Crayon masque* est utilisé pour fournir une valeur 32-bit entier permettant de désactiver l'affichage des canaux spécifiques. Le bit 0 correspond au premier canal de données du journal, le bit 1 au deuxième et ainsi de suite. Une valeur de bit affiche le canal tandis qu'une valeur nulle le cache. Une inscription vide lance le comportement par défaut, avec l'ensemble des chaînes affichées.
- La propriété *Afficher valeurs* active ou désactive l'affichage des données de Valeurs associées à chaque canal de données du journal, que ce soit en mode direct ou lors du défilement en arrière en utilisant le curseur.
- La propriété *Afficher curseur* est utilisé pour activer ou désactiver l'affichage d'un curseur sur le lecteur. Le curseur peut être activé par l'utilisateur pour permettre à un point spécifique à temps pour être déterminée avec précision, et éventuellement de permettre au Valeurs associées historiques de données à afficher.
- La propriété *Axe temporel* permet de définir si le quadrillage doit être affiché pour l'axe temporel. La hauteur du quadrillage est déterminée automatiquement par Crimson en fonction de la quantité de temps couverte par le spectateur.
- La propriété *Axe de données* est utilisée pour contrôler l'affichage du quadrillage de l'axe des données. Le quadrillage peut être défini manuellement en spécifiant simplement les principales divisions ou les deux divisions majeure et mineure,

ou peut être défini automatiquement en spécifiant des valeurs minimales et maximales pour l'axe des données et en laissant Crimson calculer le meilleur modèle de grille.

- Les propriétés de *Grandes divisions* et de *Divisions mineures* définissent le nombre de divisions à effectuer lors de la définition manuelle du quadrillage.
- Les propriétés *Minimum* et *Maximum* sont utilisées pour indiquer la plage de données qui sera affiché en utilisant le quadrillage automatique. Crimson utilisera ces valeurs pour déterminer le meilleur modèle de grille à adopter. Les valeurs de données seront également réduites à ces valeurs, et non adaptées à leurs propres limites de données.
- La propriété *Limites des valeurs* spécifie comment les valeurs haut et en bas de l'échelle sont déterminés. Si un réglage de précision n'est pas spécifié, les valeurs minimales et maximales seront utilisés exactement, même si elles dépassent des limites qui ne correspondent pas exactement à l'espacement autorisé. Cela peut produire un espacement irrégulier au sein de la grille. Un réglage de l'arrondi permet une mise à l'échelle de la primitive ajustant les limites pour atteindre les graduations régulièrement espacées.

#### PROPRIETES DE FORMAT

Ces propriétés sont utilisées pour spécifier les couleurs et les polices. Leur fonctionnement est classique.

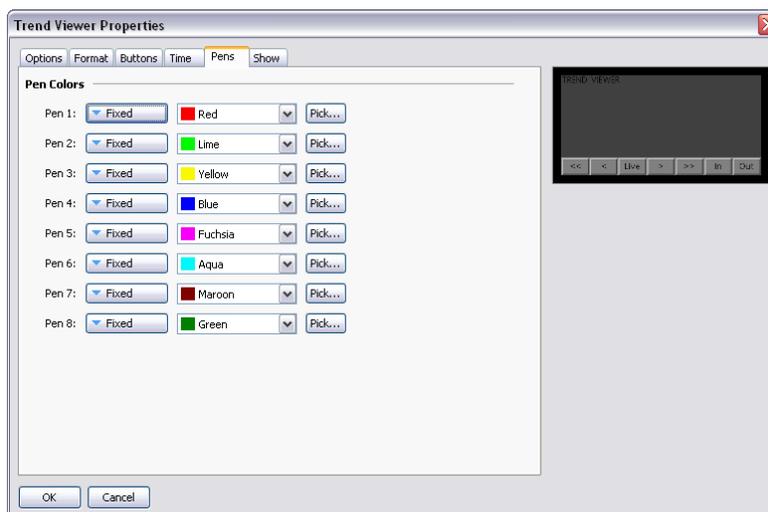
#### PROPRIETES DES BOUTONS

Ces propriétés sont utilisées pour modifier et éventuellement traduire les légendes des boutons .

#### PROPRIETES D'HEURE

L'onglet heure est utilisé pour fournir heure et date des informations relatives aux données du journal. Reportez-vous au chapitre sur l'utilisation des formats pour des informations détaillées.

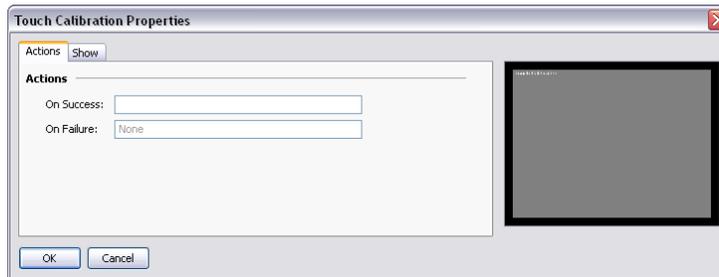
#### PROPRIETES DES CRAYONS



Ces propriétés sont utilisées pour spécifier huit couleurs qui seront utilisées pour l'élaboration des données. Les couleurs sont utilisées de manière cyclique. Il n'est pas recommandé de dessiner un nombre si important de chaînes.

### STIMULATION DE LA CALIBRATION

Cette primitive est utilisée pour calibrer l'écran tactile...



Cette primitive permet de définir des actions à prendre lorsque l'étalonnage a réussi ou échoué. Ces propriétés sont généralement configurés pour revenir à un écran de menus ou se déplacer autrement hors de la page d'étalonnage.

### TEST STIMULATION

La primitive test de simulation permet à l'utilisateur de vérifier les performances de l'écran tactile et d'étalonnage. Chaque contact produit un point sur l'écran, avec un chemin d'affichage. Il n'existe pas de propriété de configuration au-delà du contrôle de la visibilité.

# LOCALISATION

Crimson 3 prend en charge un certain nombre de fonctionnalités qui vous permettent d'adapter votre base de données pour un déploiement dans des environnements multilingues. Ce chapitre décrit comment ces fonctionnalités sont utilisées, et comment vous pouvez facilement créer des bases de données qui peuvent être utilisées à travers le monde.

## SELECTION DES LANGUES

La première étape dans la création d'une base de données multilingue consiste à configurer les langues à utiliser dans votre projet. En appuyant sur le bouton de traduction sur la page Configurer de l'interface utilisateur vous afficherez la boîte de dialogue suivante...

Language	Code	Numeric Format	Diacritical Marks	Switch Keyboard
System Locale	en	English - US	Language Default	Never
French - France	fr	English - US	Language Default	Never
German - Germany	de	English - US	Language Default	Never
Simplified Chinese	zh-cn	English - US	Language Default	In Translate
Generic	None	English - US	Language Default	Never
Generic	None	English - US	Language Default	Never
Generic	None	English - US	Language Default	Never
Generic	None	English - US	Language Default	Never
Generic	None	English - US	Language Default	Never
Generic	None	English - US	Language Default	Never

**Auto-Translation**  
 Method: System Lexicon Then Web Service  
 Service: Google WebAPIs

**Options**  
 Apply language setting globally and not just to user interface.

OK Cancel

La section supérieure de la boîte de dialogue permet de définir un certain nombre de propriétés pour chaque langue...

- La propriété *Langue* est utilisée pour sélectionner la langue souhaitée. Une langue peut exister en plusieurs variantes selon les différents pays où elle est parlée. Le paramètre générique peut être utilisé pour les langues qui ne sont pas directement prises en charge au sein de Crimson.
- La propriété *Code* est utilisée pour afficher ou entrer le code à deux caractères de la langue qui a été sélectionnée. Cette propriété sera transmise aux services de traduction web lors de la traduction automatique, et sera utilisée pour définir la ligne de titre dans un fichier lexique. Vous devez entrer le code manuellement pour les langues générique, dont Crimson n'a pas de connaissances préalables.
- La propriété *Format numérique* est utilisée pour définir si Crimson va formater les nombres en utilisant le format États-Unis ou en utilisant un format spécifique à la sélection de la langue courante. Les options numériques de formatage comprennent l'utilisation d'une virgule par rapport à des points, et l'emplacement des caractères de groupement des chiffres.

- La propriété *Marques diacritiques* est utilisée pour remplacer le réglage par défaut d'une langue pour le traitement des accents sur les caractères en majuscules. Par exemple, le français tel qu'il est utilisé en France (contrairement au Canada) applique des accents aux majuscules, ce qui peut entraîner des problèmes de rendu pour certaines polices. La sélection des lettres minuscules seulement pour ce paramètre remplace ce comportement par défaut.
- La propriété *Changer clavier* est utilisée pour sélectionner les circonstances dans lesquelles le logiciel de configuration Crimson changera la configuration du clavier pour passer à celui utilisé par la langue. Le changement peut se produire lorsque vous utilisez la Boîte de dialogue traduction, chaque fois que les textes sont en cours d'édition dans cette langue, ou jamais. Le changement de clavier dans la fenêtre de traduction est activé par défaut pour les langues comme le chinois simplifié.

Les commandes de la prochaine rubrique concernent la traduction automatique. La dernière propriété permet de spécifier si le paramètre de langue actuelle est appliqué aux services tels que serveur web et Journal de données, ou si ceux-ci devraient toujours utiliser la langue par défaut du système.

## CONFIGURATION DE LA TRADUCTION AUTOMATIQUE

Crimson contient de puissantes fonctions de traduction automatique pour vous aider à adapter votre base de données à un déploiement international. Le module de traduction automatique se compose de deux éléments, un lexique et d'un service de traduction basé sur le web.

Le lexique système est un fichier texte Unicode qui contient un grand nombre de mots et des phrases normalisés qui sont utilisés dans l'automatisation industrielle et le contrôle de processus, traduits dans un certain nombre de langues communes. Ce lexique peut être consulté pendant le processus de traduction, ce qui permet de traduire les expressions les plus fréquemment rencontrées avec un degré élevé de précision.

La partie basée web utilise l'un des deux services. Google WebAPIs propose généralement des traductions plus rapidement, car elle n'est pas soumise à des restrictions de bande passante. En revanche, Microsoft Translator fournit des traductions plus précises, mais n'est pas aussi rapide car il ya une limite sur le nombre de présentations qui peuvent être faites par minute. Vous pouvez sélectionner le service la traduction dans la boîte de dialogue Configurer.

La traduction automatique peut être configuré pour utiliser l'une ou l'autre de ces méthodes. Si vous avez une connexion Internet, il est généralement préférable d'utiliser le lexique, puis le service en ligne. Le lexique peut être utilisé seul dans certains cas pour éviter les traductions douteuses que les services sur le web peuvent parfois fournir.

## TRADUIRE VOTRE BASE DE DONNEES

La traduction d'une base de données peut être accomplie d'un certain nombre de façons.

## SAISIE DES TRADUCTIONS

La traduction manuelle est effectuée en appuyant sur le bouton Traduire correspondant à chaque chaîne dans la base. Une boîte de dialogue s'affiche, permettant de consigner les textes traduits ou de lancer une traduction automatique pour cette seule chaîne...



La traduction auto locale vous permet de revoir les traductions pour plus de précision.

## TRADUCTION AUTO GLOBALE

Le sous-menu Utilitaires dans le menu Fichier contient une commande permettant d'appliquer la traduction auto à chaque chaîne de la base de données. Cette commande peut prendre un certain temps à s'exécuter, en particulier si un service de traduction limitant la bande passante est utilisé. Certaines précautions doivent être prises lors de l'utilisation de l'outil de traduction, concernant par exemple des chaînes qui ne sont pas dans le lexique système et contenant des termes techniques.

## EXPORTATION ET IMPORTATION

Le sous-menu Utilitaires contient également les commandes d'exportation et d'importation de fichiers texte contenant tout le texte à traduire dans la base. Ces fichiers peuvent être édités en utilisant une application telle que Microsoft Excel, permettant de saisir manuellement la traduction. Cette facilité est particulièrement utile lorsque vous travaillez avec un service de traduction tierce partie. Le format de fichier comporte plusieurs colonnes qui indiquent la source de chaque chaîne, permettant à des événements distincts d'une chaîne donnée d'être traduit en texte différent selon le contexte.

## APPLIQUER UN LEXIQUE

En plus du lexique système décrit ci-dessus, vous pouvez créer vos propres lexiques, soit à partir de zéro soit en utilisant la commande Exporter dans le menu : Utilitaires Lexique. Les fichiers lexique sont des fichiers texte Unicode qui commencent par une rangée d'en-tête contenant le codes de langue. Après la ligne d'en-tête, chaque ligne contient un mot ou une expression dans chacune des langues définies.

Un exemple de fichier lexique est présenté ci-dessous...

en	fr	de
one	un	eins
two	deux	zwei
three	trois	drie

Notez que le texte doit être inscrit en lettres minuscules sauf si un terme spécifique est toujours utilisé en majuscules, comme cela peut être le cas avec un nom allemand. L'utilisation de minuscules permet à Crimson de former ses propres majuscules et les variantes de casse de titre.

## PREVISUALISATION DES TRADUCTIONS

Les traductions peuvent être prévisualisées dans l'éditeur graphique en sélectionnant la langue correspondante dans le menu déroulant qui est accessible par l'icône drapeau dans la barre d'outils. Toute modification directe du texte s'appliquera également à la langue sélectionnée, les autres langues étant laissées inchangées. L'édition dans les boîtes de dialogue continue d'être limitée à la langue par défaut, les autres langues étant accessibles via le bouton Traduire, comme d'habitude.

## BASCULER ENTRE LES LANGUES

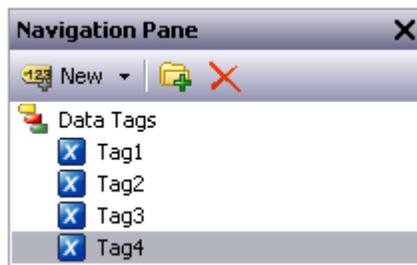
Le langage utilisé par le périphérique cible est contrôlé via des appels à la fonction `SetLanguage ()`, l'argument de la fonction étant un nombre compris entre 0 et 9 pour sélectionner l'option désirée. Par exemple, un appel à `SetLanguage (1)` dans l'exemple ci-dessus va sélectionner le français, tandis qu'une action personnalisée de `SetLanguage (2)` sélectionnera l'allemand. La fonction `GetLanguage ()` peut être utilisée pour déterminer le langage en cours.

# UTILISATION DES WIDGETS

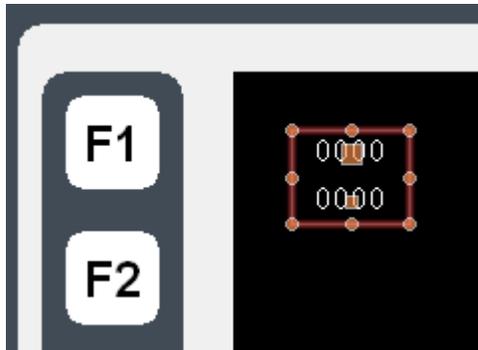
Crimson 3 supporte désormais une nouvelle fonctionnalité : la possibilité de transformer des groupes ordinaires de primitives en entités puissantes appelées widgets. En plus de ses composants primitifs, un widget contient des éléments de données définissables par l'utilisateur, qui peuvent être modifiés au niveau du groupe mais référencés par les composants du widget. Ce chapitre explique comment créer des widgets, et comment les utiliser.

## CREATION D'UN WIDGET

La meilleure façon de comprendre les widgets est d'en créer un. Commençons par créer une base de données vide et ajoutons quatre étiquettes numériques. Laissez les propriétés des tags sur leurs paramètres par défaut, pour aboutir à quatre valeurs entières nommées Tag1 à Tag4.



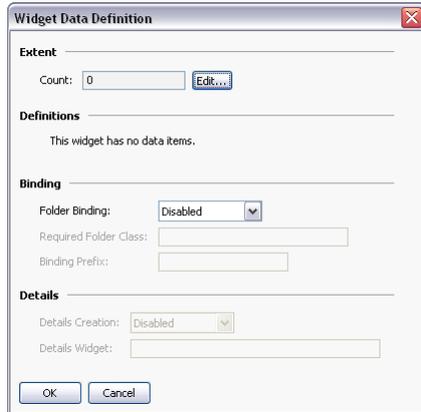
Basculez vers la section Pages d'affichage et ajoutez deux boîtes de données primitives à la page...



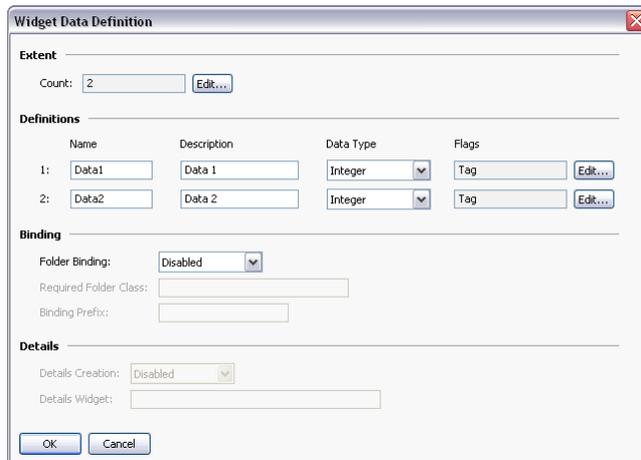
Laissez les propriétés sur leurs valeurs par défaut pour le moment, et sélectionnez les deux points. Faites un clic droit sur la sélection, et sélectionnez la commande Widgetize à partir du menu contextuel. Les éléments seront liés à un groupe, mais la boîte de dialogue suivante apparaîtra également...



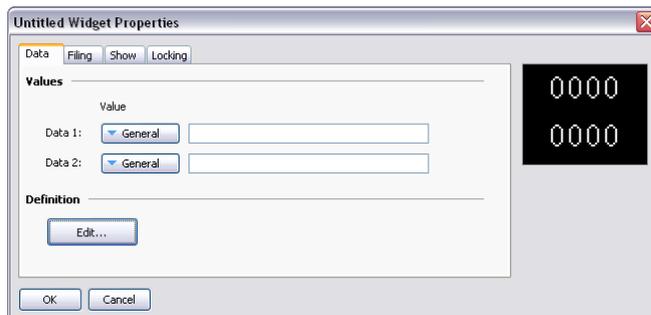
Une fois le widget créé, cette boîte de dialogue sera utilisée pour éditer des éléments de données de ce dernier, mais pour l'instant nous n'avons rien défini. Cliquez sur le bouton Modifier dans la section Définitions pour définir certains éléments de données...



En cliquant sur le bouton Modifier à côté du champ de Décompte nous créons deux propriétés...



Remplissez les champs de données comme indiqué ci-dessus, en accordant une attention particulière à trouver le bon type de données, et de modifier les champs des drapeaux pour indiquer que chaque élément de données doit être une balise. (Le champ de drapeaux peuvent être édité en utilisant le bouton Modifier à côté de la propriété.) Appuyez sur OK pour fermer la boîte de dialogue, et notez comment le widget lui-même affiche maintenant les éléments de données dans sa boîte de dialogue des propriétés propres...

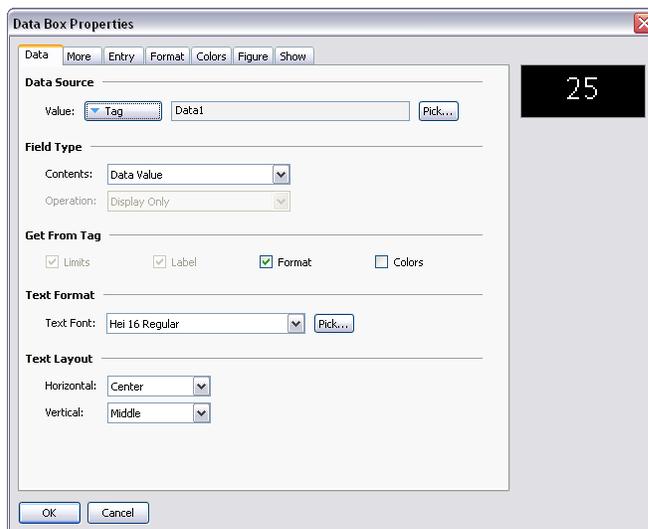


Ignorer ces informations pour le moment, et appuyez sur OK pour fermer cette boîte de dialogue.

Le widget devrait encore être sélectionné dans l'éditeur graphique. Cliquez alors sur une des cases de données contenues dans le widget pour entrer en mode d'édition du groupe. N'oubliez pas que le rectangle vert marque le groupe que nous venons de modifier, et le rectangle rouge montre l'élément sélectionné dans ce groupe...



Double-cliquez sur la case de données pour afficher ses propriétés...

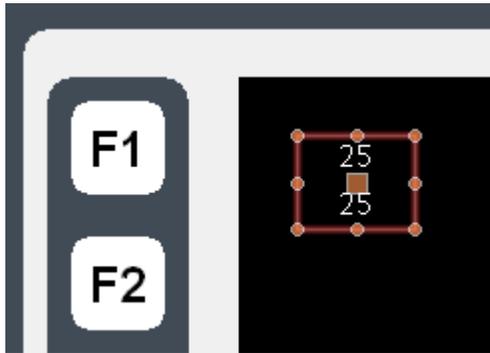


Saisissez `Données1` dans le champ Valeur, puis notez les résultats.

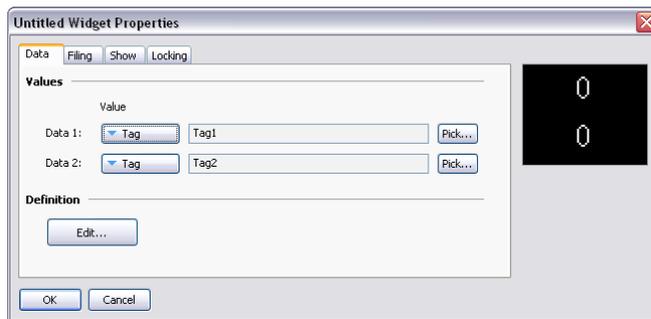
Crimson accepte ce nom en tant que nom de la balise, même si nous n'avons pas réellement une balise nommée `Données1` dans notre base de données. Cette valeur est en fait égale à l'un des éléments de données définis dans le widget, et représentera n'importe quelle balise que nous assignerons, lorsque nous modifierons les données widget. (La valeur de 25 affichée dans la fenêtre de prévisualisation est la valeur par défaut utilisée pour les widgets qui ne sont liés à rien.) Etant donné que `Données1` est marqué par un tag, on peut accéder à ses propriétés, l'utiliser comme source d'informations de formatage, ou faire tout ce que l'on pourrait effectuer avec un tag.

Répétez cette étape pour la deuxième case de données, cette fois, en définissant sa propriété Valeur sur `Data2`.

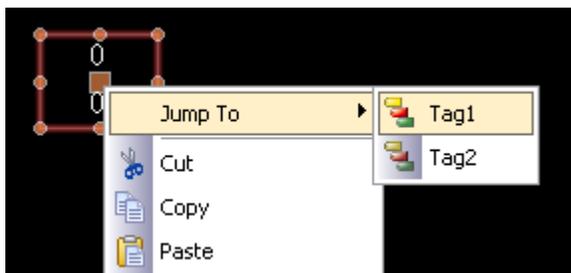
Appuyez sur **ECHAP** jusqu'à ce que seul le widget soit sélectionné. Si vous allez trop loin et effacez la sélection, cliquez simplement sur le widget lui-même, en s'assurant qu'il dispose d'un rectangle rouge autour de lui...



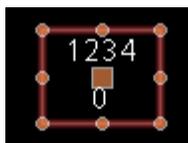
Maintenant affichez les propriétés du widget, et cette fois saisissez les valeurs des éléments de données...



Saisissez les valeurs indiquées ci-dessus, fixant les éléments de données à `Tag1` et `Tag2` respectivement. Notez comment la prévisualisation affiche maintenant les valeurs de zéro, que les boîtes de données au sein du widget reçoivent désormais leurs données à partir `Tag1` et `Tag2` respectivement. Pour rendre les choses un peu plus intéressantes, faites un clic droit sur le widget et accédez au menu **Aller à...**



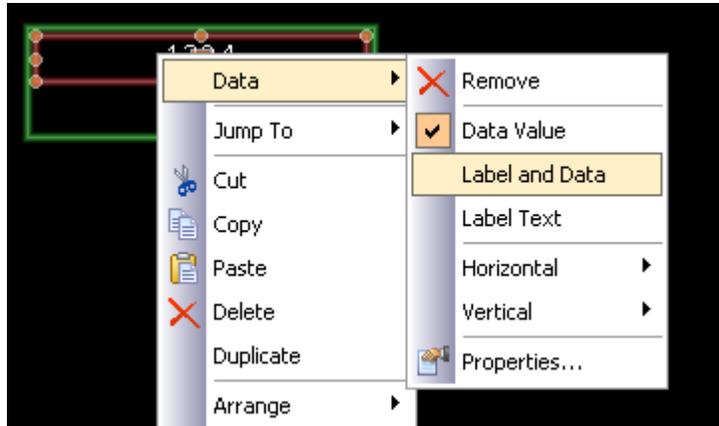
Sélectionnez `Tag1` pour accéder à cette balise, et entrez la valeur `1234` dans la propriété **Simuler comme**. Utilisez la combinaison **ALT + GAUCHE** ou le bouton **Précédent** sur la barre d'outils, et notez comment le widget continue de suivre les données du tag...



Ensuite, saisissez la poignée droite et élargissez le widget...



Effectuez un clic gauche sur le dessus des deux boîtes de données pour entrer en mode d'édition du groupe, puis cliquez à droite sur la même boîte pour accéder à son menu contextuel...



Sélectionnez le sous-menu Données, et choisissez la commande légende pour configurer cette boîte de données et afficher le nom de la balise, ainsi que ses données de valeur. Notez la nouvelle apparence du widget...



Comme vous pouvez le constater, la boîte de données est l'affichage des étiquettes à partir `Tag1`, indiquant que la valeur de `Données1` que nous avons saisie est tout à fait équivalente à la balise sur laquelle l'élément de données a ensuite été configuré.

## RESUME

Récapitulons maintenant ce que nous avons fait...

- Nous avons placé des primitives sur l'écran et les avons regroupées en un type spécial de groupe appelé widget. Le widget se comporte apparemment comme un groupe normal mais il a des propriétés supplémentaires.
- Nous avons modifié les définitions données pour le widget, en créant deux éléments de données. Chacun a reçu un nom, une description, un type de données et un certain nombre de drapeaux.
- Nous avons utilisé l'édition de groupe pour modifier le contenu du widget, définissant les propriétés selon des éléments propres au widget, se référant à leur disposition via les noms d'éléments de données.

- Nous avons modifié les éléments de données du widget, les liant à des tags, permettant ainsi la création de tags réels et de leurs informations associées au contenu de notre widget.

### POURQUOI CETTE QUESTION EST-ELLE IMPORTANTE

Alors pourquoi les widgets sont-ils si importants ? Nous pourrions facilement avoir créé les boîtes de données et lié directement ces dernières à des balises, pourquoi donc s'embêter avec ces étapes supplémentaires ? Les réponses deviennent évidentes lorsque vous essayez de créer des widgets plus complexes...

- Les widgets permettent d'utiliser à plusieurs endroits les éléments de données, avec de multiples éléments dépendant de la même balise, sans que vous ayez à sélectionner le nom du tag à plusieurs endroits.
- Les widgets peuvent encapsuler des fonctionnalités complexes, et vous permettront de reproduire et de réutiliser ces dernières au sein des bases de données. En effet, ils permettent à l'utilisateur de créer des primitives complexes.
- Widgets peuvent être sauvegardés sur le disque et ajouté au Panneau de ressources, ou distribués par courrier électronique, facilitant ainsi la coopération entre les utilisateurs Crimson ou entre les utilisateurs et le support technique.

### PLUS DE DETAILS

La section suivante examine la plupart des sujets ci-dessus, mais plus en détail.

Elle explore aussi certaines des techniques qui permettent de créer des widgets encore plus puissants.

## DEFINITIONS DES DONNEES DES WIDGETS

Les éléments de données donnent aux widgets toute leur puissance. La définition des données d'un widget est éditée à l'ouverture des propriétés de ce dernier et en cliquant sur le bouton Modifier dans la section Définitions de la page de données...

	Name	Description	Data Type	Flags
1:	Data1	Data 1	Integer	Tag
2:	Data2	Data 2	Integer	Tag

- La propriété *Etendre* définit la manière dont de nombreux éléments de données sont nécessaires pour ce widget. Cette valeur peut être modifiée à tout moment, mais la réduire entraînera une perte des éléments de données. Jusqu'à quatre-vingts articles de données peuvent être définis.

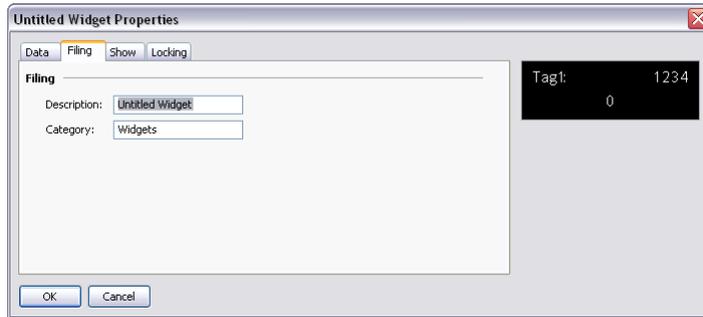
- La propriété *Nom* de chaque élément de données est utilisée pour faire référence à cet élément à partir des primitives contenues dans le widget. Elle doit donc remplir toutes les exigences d'un nom de balise. Elle ne doit contenir aucun espace ni aucune ponctuation, et doit commencer par une lettre.
- La propriété *Description* de chaque élément de données est utilisée pour fournir une version plus conviviale de ce nom, cette fois pour l'affichage dans la boîte de dialogue Élément de données d'édition. Il n'existe pas de restrictions sur le contenu de ce champ.
- La propriété *Type de données* définit pour chaque article le type de données requises. La manière dont l'élément de données est affiché dans la boîte de dialogue des propriétés du widget dépendra du paramètre qui est sélectionné. L'entier réel, et les types de données de chaîne correspondent à des valeurs d'expression, tandis que la couleur, la page et types de données permettent de créer des éléments d'action plus complexes. Page et points d'action peuvent être traités comme des noms de page d'affichage et programmes à partir des primitives du widget.
- La propriété de *Drapeaux* de chaque éléments de données est utilisée pour modifier les articles qui possèdent les types de données d'un nombre entier réel, ou d'une ligne. Elle prend en charge les paramètres suivants...

PARAMÈTRE	DESCRIPTION
Tag	La valeur entrée pour l'élément de données doit être une balise. Les primitives présentes dans le widget permettent de traiter l'élément de données comme un tag, et d'accéder à ses propriétés, au format des données et ainsi de suite.
Inscriptible	La valeur entrée pour l'élément de données doit être ouverte en écriture. Les primitives situées dans le widget sont de même autorisées à écrire dans l'élément de données.
Array	La valeur entrée pour l'élément de données doit être le nom d'un tableau. Les primitives situées dans le widget peuvent voir l'article de données dans un tableau, et doivent utiliser l'opérateur d'index pour accéder aux valeurs individuelles.
Element	La valeur entrée pour l'élément de données doit être un élément de tableau. Les primitives situées dans le widget peuvent voir l'article de données en tant qu'élément, et seront en mesure de le transmettre à des fonctions qui requièrent des arguments de ce type.
N Bind	Crimson n'appliquera pas le dossier de liaison à cette propriété, lui permettant d'être utilisée pour stocker des valeurs prédéfinies sans erreurs générés. Voir les articles concernant les options des dossiers de liaison.

- La propriété de *Liaison* est utilisée pour contrôler une fonction avancée appelé dossier liaison. Elle est examinée en détail ci-dessous.
- La propriété *Détails* est utilisée pour contrôler une fonction avancée de création de pages. Elle est examinée en détail ci-dessous.

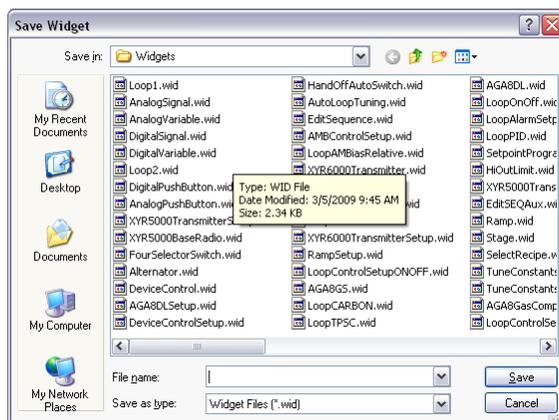
## REPLISSAGE WIDGETS

Chaque widget dispose d'un onglet Remplissage dans sa boîte de dialogue de propriété...



Les propriétés *Description* et *Catégorie* sont utilisées pour contrôler la façon dont un widget sera affiché sur le Panneau de ressources une fois enregistré. Tous les widgets de la même catégorie seront regroupés dans la même sous-catégorie de la catégorie Primitives, et la description du widget affichée quand l'utilisateur pointerait un élément.

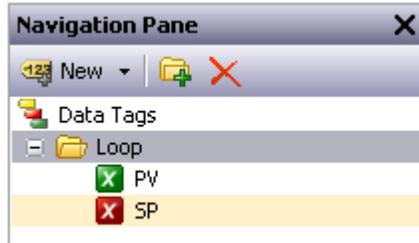
Pour enregistrer un widget, il suffit de le sélectionner et de choisir la commande Enregistrer Widget à partir du menu Edition ou d'appuyer sur les touches **CTRL + Q**. Une boîte de dialogue Enregistrer sous standard s'ouvre, vous permettant d'enregistrer le widget comme un fichier *wid* dans le répertoire widgets de Crimson...



Le Panneau de Ressources se mettra à jour automatiquement chaque fois qu'un fichier widget y sera ajouté. Cela se produira si le changement se fait via Crimson ou en déposant simplement un fichier dans le répertoire *wid* via Windows Explorer. Notez que les fichiers widget sont autonomes et peuvent être transférés entre installations Crimson sur différentes machines. Cette fonctionnalité fournit un mécanisme puissant pour le partage des éléments d'interface utilisateur, ou pour échanger des objets avec d'autres ingénieurs lorsque plusieurs personnes travaillent sur un projet.

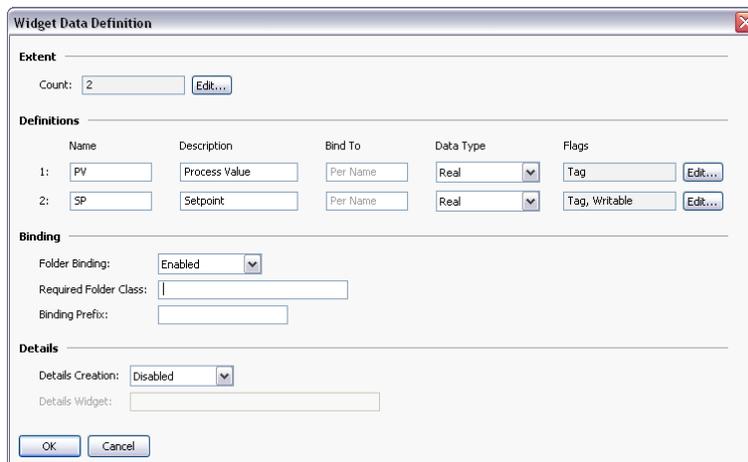
## LIER LES DOSSIERS

Crimson possède la capacité d'organiser les tags dans des dossiers dans une sorte de conception orientée objet à travers laquelle les balises qui représentent les propriétés d'un objet peuvent être regroupées dans un dossier qui représente l'objet lui-même. Prenons l'exemple ci-dessous...



Ici, un dossier a été créé pour représenter une boucle PID, et des balises ont été créées pour se référer à la valeur du processus de la boucle et la consigne. Les balises sont visées dans le code comme `Loop.PV` et `Loop.SP`, en utilisant les règles standard pour l'utilisation d'éléments imbriqués.

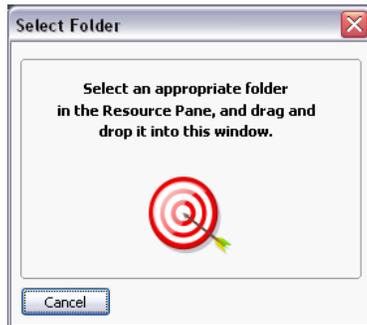
Le dossier de liaison vous permet de créer un widget qui reflète cet objet et la structure de la propriété que vous avez créée dans vos balises. Regardez les définitions des données suivantes...



Ici, nous avons créé des éléments de données dont les noms correspondent aux noms des balises qui forment une boucle PID. Nous avons fourni des noms lisibles pour eux, et nous avons marqué deux points de données sous forme de balises. Nous avons également défini la consigne accessible en écriture. Notez à ce moment qu'une nouvelle propriété appelée *Lier* à est apparue pour chaque élément de données.

Dans la section de liaison, nous avons activé un dossier de liaison. Cela indique que nous voulons que Crimson supporte la liaison automatique de tous les éléments de données à des balises à partir d'un dossier source unique. Après avoir sauvegardés ces changements et sélectionné le menu contextuel du widget, on remarquera une nouvelle commande appelée *Lier widget*.

Sélectionnez la commande ou en appuyant sur **CTRL + B** pour afficher la boîte de dialogue suivante...



Si vous faites glisser le dossier Boucle depuis le Panneau de Ressources et le déposez sur la cible, les éléments de données du widget seront automatiquement liés aux balises correspondantes dans le dossier.

L'ouverture des propriétés du widget affiche les résultats suivants...



En d'autres termes, chaque élément de données a été lié à la balise dans le dossier sélectionné qui a un nom identique à son élément de données. Pensez une seconde à la puissance de cette image : Vous pouvez définir plusieurs propriétés et les consolider en une seule opération, ce qui réduit les délais de conception et permet une meilleure réutilisation des articles préalablement conçus.

## LIAISONS AVANCEES

Le dossier de liaison prend en charge un certain nombre d'options avancées.

### CORRESPONDANCE DE CLASSE

Le premier et le plus simple des paramètres est le *Dossier de classe de paramètres* situé dans les propriétés du widget. Celui-ci peut être utilisé pour lier les dossiers qui seront acceptés, évitant par conséquent les incohérences entre différents types d'objets. La classe spécifiée sur le widget doit correspondre à la classe sur le dossier qui est lié, ou une erreur se produit.

### LIER PREFIXES

La propriété *Liaison de préfixe* peut être utilisée pour permettre aux widgets enfants à être liés à des sous-dossiers du dossier dans lequel le widget parent est lié. Par exemple, supposons que vous créez un widget-double boucle qui doit être lié à un dossier qui contient deux dossiers nommés PID Loop1 et Loop2. En réglant chacun des préfixes qui lie le widget enfant à l'un des noms de boucle, vous pouvez vous assurer qu'ils sont liés à des dossiers enfant

différents du dossier qui est traîné sur le widget parent. Par exemple, si le widget premier enfant a un préfixe contraignant de `Loop1` et sa société mère est lié à un dossier appelé `Dual`, les propriétés du widget enfant seront liées aux expressions `Dual.Loop1.PV` et `Dual.Loop1.SP` respectivement.

### UTILISER LES LIAISONS POUR

La propriété *Lier à* d'un élément de données peut être utilisée pour modifier l'expression à laquelle cet élément de données est lié. L'option la plus simple consiste à entrer un nom différent du nom de l'élément de données, auquel cas ce nom sera utilisé pour sélectionner la balise à laquelle se lier.

### UTILISER LES POINTS

Vous pouvez également saisir un nom qui contient des points. Ces derniers sélectionnent des tags dans les dossiers enfants du dossier source. Par exemple, entrer `Remote.SP` liera l'élément de données en question à une expression de `Loop.Remote.SP` au moment de la liaison dans le dossier en boucle.

### UTILISER LES CARETS

Pour remonter l'arborescence des dossiers, vous pouvez préfixer le nom avec des caractères caret, dont chacun se déplace d'un niveau. Un élément de données avec une liaison vers fixation de `^Name` dans un widget qui est lié à `Dual.Loop` sera lui-même lié à l'expression de `Dual.Name`.

### NOM SPECIAL

Vous pouvez également utiliser l'option *Liaison spéciale* pour les noms...

NOM	RÉSULTAT
<code>::Path</code>	Le chemin complet de la balise à laquelle ce widget est lié, y compris les dossiers parents.
<code>::Name</code>	Le nom de la balise à laquelle ce widget est lié, à l'exclusion des dossiers parents.
<code>::TopPath</code>	Le chemin complet de la marque pour laquelle le widget racine a été liée dans une opération de liaison imbriqués.Équivalent à <code>::Path</code> pour les non-imbriqués liés.
<code>::TopName</code>	Le nom de la balise à laquelle le widget racine a été lié dans une opération de liaison imbriquée. Équivalent à <code>::Name</code> .

Notez que chacun de ces noms spéciaux est équivalent à une chaîne constante égale à la dénomination nécessaires et non pas à la balise elle-même. Ils sont généralement utilisés pour fournir des informations aux utilisateurs sur le dossier dans lequel un widget ou son widget racine ont été liés.

## WIDGETS DE DÉTAILS

Supposons que vous avez créé un widget PID, mais que vous voulez afficher des informations plus détaillées d'état lorsque l'utilisateur appuie sur un bouton dans ce dernier. La réponse facile est de créer un widget de plus, peut-être plus complexe, et de le lier à la même boucle. Vous placez alors ce widget sur une autre page, puis sélectionnez cette page à partir du

widget aperçu original, en utilisant peut-être un élément de données à dire le widget de la page à utiliser.

Eh bien, la fonction de création Détails widget effectue toutes ces étapes automatiquement!

### ACTIVER LA CREATION DE DETAILS

Cette fonctionnalité est contrôlée via la propriété *Création Détails* des données du widget...

**Details**

Details Creation:  ▼

Details Widget:

La propriété *Détails widget* est utilisée pour fournir une liste séparée par des virgules de widgets que vous souhaitez placer sur leurs propres pages. Chaque widget est spécifié en donnant le nom de fichier vers lequel il a été sauvé. Dans l'exemple ci-dessus, nous avons un widget extrait d'un fichier appelé `PIDDetails.wid` dans le répertoire widgets de Crimson.

### DEFINITION DES ELEMENTS DE DONNEES

Nous devons aussi fournir des éléments de données dans le widget aperçu afin de pouvoir accéder aux noms des pages qui sont créés pour les widgets de détails. Ces propriétés doivent être nommés `details1`, `Details2`, et ainsi de suite, avec les données d'un point pour chaque élément de la liste Widgets Détails. Dans l'exemple ci-dessous, nous avons créé une telle propriété contenant le nom de notre page d'informations unique...

**Widget Data Definition**

**Extent**

Count:

**Definitions**

	Name	Description	Bind To	Data Type	Flags	
1:	<input type="text" value="PV"/>	<input type="text" value="PV"/>	<input type="text" value="Per Name"/>	<input type="text" value="Real"/> ▼	<input type="text" value="None"/>	<input type="button" value="Edit..."/>
2:	<input type="text" value="SP"/>	<input type="text" value="SP"/>	<input type="text" value="Per Name"/>	<input type="text" value="Real"/> ▼	<input type="text" value="None"/>	<input type="button" value="Edit..."/>
3:	<input type="text" value="Details1"/>	<input type="text" value="Details 1"/>	<input type="text" value="Per Name"/>	<input type="text" value="Page"/> ▼	<input type="text" value="None"/>	<input type="button" value="Edit..."/>

**Binding**

Folder Binding:  ▼

Required Folder Class:

Binding Prefix:

**Details**

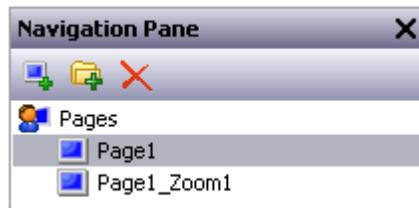
Details Creation:  ▼

Details Widget:

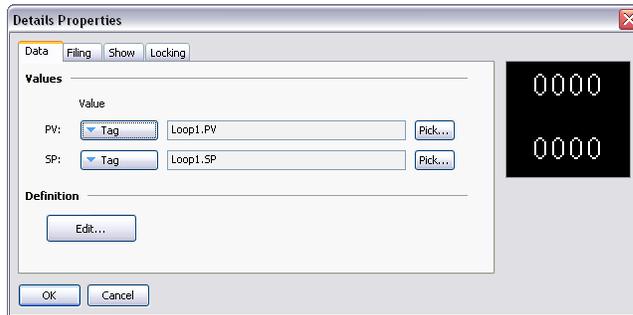
### RESULTATS DE LA LIAISON

Lorsque nous connectons le widget à notre boucle PID, une nouvelle page est créée pour contenir ce dernier. Le nom de la nouvelle page est basé sur le nom de notre page contenant le widget aperçu, mais avec un suffixe « Zoom » et un nombre choisi rendant le nom unique.

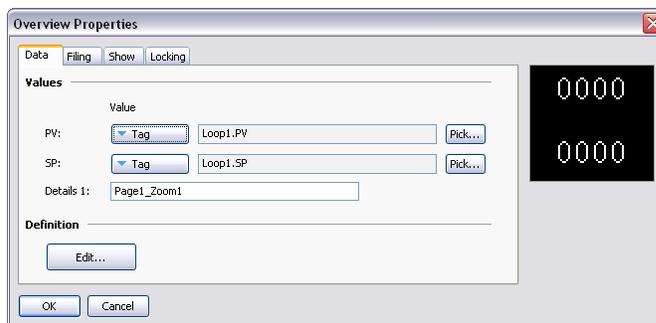
Cette page est placée dans la liste de navigation située sous la page en cours...



Les détails créés sur cette page sont liés à notre boucle...



Et les propriétés de notre widget aperçu sont modifiées comme suit...



On peut facilement définir un bouton dans notre widget vue d'ensemble, ce dernier invoquant une action de `ShowPopup(Details1)` et affichant ainsi le widget de détails associé. Le widget de détails peut lui-même fermer le popup en appelant la fonction `HidePopup()`.

## PAGES DETAILS MULTIPLES

Si plusieurs pages de détails sont créées, vous vous souviendrez que les points de données appelée `Details1`, `Detail2` et ainsi de suite dans le widget aperçu dépendront des noms de ces pages. Ces données peuvent également être définies sur les widgets de détails, et le seront également par les noms des pages qui ont été créés. Ceci est utile si vous voulez permettre à la première page de détails d'accéder à la deuxième et ainsi de suite, associant ainsi les pages ensemble. Le widget de détail peut également définir un élément de données particulier appelé `DetailsP` qui sera égal à la page qui contient le widget de vue d'ensemble. Ceci peut être utilisé pour revenir à la vue d'ensemble, quelque chose qui ne peut être atteint grâce à un simple `GotoPrevious()` lorsque plusieurs pages de détails sont fournies.



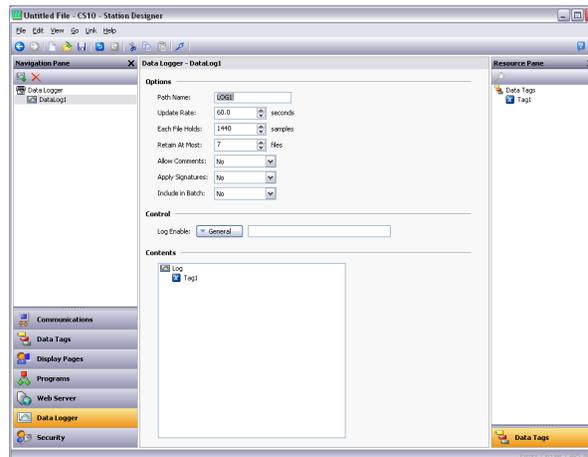
# UTILISATION DES JOURNAUX DE DONNÉES

Maintenant que vous avez configuré le noyau de votre application, vous pouvez décider de faire usage du Journal de données de Crimson pour enregistrer les valeurs de balises sur CompactFlash. Les données enregistrées de cette manière sont stockées dans des fichiers CSV, et peuvent être facilement importées dans des applications comme Excel via de nombreuses méthodes. Pour configurer la journalisation des données, sélectionnez le Journal de données dans la catégorie du Panneau de navigation.

## CREATION DE JOURNAUX DE DONNEES

Les journaux de données sont créés dans la liste de navigation de la manière habituelle.

Chaque journal possède les propriétés suivantes...



- La propriété *Nom Chemin* vous permet de modifier le répertoire dans lequel le journal sera enregistré. Par défaut, le journal est enregistré dans un répertoire nommé d'après le nom de ce dernier. Cette propriété vous permet de renommer les journaux d'une manière qui n'est pas compatible avec la convention 8.3.
- La propriété *Taux de mise à jour* est utilisée pour indiquer combien de fois Crimson prélèvera un échantillon des éléments de données à enregistrer. Même si une décimale peut être saisie, l'échantillonnage n'est que de 200ms. Le plus rapide des taux d'échantillonnage est d'une seconde, mais notez que l'utilisation d'un taux si élevé produira de très grandes quantités de données. Tous les tags du journal seront échantillonnés à la même vitesse.
- La propriété *Chaque fichier contient* est utilisée pour indiquer combien d'échantillons seront inclus dans chaque fichier journal. Lorsque de nombreux échantillons ont été enregistrés, un nouveau fichier journal est créé en utilisant un nom différent. Généralement, cette valeur est définie de telle sorte que chaque fichier journal contient une quantité raisonnable de données. Par exemple, le journal ci-dessus est configuré pour utiliser un nouveau fichier journal chaque jour.

- La propriété *Conserver à la plupart* est utilisée pour indiquer la manière dont de nombreux journaux seront conservés sur CompactFlash avant que le fichier le plus ancien ne soit supprimé.
- La propriété *Autoriser les commentaires* est utilisée pour activer ou désactiver l'ajout de commentaires dans le journal de données via la fonction `LogComment()`. Reportez-vous au manuel de référence pour savoir comment cette fonction peut être utilisée.
- La propriété *Inclure dans le lot* est utilisée pour inclure ou exclure ce journal à partir du lot de journalisation système. Voir ci-dessous pour obtenir des renseignements sur la façon dont fonctionne l'enregistrement par lots.
- La propriété *Activation Journal* est utilisé pour permettre ou empêcher l'enregistrement du journal. Si l'expression saisie est True, le journal activé. Si l'expression est fausse, le journal sera désactivé. Si aucune expression n'est inscrit, le journal sera activé par défaut.
- La propriété *Contenu* est utilisée pour indiquer les balises qui doivent être incluses dans les données du journal. Les tags peuvent être glissés dans la liste du panneau de ressources et de haut en bas via un cliquer-déposer standard.

## JOURNALISATION GROUPEE

Lorsque vous accédez pour la première fois au journal de données, vous remarquerez un paramètre global permettant d'activer ou de désactiver l'enregistrement de commandes. Pour les données normales de « logging », le Journal de données permettra de sauver les fichiers de log sous le nom de dossier spécifié pour chaque journal. La journalisation groupée, d'autre part, enregistre également tous les journaux qui sont configurés dans à un répertoire nommé d'après le lot de production actuel. Cela permet à tous les journaux liés à un lot particulier d'être accédés et manipulés en tant que groupe.

Pour illustrer cela, pensez à la structure des répertoires suivants...



Cet exemple est tiré d'un dispositif cible qui a activé la journalisation de groupe et possède deux journaux de données. Les premières données du journal sont incluses dans le lot, alors que les secondes ne le sont pas. Notez que les fichiers journaux sont stockés par défaut dans les répertoires nommés `LOGS/LOG1` et `LOGS/LOG2`. À noter également, toutefois, que le premier journal est également mis en sous-répertoire sous le répertoire `BATCH`. Chaque sous-répertoire contient les données échantillonnées entre le moment où ce lot a été lancé et le moment où le lot a été terminé.

## CONTROLE DES LOTS

La journalisation groupée est contrôlée par un certain nombre de fonctions. `NewBatch (nom)` va créer un dossier nommé *nom*, terminant le lot en cours et lançant les suivants. Les fichiers enregistrés après cette commande seront sauvegardés dans le nouveau dossier. La commande `EndBatch ()` va arrêter le lot en cours, tandis que `GetBatch ()` retournera le nom du lot qui est actuellement actif. Pour de plus amples renseignements, veuillez consulter le manuel de référence.

## STOCKAGE DES FICHIERS JOURNAUX

Comme décrit ci-dessus, les journaux de données stockent leurs données dans une série de fichiers sur la carte CompactFlash de l'appareil cible. Les fichiers sont placés dans le sous-répertoire spécifié dans les propriétés du journal, sous une entrée de répertoire racine, et sont appelés LOGS.

Les fichiers LOG sont nommés d'après l'heure et la date auxquelles le journal est programmé pour commencer. Si chaque fichier contient une heure ou plus d'informations, les fichiers seront nommés `YYMMJJhh.CSV`, où `AA` représente l'année du dossier, `MM` représente la mois, `JJ` correspond au jour, et `hh` représente l'heure.

La longueur de chaque fichier dépend de la *Fréquence de mise à jour* et de la propriété *Chaque fichier contient*. Par exemple, si la fréquence de mise à jour est de 5 secondes et le nombre d'éléments pouvant être contenus de 360, chaque fichier contiendra  $(5 \times 360) / 60 = 30$  minutes de données, utilisant ainsi le format `MMJJhhmm.CSV`. Un nouveau fichier sera créé toutes les 30 minutes.

## PROCESSUS DE JOURNALISATION

Le Journal de données Crimson fonctionne en utilisant deux procédés distincts. Les premiers échantillonne chaque point de données au taux prévu par les propriétés de chaque journal, et place ces dernières en mémoire tampon dans la RAM de l'appareil cible. Le second processus s'exécute toutes les deux minutes, et écrit les données de la mémoire sur la carte CompactFlash .

Cette structure possède plusieurs avantages...

- Les données enregistrées sur la carte CompactFlash possèdent la garantie de commencer avec une limite de 2 minutes. Cela signifie que si votre appareil cible prend en charge le remplacement à chaud des cartes CF, vous pouvez attendre la prochaine rafale d'écritures pour commencer, et, lorsque la LED de la CompactFlash cesse de clignoter, vous êtes assurés d'avoir au moins jusqu'au début du prochain intervalle de deux minutes avant que de nouvelles écritures soient tentées. Cela signifie que vous pouvez retirer la carte sans crainte de corruption de données. Tant que vous insérez une nouvelle carte avant que quatre minutes se sont écoulées, aucune donnée ne sera perdue.
- Les écritures sur la carte CompactFlash peuvent atteindre un niveau beaucoup plus élevé de performance, si on décide de ne pas actualiser en permanence la carte des structures de données du système de fichiers pour chaque échantillon unique. Pour les journaux configurés pour l'échantillon à des débits très élevés, la bande passante d'une carte CompactFlash type ne permettrait pas d' écrire de

données de façon fiable en l'absence d'un tel processus de mise en mémoire tampon.

Notez que parce que les données ne sont pas engagées dans la carte CompactFlash pour un maximum de deux minutes, cette quantité de données du journal peut être perdue lorsque le terminal est éteint. En outre, si le périphérique cible est éteint tandis que l'écriture est en cours, la carte CompactFlash peut être endommagée. Pour s'assurer que cette corruption n'est pas permanente, Crimson utilise un système de journalisation qui met en cache les écrits supplémentaires de mémoire non volatile dans le terminal. Si l'appareil détecte une interruption, l'écriture sera répétée une fois l'alimentation est rétablie.

Si vous souhaitez retirer une carte CompactFlash à partir d'un panneau d'interprétation d'enregistrement de données, vous devez suivre la procédure décrite ci-dessus concernant la LED d'activité, et ne couper l'alimentation que lorsque l'activité a cessé. Si vous n'êtes pas sûr que le terminal a été éteint correctement, rallumez ce dernier, laissez le temps d'une séquence complète s'écouler et éteignez le terminal en suivant la procédure normale. La carte peut alors être enlevée en toute sécurité.

Étant données les fluctuations nécessaires pour retirer une carte CompactFlash, Crimson fournit toute une gamme d'autres mécanismes permettant d'accéder aux fichiers journaux, éliminant ainsi la nécessité de telles expulsions. Ces méthodes sont décrites ci-dessous.

## ACCEDER AUX JOURNAUX

Il existe cinq méthodes d'accès aux fichiers de log...

- Vous pouvez monter la carte CompactFlash comme un lecteur sur un PC via la procédure décrite au début de ce manuel. Cela permettra de copier les journaux via Windows Explorer. Cette méthode a certains inconvénients en termes de quantité de charge limitée lors de la première utilisation de la carte.
- Vous pouvez utiliser le serveur web qui est décrit dans le chapitre suivant. Une fois ce dernier activé, les fichiers d'historique peuvent être consultés via une connexion TCP / IP, en utilisant un navigateur Web tel que Microsoft Internet Explorer, ou en utilisant le processus automatisé mis en œuvre par l'utilitaire WebSync fourni avec Crimson.
- Vous pouvez utiliser le serveur FTP pour permettre aux clients distants de se connecter à l'appareil Crimson et de télécharger les journaux. Se référer au chapitre Services pour les détails.
- Vous pouvez utiliser le Gestionnaire de synchronisation pour envoyer les fichiers vers un serveur FTP sur une base périodique. Encore une fois, consultez le chapitre Utilisation de services pour plus de détails.
- Vous pouvez activer la copie automatique des fichiers journaux sur une clé USB en configurant l'option Carte mémoire dans la catégorie communication. Se référer au chapitre Communications de ce manuel pour plus de détails.

# UTILISATION DU SERVEUR WEB

Le serveur web Crimson peut être utilisé pour exposer les diverses données via des connexions TCP / IP, en utilisant des modems ou les ports Ethernet du périphérique cible. Cela permet un accès distant à des informations de diagnostic ou aux valeurs enregistrées par les données de journal. Le serveur web est configuré en sélectionnant la catégorie Serveur web dans le Panneau de navigation.

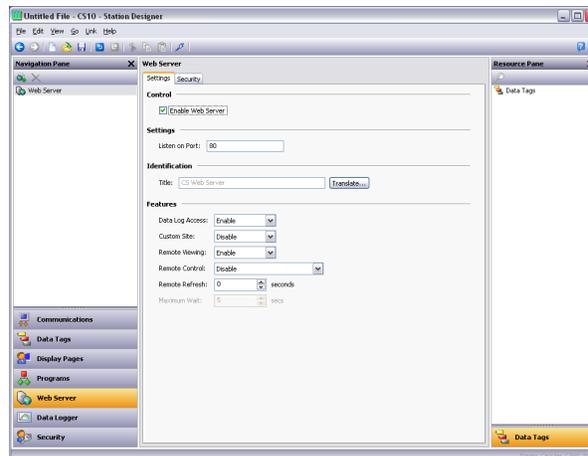
## NOTE IMPORTANTE

Si que Crimson fournit toute une gamme de mécanismes de protection pour limiter l'accès au serveur web, vous devez toutefois maîtriser les bonnes pratiques d'ingénierie pour concevoir votre système. Cela signifie que vous devez éviter d'effectuer toute les opérations liées à la sécurité via le serveur web, et que vous devriez idéalement utiliser une pare-feu externe pour empêcher l'accès non autorisé. La sécurité relève de votre propre responsabilité, et Red Lion Controls ne recommande pas que vous comptiez uniquement sur les mesures de sécurité proposées par Crimson.

## PROPRIETES DU SERVEUR WEB

Les propriétés du serveur web sont accessibles à partir de l'entrée racine de la liste de navigation.

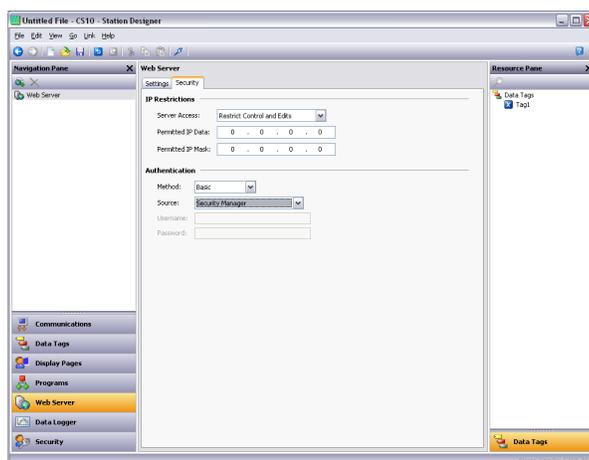
### PARAMETRAGE DES PROPRIETES



- La propriété *Activer le Serveur Web* est utilisé pour activer ou désactiver le serveur web. Si le serveur est activé, le panneau va attendre les demandes entrantes, puis y répondra selon les besoins. Si le serveur est désactivé, les connexions sur ce port seront refusées. N'oubliez pas que pour que le serveur fonctionne, une connexion TCP / IP doit avoir été configurée via la catégorie Communications.
- La propriété *Ecouter sur le Port* indique le numéro de port TCP sur lequel le serveur web va écouter. Le port 80 est le port standard utilisé par le protocole HTTP et conviendra très probablement à votre application.

- La propriété *Titre* est utilisée pour fournir le titre devant figurer sur le menu du serveur web. Ce titre peut être utilisé pour différencier plusieurs terminaux sur un réseau, assurant ainsi une administration correcte.
- La propriété *Accès journal de données* est utilisée pour activer ou désactiver l'accès au Web des fichiers créés par les journaux de données. Cette fonction ne doit être activée que si le serveur web est utilisé par un programme client à distance synchronisant automatiquement les journaux de données.
- La propriété *Visualisation à distance* est utilisée pour activer ou désactiver une installation via laquelle un navigateur Web peut être utilisé pour afficher le contenu actuel de l'affichage du périphérique cible. Cette installation est très utile pour le diagnostic à distance des problèmes que l'exploitant peut rencontrer avec le panneau de commandes ou avec la machine qu'il contrôle.
- La propriété *Contrôle à distance* est utilisée pour activer ou désactiver une option par laquelle l'installation de visualisation à distance permet à un navigateur Web destiné d'être utilisé pour simuler la pression de touches ou de primitives d'affichage. Bien que cette fonctionnalité soit extrêmement utile, il faut prendre soin de maîtriser les différents paramètres de sécurité afin d'éviter toute manipulation non autorisée d'une machine. L'utilisation d'un pare-feu externe est aussi fortement recommandée si le panneau est accessible depuis l'Internet.
- La propriété *Site personnalisé* est utilisée pour activer ou désactiver une installation par laquelle les fichiers stockés dans le répertoire \WEB de la carte CompactFlash sont exposés via le serveur web. Cette installation est décrite plus en détail ci-dessous.
- La propriété *Rafraîchissement à distance* présente la fréquence à laquelle le navigateur Web connecté au serveur web actualise l'affichage. Une valeur de zéro entraînera un rafraîchissement aussi rapide que possible. Des valeurs plus élevées permettront de réduire la bande passante, et peuvent être adaptés aux connexions modem.

## PROPRIÉTÉS DE SÉCURITÉ



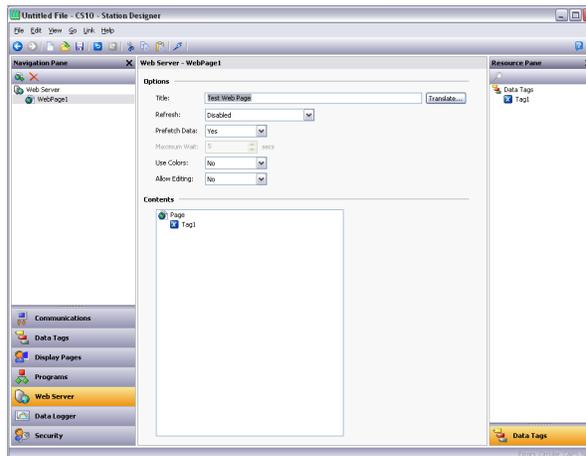
- Le groupe *Restrictions IP* est utilisé pour restreindre l'accès du Serveur Web à des hôtes dont l'adresse IP correspond à un masque et à des données indiquées.

Tous les accès peuvent être restreints, ou le filtre peut être utilisé pour restreindre simplement la commande à distance ou d'autres fonctionnalités.

- Le groupe *Authentication* permet de définir si les utilisateurs se connectant au serveur web sera tenu de fournir des informations de type nom d'utilisateur / mot de passe, et comment ces dernières seront validées. La *Méthode* définit l'algorithme à utiliser, Digest étant le choix recommandé. La propriété *Source* est utilisée pour indiquer si vous entrez le nom d'utilisateur et mot de passe requis directement dans les propriétés du serveur web, ou si vous allez créer des utilisateurs dans le système de Sécurité Crimson puis leur accorder l'accès au serveur web.

## AJOUT DE PAGES WEB

En plus des installations décrites ci-dessus, le serveur web supporte l'affichage générique des pages web, dont chacune contient une liste prédéfinie de valeurs de balises. Ces pages sont créées dans le Panneau de navigation de la manière habituelle. Chaque page Web possède les propriétés suivantes...



- La propriété *Titre* est utilisée pour identifier la page Web dans le menu présenté à l'utilisateur via son navigateur Web. Bien que le titre est traduisible, les versions actuelles de Crimson utilisent la version américaine du texte.
- La propriété *Rafraîchir* est utilisée pour indiquer si oui ou non le navigateur Web doit être chargé d'actualiser le contenu de la page automatiquement. Le taux de mise à jour est compris entre 1 et 8 secondes. Notez que le montant de scintillement du navigateur Web variera selon les performances de la machine employée. La mise à jour n'est pas destinée à être exempte de scintillements.
- La propriété *Utiliser couleurs* est utilisée pour indiquer si des couleurs définies par une balise doivent être utilisées lors du rendu sur cette page. Si activée, la couleur affichée dans le navigateur web changera en fonction du statut du tag. Reportez-vous au chapitre Utilisation des tags de données pour plus de détails.
- La propriété *Autoriser les Modification* est utilisée pour permettre l'édition des données de Tags via cette page. Si activée, chaque valeur de données possédera un bouton Modifier permettant à l'utilisateur de changer cette valeur. Si l'e tag possède des paramètres de sécurité définis, l'utilisateur connecté sur le serveur

web doit posséder les droits suffisants pour modifier ce dernier L'utilisation de l'authentification est recommandée pour cette fonctionnalité.

- La propriété *Contenu* est utilisée pour indiquer les balises qui doivent être incluses sur la page. les tags peuvent être glissés dans la liste à partir du Panneau de ressources, et de haut en bas dans la liste en utilisant les techniques standards de glisser-déposer.

## UTILISATION D'UN SITE WEB PERSONNALISE

Alors que les pages Web standard fournissent un accès rapide et simple aux données dans le maître, vous risquez de trouver que votre incapacité à modifier leur mise en forme precise contrarie assez vos capacités artistiques. Ainsi, vous pouvez utiliser la fonction de site personnalisé du maître pour créer un site Web entièrement personnalisé à l'aide de votre éditeur HTML tiers favori et, en insérant certaines séquences spéciales et en enregistrant les fichiers obtenus sur la carte CompactFlash du maître, exposer ce site à l'aide du serveur Web du périphérique cible.

### CREATION DU SITE

Le site Web peut utiliser n'importe quelle fonction HTML prise en charge par votre navigateur, mais il ne doit pas utiliser ASP, CGI ou d'autres astuces côté serveur. Les noms de fichiers utilisés pour les fichiers HTML et les graphiques associés doivent également être conformes à l'ancienne convention d'attribution de nom 8.3. Ainsi, les extensions de fichier sont, par exemple, `HTM` au lieu de `HTML` et `JPG` au lieu de `JPEG`. Cela signifie également que le corps du nom de fichier doit comporter huit caractères maximum et que vous ne devez pas vous fier à la différence entre les majuscules et les minuscules pour différencier les pages. Vous pouvez utiliser n'importe quelle structure de répertoire si vous vous assurez que vos répertoires observent la convention d'attribution de nom 8.3 et si vous ne vous fiez pas aux différences de casse.

### DONNEES EMBARQUEES

Pour incorporer des données d'étiquettes dans une page Web, insérez la séquence `[[N]]` en remplaçant `N` par le numéro d'index de l'étiquette en question. Ce numéro d'index s'affiche dans la barre d'état lorsqu'une étiquette est sélectionnée dans la fenêtre Etiquettes de données et correspond plus ou moins à l'ordre de création des étiquettes. Lorsque la page Web qui contient cette séquence est utilisée, la séquence est remplacée par la valeur actuelle de l'étiquette, mise en forme selon les propriétés de l'étiquette.

### DEPLOIEMENT DU SITE

Pour déployer votre site Web personnalisé, copiez-le dans le répertoire `\WEB` de la carte CompactFlash qui doit être installée sur le maître. Pour copier les fichiers, montez la carte comme lecteur sur votre PC, tel que décrit au début du présent manuel, ou utilisez un programme d'écriture de carte approprié connecté à votre PC. Assurez-vous que la propriété Activer site Web personnalisé est définie et le site personnalisé s'affichera alors dans le menu du serveur Web. Une fois que le site est sélectionné, un fichier appelé `DEFAULT.HTM` du répertoire `\WEB` s'affiche. Au-delà de ce point, la navigation s'effectue en fonction des liens présents dans le site.

# UTILISATION DU SYSTEME DE SECURITE

Crimson contient des fonctionnalités qui vous permettent de définir les opérateurs qui ont accès à telle ou telle page d'affichage et de limiter ces opérateurs qui peuvent apporter des modifications aux données sensibles. Le logiciel contient également une fonctionnalité de journalisation de la sécurité que vous pouvez utiliser pour enregistrer les modifications apportées aux valeurs de données, indiquant la date de la modification et l'utilisateur à l'origine de cette modification.

## PRINCIPES DE BASE

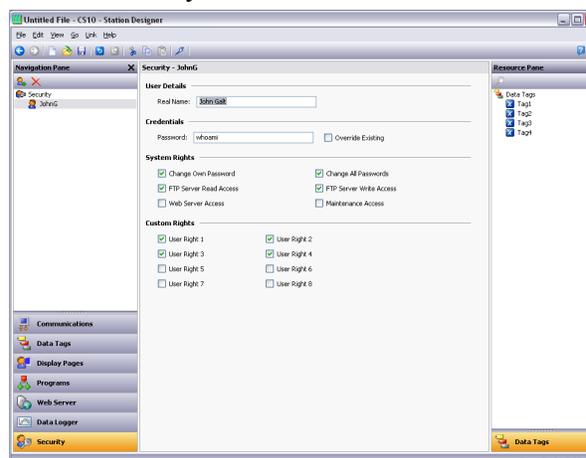
Les sections suivantes détaillent certains des concepts de base que le système de sécurité utilise.

### SECURITE BASEE OBJET

Le système de sécurité de Crimson est basé sur l'objet. Ainsi, les caractéristiques de sécurité sont appliquées à une page d'affichage ou une étiquette, et pas à l'élément de l'interface utilisateur qui accède à la page ou apporte une modification à l'étiquette. L'autre approche basée sur le sujet implique généralement que vous devez être attentif à appliquer les paramètres de sécurité à chaque élément de l'interface utilisateur qui peut modifier les données sécurisée. L'approche de Crimson évite cette duplication et garantit qu'une fois que vous avez décidé de protéger une étiquette, elle reste protégée dans toute votre base de données.

### UTILISATEURS NOMMES

Crimson prend en charge la fonctionnalité de création d'utilisateurs (quel que soit leur nombre). Chacun d'entre eux aura alors un nom d'utilisateur, un nom complet et un mot de passe. Le nom d'utilisateur est une chaîne respectant la casse sans aucun espace incorporé, qui permet d'identifier l'utilisateur lors de la connexion alors que le nom réel est en general une chaîne plus longue utilisée dans les fichiers de connexion pour enregistrer l'identité lisible de l'utilisateur qui apporte la modification. Notez que vous êtes libre d'utiliser ces champs d'autres façons s'ils sont adaptés à votre application. Vous pouvez par exemple créer des utilisateurs qui représentent des groupes d'individus ou peut-être des rôles comme Opérateurs, Superviseurs et Managers. Vous pouvez également décider d'utiliser le nom complet pour conserver un élément comme un numéro d'horloge afin de lier les identités des utilisateurs à votre système MRP.



## DROITS UTILISATEURS

Chaque utilisateur dispose d'aucun ou de plusieurs droits d'accès. Un utilisateur sans aucun droit peut accéder aux objets qui nécessitent simplement l'enregistrement de l'identité de l'utilisateur alors que les utilisateurs avec plusieurs droits peuvent accéder aux objets qui exigent que ces droits soient présents. Les droits sont divisés en Droits système et en Droits utilisateur, les premiers contrôlant l'accès aux fonctions du logiciel Crimson et les seconds étant disponibles pour une utilisation générale. Par exemple, le Droit utilisateur 1 peut être utilisé dans votre base de données pour contrôler l'accès aux cibles de production. Seuls les utilisateurs qui, selon vous, devraient pouvoir modifier ces éléments disposent de ce droit.

## CONTROLE D'ACCES

Les objets qui sont soumis à la sécurité possèdent un paramètre de *Contrôle d'accès* associé.

L'édition des propriétés se présente sous cette forme...

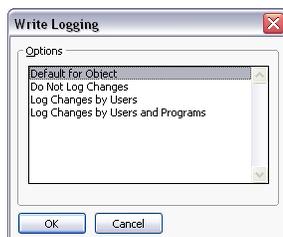


Ces paramètres vous permettent de spécifier si n'importe quel utilisateur peut accéder à l'élément, si n'importe quel opérateur authentifié peut y accéder ou si les utilisateurs avec des droits d'utilisateur spécifiques peuvent y accéder. Le paramètre du contrôle d'accès vous permet également de spécifier si une étiquette peut être modifiée par un programme qui s'exécute suite à quelque chose d'autre que l'action d'utilisateur. Cette fonctionnalité vous permet de garantir qu'aucune modification en arrière-plan des données sensible ne s'est produite même si une erreur de programmation tente d'apporter une telle modification.

## JOURNALISATION DES ECRITURES

Les tags possèdent également une propriété *Journalisation des écritures*.

L'édition de cette dernière se fait dans la fenêtre ci-dessous...



Elle indique si les modifications apportées à une étiquette par des utilisateurs ou des programmes doivent être enregistrées. Cette fonctionnalité vous permet de créer un suivi

d'audit des modifications apportées à votre système, simplifiant de ce fait la détection d'erreurs et fournissant des informations de contrôle qualité permettant de traiter la configuration. Notez que vous devez être attentif lorsque vous enregistrez les modifications apportées par les programmes car certaines bases de données peuvent enregistrer des quantités incontrôlables de données dans de telles circonstances.

### **ACCES PAR DEFAULT**

Pour accélérer le processus de configuration, Crimson permet également de spécifier un accès par défaut et des paramètres de journalisation d'écritures pour les étiquettes mappées et internes ainsi que pour les pages d'affichage. La différenciation entre les étiquettes mappées et non mappées est importante dans les systèmes où toutes les modifications apportées aux données externes doivent être enregistrées, mais où les données internes de Crimson peuvent être manipulées sans avoir à recourir à un tel suivi d'audit.

### **CONNEXION A LA DEMANDE**

Le système de sécurité de Crimson prend en charge la connexion classique et la connexion à la demande. Une connexion classique peut se produire lorsque vous utilisez un élément de l'interface utilisateur comme un bouton-poussoir pour activer l'action Authentification Utilisateur ou pour appeler la fonction `UserLogOn()`. Une connexion à la demande se produit si l'opérateur tente une action sans disposer des droits d'accès suffisants et si cette tentative de connexion à abouti. Par exemple, un utilisateur peut appuyer sur un bouton qui exécute un programme pour réinitialiser plusieurs valeurs. Dès que le programme tente de modifier une valeur qui nécessite un accès sécurisé, le système invite à entrer les informations d'identification d'ouverture de session. Cette méthode réduit l'interaction de l'opérateur et le système est plus réactif...

### **ACCES A LA MAINTENANCE**

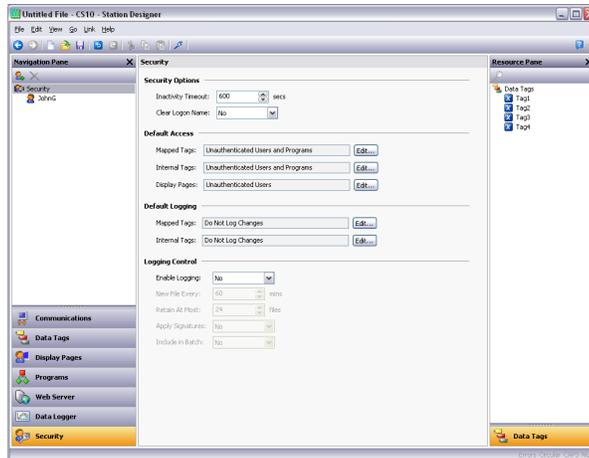
Le système fournit également une fonctionnalité appelée Mode Maintenance qui permet de supprimer le délai d'inactivité de l'utilisateur lors de la mise en service du système. Ce mode est activé si une page d'affichage est marquée comme étant accessible avec le droit Accès Maintenance et si l'utilisateur actuel a obtenu l'accès à la page en conséquence de ce droit. L'utilisation de ce mode vous évite de devoir vous connecter de façon répétée lorsque vous testez le système.

### **VERIFIER AVANT FONCTIONNEMENT**

La fonctionnalité Vérifier avant fonctionnement permet de forcer l'utilisateur à confirmer chaque modification d'un élément particulier de données sensibles. Cette fonctionnalité est activée en sélectionnant les paramètres appropriés dans le descripteur de sécurité d'un tag de données. Quand une modification est apportée à une balise pour laquelle cette fonction est activée, une fenêtre apparaît et affiche les valeurs anciennes et nouvelles, demandant confirmation avant d'autoriser le changement.

## PARAMETRES DE SECURITE

Vous pouvez accéder aux Paramètres de sécurité via l'élément racine de la catégorie Sécurité...

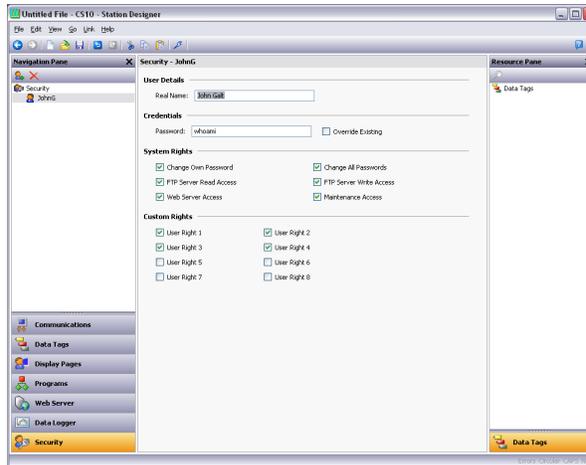


Les propriétés disponibles se présentent comme suit...

- La propriété *Délai d'inactivité* permet d'indiquer la durée écoulée dans aucune entrée d'utilisateur avant que l'utilisateur actuel soit automatiquement déconnecté. Si la valeur du paramètre est trop élevée, le système n'est pas sécurisé. Si la valeur du paramètre est trop basse, les opérateurs peuvent rencontrer des difficultés manipulations avec le système.
- La propriété *Effacer le Nom d'Utilisateur* permet d'indiquer si le nom d'utilisateur doit être effacé ou non avant de demander à l'opérateur de se connecter. Si ce paramètre est désactivé, le nom d'utilisateur précédent est affiché et seul le mot de passe doit être ressaisi. Si cette fonctionnalité est activée, une plus grande sécurité est garantie. Elle peut également être requise pour une conformité avec les normes de sécurité de certains secteurs.
- Les propriétés *Accès par défaut* permettent d'indiquer l'accès à fournir aux différents objets au cas où aucun accès spécifique ne serait défini pour cet élément. Les paramètres sont identiques à ceux qui sont décrits dans la section Contrôle de l'accès ci-dessus.
- Les propriétés *Enregistrement par Défaut* permettent d'indiquer si les modifications apportées aux étiquettes mappées et non mappées doivent être enregistrées si aucun critère de journalisation spécifique n'est défini pour une étiquette. Il est impossible d'enregistrer un accès par programmation par défaut car vous devez envisager une telle journalisation avec soin pour éviter une activité d'historique excessive.
- Les propriétés *Contrôle de l'Enregistrement* permettent de définir si et comment les journaux de sécurité doivent être créés. Reportez-vous au chapitre Configuration de l'enregistrement des données pour obtenir des informations sur l'écriture des données et l'attribution de noms aux fichiers.

## CREATION DE COMPTES UTILISATEURS

List Les comptes utilisateurs sont créés et manipulés selon les methods usuelles de la Liste de navigation...



Chaque utilisateur possède les propriétés suivantes...

- La propriété *Nom Réel* permet d'enregistrer l'identité d'un utilisateur dans les journaux de sécurité et dans la primitive Gestionnaire de sécurité qui est utilisée pour modifier les mots de passe à partir du terminal de l'opérateur. Si la sécurité maximale est requise, le nom de l'utilisateur ne doit pas être facilement dérivé du nom complet.
- La propriété *Mot de passe* permet de spécifier un mot de passe initial pour cet utilisateur. Le mot de passe respecte la casse et se compose de caractères alphanumériques. Notez que si la case *Réécriture existant* est cochée, toutes les modifications apportées à ce mot de passe à partir du panneau de l'opérateur sont remplacées lorsque cette base de données est téléchargée dans le panneau.
- Les propriétés *Droits système* permettent d'attribuer à un utilisateur la capacité à effectuer certaines actions du système. Les propriétés relatives aux modifications du mot de passe sont évidentes et l'utilisateur du Mode de maintenance est décrit ci-dessus.
- Les propriétés *Droits personnalisés* permettent d'attribuer à un utilisateur certains droits qu'il peut alors utiliser dans la base de données pour autoriser l'accès aux groupes d'étiquettes ou de pages d'affichage. L'utilisation exacte de ces droits dépend du concepteur du système.

## SECURITE DES TAGS

Chaque étiquette accessible en écriture dispose d'un onglet appelé Sécurité qui permet de définir les paramètres du contrôle d'accès et de la journalisation d'écritures de cette étiquette. Si vous ne définissez pas les paramètres spécifiques, le système utilise les paramètres par défaut appropriés, selon qu'il est mappé ou non aux données externes.

## SECURITE DES PAGES

Les paramètres du contrôle d'accès d'une page d'affichage sont définis via la boîte de dialogue Propriétés...



Une nouvelle fois, si aucun paramètre n'est défini, les paramètres par défaut sont utilisés.

## FONCTIONS RELATIVES A LA SECURITE

Reportez-vous à l'Annexe A de ce manuel pour obtenir des détails sur les fonctions `UserLogOn()`, `UserLogOff()` et `TestAccess()`. Cette troisième fonction est utile lorsque vous modifiez plusieurs valeurs dans un programme car elle vous permet de forcer une vérification anticipée de l'accès dans le code pour éviter d'apporter des modifications uniquement pour que les opérations ultérieures échouent à cause de droits d'utilisateur insuffisants.

# UTILISATION DES SERVICES

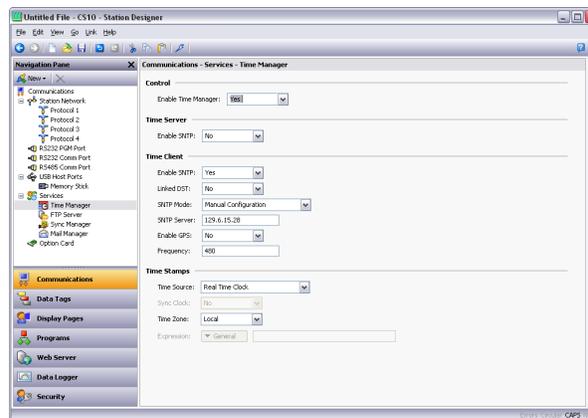
In addition to the core functions described earlier in this document, the Communications category also allows various services to be configured. These services appear in le Panneau de navigation under the Services icon, and each is described below.

## GESTION DU TEMPS

Crimson propose des fonctions qui vous permettent de synchroniser l'heure et la date dans le périphérique cible avec différentes sources. Le Gestionnaire de l'heure peut également conserver des informations sur le fuseau horaire actuel du maître et si l'heure d'été est actuellement activée. En effet, les informations précises sur le fuseau horaire représentent une synchronisation vitale et correcte car les différentes méthodes de synchronisation sont toutes conçues pour fonctionner avec l'heure UCT (également appelée heure de Greenwich).

## CONFIGURATION DU SERVICE

Ce service est configure via l'icône qui lui est associée dans le panneau de navigation...



La propriété *Activer le Gestionnaire de l'heure* permet de contrôler l'accès aux autres fonctions. Si elle n'est pas activée, Crimson fonctionne à l'heure locale et ne reconnaît pas les fuseaux horaires ou les autres informations de gestion de l'heure.

## SERVEUR TEMPS

Configurer correctement la propriété Activer SNTP du serveur temporel fera agir Crimson comme un serveur SNTP. Cela permettra à d'autres dispositifs de Crimson desynchroniser leurs propres horloges sur cet appareil. Notez que l'implémentation du protocole SNTP Crimson n'est pas entièrement conforme à la RFC, et n'est pas prise en charge comme une source de synchronisation pour des clients tiers.

## CLIENT TEMPS

En sélectionnant Oui dans la propriété Activer SNTP de la section client de temps, vous autorisez Crimson à lancer son client SNTP. Crimson tentera ensuite de synchroniser son horloge avec un autre périphérique, ou sur un autre réseau accessible via SNTP, comme un ordinateur fonctionnant sous Windows XP. Le client possède les fonctionnalités suivantes...

- La propriété *DST liée* permet de demander au client SNTP de tenter de lire le paramètre actuel Heure d'été à partir du serveur SNTP. Comme cette fonction ne fait pas partie standard du protocole SNTP, elle ne fonctionne que si un autre maître ou un écran opérateur G3 est défini comme serveur. Cette fonction est utile dans le sens où elle permet de régler l'heure d'été via un seul périphérique sur le réseau de l'usine et les autres périphériques suivent le paramètre central.
- Les propriétés *Mode du SNTP* et *Serveur SNTP* permettent de configurer l'adresse IP du serveur SNTP (Simple Network Time Service). Si Configuré via DHCP est sélectionné, le port Ethernet de l'unité doit être configuré pour utiliser le protocole DHCP et le serveur DHCP du réseau doit être configuré pour créer un serveur via l'option 42.
- La propriété *Sync GPS* permet de demander au client de temps d'utiliser une unité GPS connectée via NMEA-0183 comme autre méthode d'obtention de l'heure actuelle. L'unité peut être connectée sur n'importe quel port série à l'aide du pilote approprié.
- La propriété *Fréquence* permet de spécifier la fréquence à laquelle Crimson doit tenter de synchroniser son heure par les méthodes activées ci-dessus. Crimson tente de synchroniser vingt secondes après la mise sous tension, puis tel que spécifié par cette propriété. Si une tentative donnée de synchroniser échoue, l'unité retente toutes les 30 secondes jusqu'à ce que l'opération réussisse. Si les synchronisations du GPS et du SNTP sont activées, le SNTP est uniquement utilisé si un GPS n'est pas disponible.

#### MARQUES TEMPORELLES

Crimson peut enregistrer plusieurs fichiers journaux sur la carte CompactFlash de l'appareil cible, chaque entrée de journal possédant un marquage temporel. Par défaut, le marquage de temps vient de l'horloge locale en temps réel et se trouve dans le fuseau horaire local. Le comportement peut être modifié via les propriétés suivantes...

- La propriété *Source de temps* est utilisée pour indiquer d'où les marquages temporels devraient être obtenus. Le réglage par défaut obtient l'heure de l'unité propre horloge temps réel, tandis qu'une variante permet l'utilisation d'une expression pour définir l'heure actuelle. Cette expression est généralement une référence à un élément de données dans un périphérique connecté, ce qui permet à l'horloge d'utiliser le périphérique pour l'enregistrement de données. L'expression doit être inscrite dans la propriété *Expression*.
- La propriété *Synchronisation d'horloge* est utilisée pour indiquer si l'horloge locale en temps réel devrait être synchronisée à la source de temps de remplacement mentionné ci-dessus. Si cette option est activée, l'horloge locale sera synchronisée sur le démarrage et périodiquement par la suite, et sera utilisée comme source d'horodatage si la source de remplacement n'est pas disponible en raison de problèmes de communication.
- La propriété *Fuseau horaire* est utilisée pour indiquer le fuseau horaire à utiliser pour les marques de temps. Elle n'est applicable que lorsque l'horloge locale en temps réel est configurée en tant que source pour l'horodatage. La sélection locale utilisera le fuseau horaire local, la sélection utilisera à la place le temps universel

coordonné UTC. Le paramétrage de ce dernier produit des fichiers de log qui sont plus facilement portables sur les fuseaux horaires, et qui ne souffrent pas de discontinuités lors du changement d'heure.

### **SELECTION D'UN SERVEUR SNTP**

Lorsque vous configurez le client SNTP, plusieurs options s'offrent à vous lors de la sélection d'un serveur.

Si vous disposez d'un serveur de temps Windows ou Unix dans le cadre de votre infrastructure réseau, vous devez finir par synchroniser sur cette source pour garantir une synchronisation sur l'ensemble de l'entreprise. Si plusieurs maîtres sont synchronisés sur le même réseau, il vous est plus simple de nommer l'un d'entre eux en tant que périphérique maître afin de définir l'heure d'été et de faire synchroniser ce maître seul sur la source de l'heure de l'entreprise. Vous pouvez ensuite configurer les autres périphériques pour qu'ils synchronisent sur le périphérique maître, puis activer la fonction DST liée afin de diffuser le paramètre Heure d'été au sein de votre usine.

Si aucune source de l'heure d'entreprise n'est disponible, vous pouvez choisir de nommer un seul maître en tant que point là où un opérateur définit l'heure, puis faire synchroniser les autres maîtres sur cette source. Sinon, si votre installation fournit un accès TCP / IP sur Internet via l'Ethernet ou une connexion par modem, vous pouvez configurer le client SNTP pour qu'il synchronise sur un serveur de temps public. Exemple : 192.6.15.28, qui est l'adresse IP actuelle d'un serveur de temps public fourni par NIST.

Une liste des autres serveurs se trouvent à l'adresse :

<http://support.microsoft.com/kb/262680>

Notez que comme Crimson utilise une adresse IP et pas un nom d'hôte pour désigner le serveur SNTP, il perd la connexion avec tous les serveurs qui sont déplacés sur une nouvelle adresse réseau. Ême si ces déplacements sont très rares, ils dépassent votre contrôle et celui de Red Lion. L'utilisation d'une source de l'heure d'entreprise qui accède à sa propre source via le serveur DNS est par conséquent préférable !

### **CONFIGURATION DES FUSEAUX HORAIRES**

Comme nous venons de l'expliquer, une unite Crimson doit reconnaître le fuseau horaire actuel s'il doit utiliser la gestion de l'heure avancée. Ces informations peuvent être fournies au maître de deux façons : La méthode la plus simple consiste à utiliser la commande Envoyer l'heure du menu Lien du logiciel de configuration de Crimson. Outre la configuration de l'horloge du maître, cette commande envoie également le fuseau horaire actuel du maître et le statut de l'heure d'été. Le maître enregistre ces données dans une mémoire non volatile et les utilise à partir de ce point. Vous devez vous assurer que le PC contient des informations valides sur l'heure et la date avant de les envoyer vers l'unité !

Une autre méthode consiste à utiliser les variables système `TimeZone` et `UseDST`. La première variable contient le nombre d'heures selon lequel le fuseau horaire local diffère de l'UTC et peut être négatif ou positif. Par exemple, un parameter de `-5` correspond à l'heure de l'Est des Etats-Unis. La deuxième variable contient soit 0, soit 1, selon si l'heure d'été est active. La modification de l'une de ces variables via l'interface utilisateur entraîne le changement d'heure de l'unité pour prendre en compte les nouveaux paramètres. Par exemple, l'activation

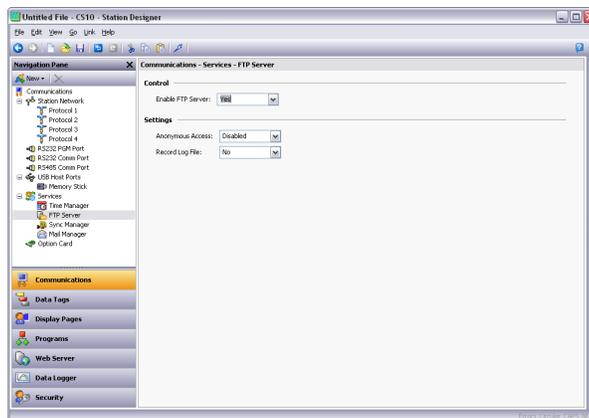
de l'heure d'été avance l'heure d'une heure alors que la désactivation la recule. Une base de données classique doit uniquement exposer `UseDST` pour la modification par l'utilisateur et même cela peut ne pas être nécessaire si la fonction DST liée décrite ci-dessus est en cours d'utilisation.

## UTILISATION DU SERVEUR FTP

Le serveur FTP Crimson fournit une méthode pour échanger des fichiers entre un appareil Crimson et un ordinateur distant via une application client FTP. Le dispositif Crimson agira en tant que serveur, dans l'attente pour les applications client de se connecter et de télécharger ou de téléverser des fichiers.

## CONFIGURATION DU SERVICE

Le serveur FTP est configuré via l'icône associée dans le Panneau de navigation



Les propriétés suivantes peuvent être configurées...

- La propriété *Accès anonyme* définit les droits, le cas échéant, accordés à un utilisateur FTP en anonyme. Un réglage permettra d'éviter les accès anonymes. Un réglage de lecture seule permettra à l'utilisateur de télécharger des fichiers depuis la carte CompactFlash, mais empêchera les envois. Un réglage de lecture-écriture permettra aux deux chargements et téléchargements.
- Activer le *Journal des fichiers* permet de tenir un journal de toutes les interactions FTP dans le répertoire racine de la carte CompactFlash. Ce fichier peut être utile lors du débogage des opérations FTP, mais il aura tendance à se dégrader légèrement les performances.

## SECURITE FTP

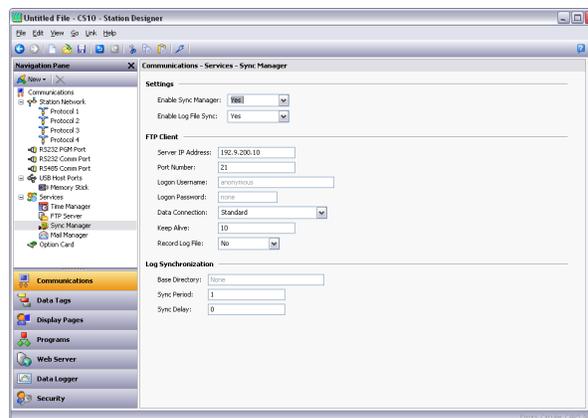
Comme le serveur FTP peut fournir un accès complet à la carte CompactFlash, il est grandement recommandé que vous utilisiez le Gestionnaire de Sécurité pour définir des combinaisons de nom d'utilisateur et mot de passé spécifiques et d'accorder à ces utilisateurs les droits d'accès appropriés. En général, vous devriez éviter d'accorder l'accès anonyme, et ne devez pas autoriser les écritures en anonyme.

## SYNCHRONISATION DES FICHIERS

Le Gestionnaire de synchronisation peut être utilisé pour échanger des fichiers entre un appareil Crimson et un serveur FTP. Cette installation peut être utilisée pour synchroniser les fichiers avec un ordinateur serveur, automatiquement ou sur demande, fournissant ainsi une alternative à l'accès au fichier log via le serveur web, et qui permette des transferts sans assistance de fichiers à partir de plusieurs stations à un point central. (Notez que bien qu'il soit appelé le Gestionnaire de synchronisation pour des raisons historiques, ce service est en fait basé sur un client à des fins générales de ftp qui peuvent également être utilisés pour effectuer d'autres transferts FTP, ou non les fichiers journaux sont synchronisés).

### CONFIGURATION DU SERVICE

Le Gestionnaire de synchronisation est configuré via l'icône associée dans le Panneau de navigation...



### CLIENT FTP

Les propriétés suivantes concernent le client FTP...

- La propriété *Activer la synchronisation* est utilisée pour activer le client FTP. Le client peut être activé sans permettre effectivement de synchronisation, lui permettant d'être utilisé pour le transfert de fichiers via les fonctions manuelles `FtpFilePut()` et `FtpFileGet()`.
- La propriété *Activer le fichier journal* est utilisée pour activer la synchronisation réelle. Voir la section suivante pour des détails sur les autres paramètres relatifs à cette fonctionnalité.
- La propriété *Adresse IP du serveur* est utilisée pour indiquer l'adresse IP du serveur.
- La propriété *Numéro de port* est utilisée pour indiquer le port TCP sur lequel le service client FTP tentera de se connecter. La valeur par défaut convient à la plupart des applications, comme la plupart des serveurs d'écoute sur le port 21.
- Les *Noms d'utilisateur* et *Mot de passe* sont les pouvoirs qui sont soumis au serveur lorsque la connexion est établie. Les deux sont généralement sensibles à la casse, bien que cela dépende de l'application serveur. Pour la connexion anonyme, laissez le nom d'utilisateur sur sa valeur par défaut, et laissez le mot de

passer vierge ou entrez votre adresse électronique comme une formalité pour le serveur fourni.

- La *Connexion de données* offre le choix entre le standard et le mode PASV. Vous pouvez activer le mode PASV pour que le client FTP initie toutes les connexions de données plutôt que d'attendre les connexions entrantes sur le serveur. Ce mode est parfois nécessaire lorsque l'on travaille hors firewall ou lors d'opérations par l'intermédiaire de certaines formes de traduction d'adresses réseau. En règle générale, il est également utilisé lors du travail via une connexion modem GPRS.
- La durée *Keep Alive* est la période durant laquelle la connexion FTP doit être gardée en vie en cas de transferts supplémentaires sont nécessaires. Une valeur de zéro va fermer la connexion dès que le courant de transfert a été achevé. Des valeurs non nulles permettent un fonctionnement plus efficace lors du transfert de fichiers multiples.
- La propriété *Enregistrement du fichier journal* peut être utilisée pour tenir un journal de toutes les interactions FTP dans le répertoire racine de la carte CompactFlash. Ce fichier peut être utile lors du débogage des opérations FTP, mais il aura tendance à dégrader légèrement les performances.

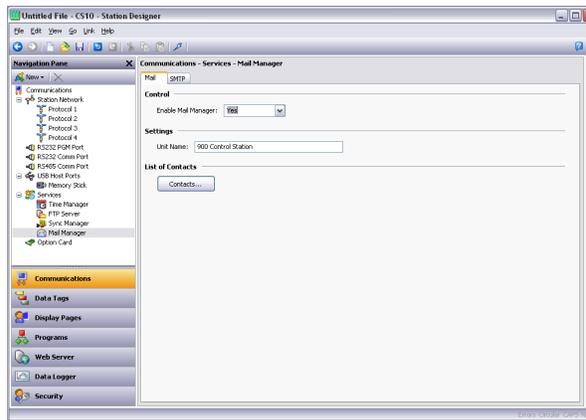
#### SYNCHRONISATION JOURNAL DE DONNEES

Les propriétés suivantes se rapportent spécifiquement à la synchronisation de fichiers log...

- La propriété de la *Base Annuaire* définit le répertoire sur le serveur où les fichiers journaux sont placés. Ce répertoire est relatif à l'espace dossier du serveur FTP, et non à la structure de répertoire sous-jacente au système de classement propre du serveur. Vous aurez généralement à spécifier un répertoire de base différent pour chaque périphérique Crimson lors de la synchronisation à un serveur donné.
- La propriété *Période de Sync.* spécifie combien de fois le client FTP se connecte au serveur et transfère ses fichiers. Elle est mesurée en heures, et est commencée toujours à partir de minuit, ainsi la sélection d'une valeur de trois se traduira par des transferts à minuit, 3h00, 6h00 et ainsi de suite.
- La propriété *Délai de Sync.* définit un décalage en minutes de l'heure légale au cours de laquelle les transferts de fichiers se produiront. Cette propriété peut être utilisée pour permettre à plusieurs terminaux de parler à un serveur sans que tous les transferts de fichiers se produisent à la fois, et ainsi surchargent les capacités de la cible.

## UTILISATION DE LA MESSAGERIE ELECTRONIQUE

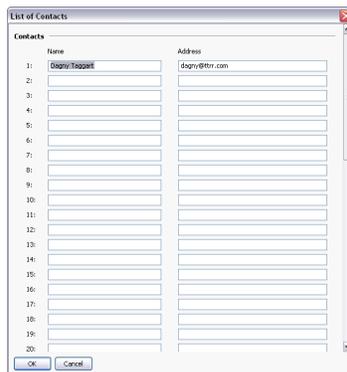
Crimson peut être configuré pour envoyer des messages électroniques lorsque les conditions d'alarme sont présentes ou lorsque des notifications doivent être fournies pour d'autres événements du système. Les méthodes utilisées pour envoyer le courrier électronique sont configurées via le gestionnaire de Courrier...



Les propriétés de l'onglet Général sont utilisées pour activer ou désactiver le Gestionnaire de Messages et pour fournir un nom au périphérique sur lequel Crimson fonctionne. Ce nom est utilisé dans les messages électroniques pour identifier l'auteur du message. Les applications utilisent généralement le nom de la machine sur laquelle le périphérique est installé ou le nom du site qu'il contrôle.

### AJOUT DE CONTACTS

Le bouton contact peut être accédé via le carnet d'adresses de Crimson...

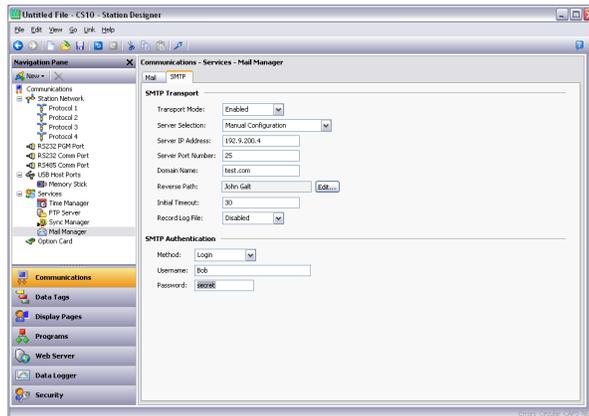


Chaque entrée permet l'affichage d'un nom et d'une adresse. L'adresse doit être inscrite dans un format correspondant au mode de transport requis. Par exemple, les noms SMTP peuvent être dans le format usuel `nom@domaine` tandis que les noms SMS doivent être saisis dans un format téléphonique international sans signe plus. Les adresses email multiples peuvent être saisies si elles sont séparées par des points virgules, permettant ainsi la création de listes de diffusion simples.

### CONFIGURATION DU PROTOCOLE SMTP

L'onglet SMTP permet de configurer le protocole SMTP (Simple Mail Transport Protocol). C'est le protocole standard utilisé pour envoyer des messages sur Internet ou sur les autres réseaux TCP / IP. Les adresses SMTP suivent la même norme habituelle `name@domaine`.

Les options de configuration du transport SMTP sont affichées ci-dessous...



## SMTP TRANSPORT

- La propriété *Mode de transport* permet d'activer ou de désactiver le transport. Notez que le Gestionnaire de Messages doit être activé via l'onglet Général avant que le transport SMTP ne puisse être activé. Notez aussi que le protocole SMTP ou SMS doit être activé pour que le Gestionnaire de Messages puisse envoyer des messages.
- La propriété *Sélection Serveur* permet de définir la façon dont le transport localise un serveur SMTP. Si Sélection Manuelle est utilisée, la propriété *Adresse IP Serveur* doit être utilisée pour créer manuellement un serveur. Si Configuré via DHCP est sélectionné, le port Ethernet de l'unité doit être configuré pour utiliser le protocole DHCP et le serveur DHCP du réseau doit être configuré pour créer un serveur SMTP via l'option 69.
- La propriété *Adresse IP Serveur* permet de créer un serveur SMTP lorsque la sélection du serveur manuel est activée. Le serveur doit être configuré pour accepter les messages électroniques à partir du G3 et pour les relayer si l'application le nécessite.
- La propriété *Numéro de port Serveur* permet de définir le numéro de port TCP qui est utilisé lors des sessions SMTP. La valeur par défaut est 25. Cette valeur est adaptée pour la majorité des applications et ne nécessite qu'un réglage si le serveur SMTP a été reconfiguré pour utiliser un autre port.
- La propriété *Nom de domaine* permet de spécifier le nom de domaine qui sera passé au serveur SMTP dans la commande HELO. La grande majorité des serveurs SMTP ignorent cette chaîne. Dans le cas improbable où votre serveur SMTP tente de rechercher un DNS pour confirmer l'identité de son client, vous devrez peut-être entrer des données appropriées à votre configuration DNS.
- La propriété *E-mail de retour* permet de spécifier l'adresse électronique qui est fournie pour l'auteur des messages envoyés par le G3. Cette propriété comprend un nom complet et une adresse électronique. Puisque le G3 n'est pas en mesure de recevoir des messages, l'adresse électronique est souvent définie sur quelque chose qui renvoie un message « Non remis » si une réponse est envoyée.

- La propriété *Temps Max Initial* permet de spécifier le temps d'attente où le client de la messagerie attend que le serveur SMTP envoie sa bannière de bienvenue. Certains serveurs Microsoft essaient de négocier une authentification spécifique à Microsoft avec les clients de la messagerie, retardant ainsi le point sur lequel la bannière s'affiche. Vous devrez peut-être rallonger ce délai de 2 minutes ou plus lorsque vous travaillerez avec de tels serveurs.
- La propriété *Enregistrement du journal* peut être active et demander de tenir un journal de toutes les interactions SMTP dans le répertoire racine de la carte CompactFlash. Ce fichier peut être utile lors du débogage des opérations SMTP, aura tendance à se dégrader légèrement les performances.

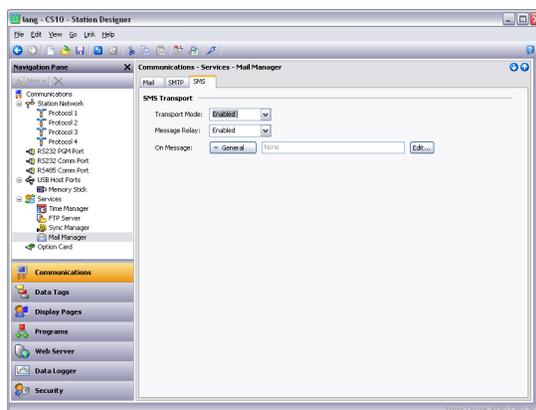
### AUTENTIFICATION SMTP

- La propriété *Méthode* indique le type d'authentification tentée par le client. Une sélection *Digest* insistera sur une technique d'authentification qui envoie le mot de passe sous forme cryptée, et qui va sauter d'authentification si le serveur ne supporte pas une telle méthode. Une sélection de *base* va tenter d'utiliser la technique sûre, mais la volonté de repli de transmission codée trivialement si nécessaire. Une sélection de *Néant* tentera pas d'authentifier. Votre serveur peut ou nécessite pas d'authentification. Contactez votre administrateur réseau ou votre fournisseur de messagerie pour plus de détails sur le réglage qui doit être utilisé pour votre serveur.
- Les propriétés de *Nom d'utilisateur / Mot de passe* sont des informations facultatives pour le processus d'authentification décrites ci-dessus.

### CONFIGURATION DU PROTOCOLE SMS

L'onglet SMS permet de configurer le service SMS (Short Messaging Service). Ce transport est utilisé pour envoyer des messages texte aux téléphones portables via un modem GSM. Les adresses électroniques SMS comprennent un numéro de téléphone au format international, mais sans le signe plus. Par exemple, l'adresse 17175551111 enverrait un message aux États-Unis au numéro (717) 555-1111.

Les options de configuration du service SMS sont affichées ci-dessous...



- La propriété *Mode de transport* permet d'activer ou de désactiver le transport.

La propriété *Relais des Messages* permet d'activer ou de désactiver la fonctionnalité de relais SMS de Crimson. Si cette fonctionnalité est activée, un utilisateur qui reçoit un message SMS envoyé à plusieurs destinataires peut répondre à ce message et le périphérique Crimson relaie le message aux autres destinataires. Ainsi, un service de conférence simple est offert entre les destinataires des messages.

- La propriété *Sur message* est utilisée pour définir une action devant être exécutée à chaque fois qu'un message est reçu. Des variables système locales appelées `Données` sont définies au sein de l'action, permettant d'accéder au message lui-même. Le numéro source du SMS est préfixé au message, une virgule le séparant du message lui-même.



# PARTAGE DE PORT

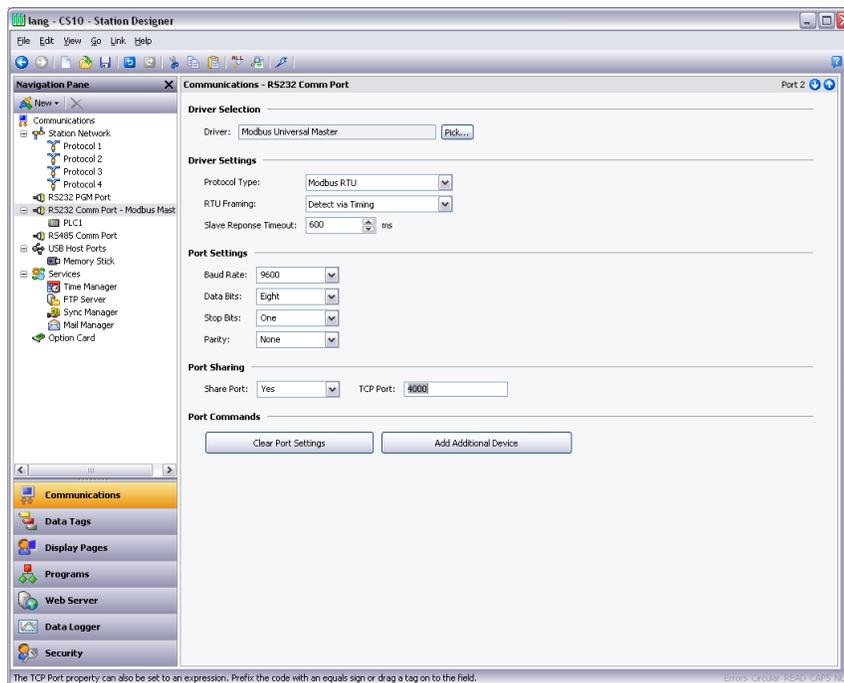
Crimson fournit ce qu'on appelle une fonction de « partage de port » qui permet d'établir des connexions série physiques ou virtuelles sur n'importe quel périphérique connecté dessus. Par exemple, vous pouvez utiliser un panneau d'utilisation avec un petit automate programmable, mais comme celui-ci ne dispose que d'un seul port série, il se peut que vous deviez en permanence échanger les câbles lorsque vous modifiez le programme de l'API. En partageant le port de communication pour vous connecter à l'API, vous pouvez envoyer directement des données au contrôleur, depuis un port série ou via une connexion TCP / IP.

## ACTIVATION TCP / IP

La première étape de la configuration lors de l'utilisation du partage des ports consiste à activer le port Ethernet de Crimson, tel que décrit dans le chapitre précédent. Même si vous choisissez de ne pas utiliser la fonction Port série virtuel, le partage local des ports repose sur le protocole TCP / IP qui n'est disponible que si l'Ethernet est activé.

## PARTAGE DU PORT

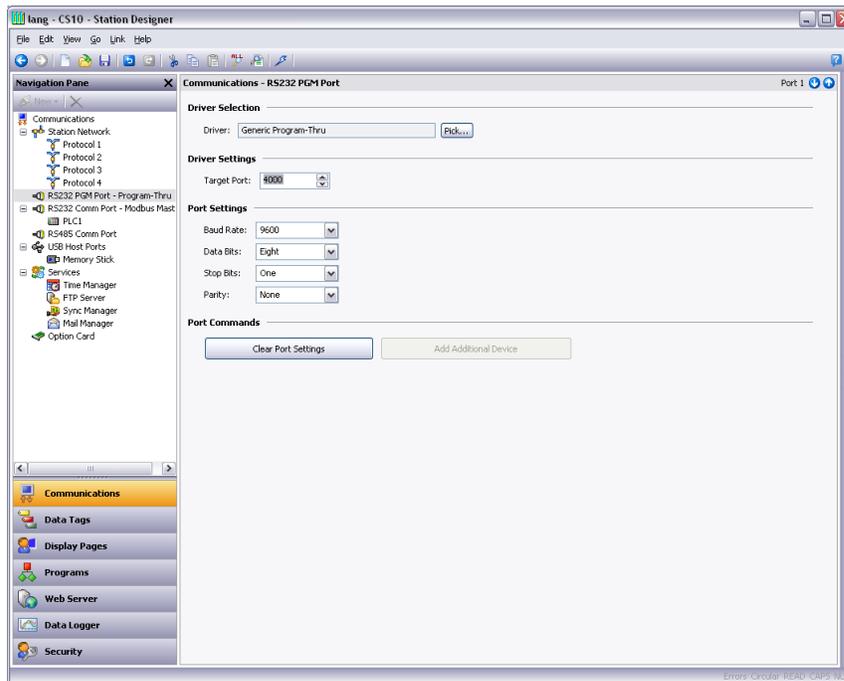
La prochaine étape consiste à partager le port, ce que vous pouvez effectuer en sélectionnant Oui dans la propriété « Partage du port » et en entrant un numéro de port TCP / IP adapté. Ce numéro représente le port virtuel que vous utilisez pour exposer le port série à un accès via le protocole sur TCP / IP...



Si vous laissez le paramètre du port sur zéro, un numéro de 4 000 plus l'index logique du port sont utilisés. Vous pouvez utiliser tout numéro qui n'est pas déjà utilisé par un autre protocole TCP / IP. Si vous êtes à court d'idées, nous vous conseillons des nombres entre 4 000 et 4 099.

## CONNEXION VIA UN AUTRE PORT

Si vous souhaitez utiliser un autre port sur le maître pour router les données vers le port partagé, vous devez sélectionner le pilote Generic Program Thru pour ce port, puis configurer ce pilote avec le numéro de port TCP / IP du port série que vous avez partagé. Dans l'exemple ci-dessous, nous routons les données à partir du port de programmation vers un API qui est connectée via le port de communication RS-232...



Notez que Débit en bauds et les autres paramètres de port ne doivent pas forcément être identiques à ceux du port que nous partageons, Crimson prenant en charge la conversion. La seule exception intervient lorsqu'un périphérique transmet de grandes quantités de données sans recevoir aucune réponse. Dans ce cas, le périphérique en charge des transmissions les plus importantes doit être utilisé à un taux de Baud plus important que celui du périphérique les recevant. Dans le cas contraire, Crimson pourrait ne pas disposer de suffisamment de mémoire pour mettre les données en tampon dans l'attente de leur retransmission.

Dans cet exemple, pour utiliser le port partagé, vous devez connecter un port série disponible de votre PC au port de programmation du maître, puis configurer le logiciel de programmation de l'API pour qu'il communique avec le port COM de votre PC. Dès que le PC commence à communiquer avec l'API, les communications entre le maître et l'API sont suspendues et les deux ports du maître (celui du PC et de l'API) sont « connectés » comme si le PC communiquait directement avec l'API. Si aucune donnée n'est transférée pendant plus d'une minute, les communications entre le maître et l'API reprennent.

## CONNEXION VIA ETHERNET

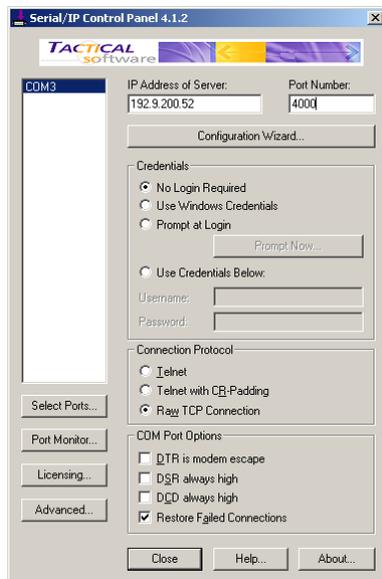
Au lieu d'utiliser un port série sur votre PC et sur Crimson, vous pouvez utiliser un utilitaire tiers pour créer ce qu'on appelle des ports série virtuels sur votre ordinateur. Ils apparaissent comme des ports COM physiques aux applications, mais en fait, ils envoient et reçoivent des données sur un périphérique distant par le protocole TCP / IP. En installant l'un de ces

utilitaires et en le configurant pour qu'il s'adresse au maître, vous pouvez disposer d'un accès série sur tous les périphériques connectés au maître sans câblage supplémentaire. En effet, il est complètement inutile d'avoir des ports série disponibles sur le PC, ce qui est très précieux lorsque vous travaillez avec des ordinateurs portables modernes sur lesquels un port COM série est souvent une option onéreuse.

Plusieurs utilitaires de port série virtuels tiers sont disponibles. Côté freeware, une société nommée HW Group (<http://www.hw-group.com>) propose un utilitaire appelé HW Virtual Serial Port. Il existe également plusieurs autres freeware mais la plupart d'entre eux semblent provenir de la même base source. Côté commercial, une société appelée Tactical Software (<http://www.tacticalsoftware.com>) propose la solution Serial / IP pour environ 100 \$ le port.

Même si les différents freeware ont sans aucun doute de nombreux clients satisfaits, nous avons découvert que ces pilotes présentaient des problèmes de stabilité occasionnels sur certains PC. La solution Serial / IP de Tactical Software est ainsi le seul package que nous sommes en mesure de prendre en charge et les informations suivantes supposent que vous utilisez ce package.

Pour créer un port série virtuel, ouvrez l'écran de configuration Serial / IP, puis sélectionnez le nom du port COM série que vous souhaitez définir. En général, c'est le premier port série libre après ceux qui ont été alloués aux ports physiques et aux modems etc ... installés sur votre PC. Saisissez ensuite l'adresse IP du maître puis le numéro de port TCP / IP que vous avez alloué lors du partage du port. L'exemple ci-dessous est configuré comme décrit ci-dessus dans ce document. Pour finir, assurez-vous que Connexion TCP Raw est sélectionnée, puis fermez la boîte de dialogue Serial / IP.



Vous pouvez maintenant configurer n'importe quels logiciels Windows pour qu'il utilise le port COM série virtuel qui vient d'être créé pour effectuer un téléchargement. Lorsque le logiciel ouvre la connexion, Crimson suspend les communications sur le port partagé, puis les données sont ensuite échangées entre le logiciel de programmation de l'API distant, comme s'ils étaient directement connectés ! Une fois que le port est fermé ou si aucune donnée n'est transférée pendant une minute, les communications reprennent.

Notez qu'en supposant que vous avez acheté le nombre de licences approprié pour la solution Serial / IP, vous pouvez créer autant de ports virtuels que vous le souhaitez. Ainsi, vous pouvez être connecté à plusieurs périphériques à partir du même PC en téléchargeant sur chacun par le biais de son package de programmation respectif, le tout sans connecter ou déconnecter un seul câble. Cette fonctionnalité est très utile lorsque vous disposez de plusieurs périphériques dans un système complexe.

### **PORTS VIRTUELS PURS**

In some circumstances, you may want to use a spare serial port on a Crimson-based device to provide access to a remote device that is not otherwise referenced in your database, effectively using the spare port as a remote serial server. To do this, configure the port in the usual way, selecting the Virtual Serial Port driver for that port. Then, share the port as described above, exposing it via TCP / IP. The Virtual Serial Port driver performs no comms activity of its own, but still allows the device to be shared for remote access.

### **LIMITATIONS**

Notez que certains logiciels de programmation d'API peuvent ne pas fonctionner avec des ports partagés de façon virtuelle ou physique. Les problèmes à surveiller sont : les délais d'attente court qui ne permettent pas à Crimson de relayer les données sur l'API ; une dépendance à l'envoi de signaux d'interruption ou à la manipulation de lignes d'établissement de liaison du matériel ; ou un accès au port de type DOS tel que le logiciel ne peut pas « voir » les ports série virtuels. Heureusement, ces problèmes sont rares et la majorité des logiciels communiquent sans problème comme s'ils étaient directement connectés à l'API en question.



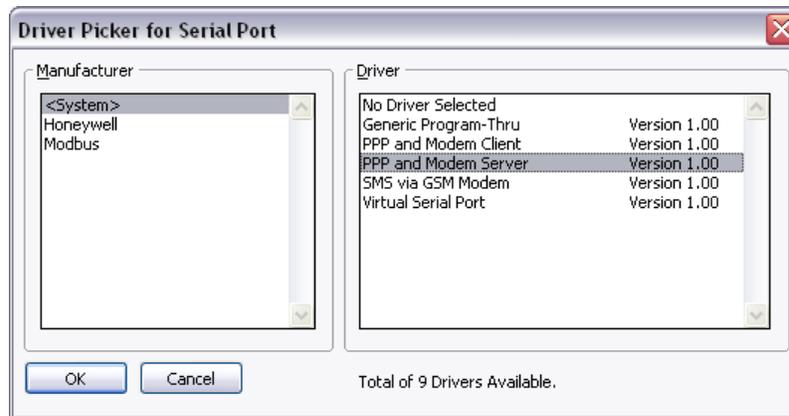
# UTILISATION DES MODEMS

Cette section décrit la configuration de Crimson pour qu'il fonctionne avec des modems ou des connexions série directes sur des ordinateurs dotés du système d'exploitation Windows.

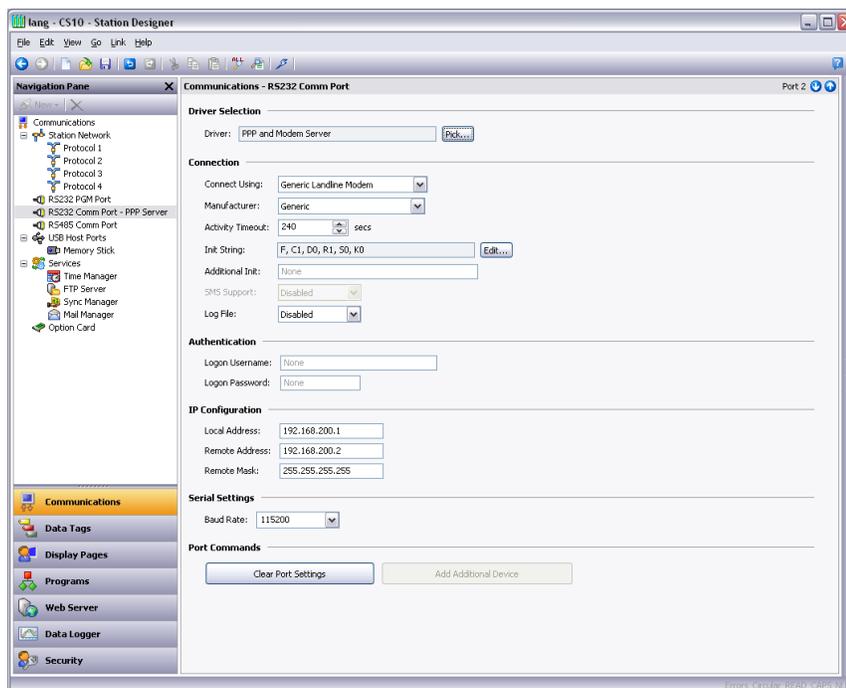
Note that Crimson's modem support is entirely based upon the Point-To-Point Protocol, otherwise known as PPP. While protocols such a Modbus allow a single conversation to occur between any two devices, PPP is more akin to an Ethernet connection in that it allows an unlimited number of logical connections to exist on a single physical link. A single PPP connection can thus allow simultaneous access to the panel's TCP / IP download facility, its Serveur web, its shared serial ports, and to any TCP / IP protocols that have been defined.

## AJOUT D'UNE CONNEXION D'APPEL ENTRANT

Pour ajouter une connexion d'appel entrant à votre base de données, ouvrez la fenêtre Communications et sélectionnez le port série sur lequel la connexion sera établie. Cliquez sur le Bouton Choisir dans les propriétés pilote et sélectionnez *Modem Serveur PPP* dans la section Système...



Le Panneau d'édition affiche alors la configuration Modem...



Le modem est doté des options de configuration suivantes...

- La propriété *Se connecter via* permet de sélectionner le périphérique physique qui est utilisé pour établir la connexion. Les périphériques pris en charge à ce niveau sont des connexions série directes sur des ordinateurs dotés d'un système d'exploitation Microsoft Windows, des modems à ligne terrestre générique qui implémentent le jeu de commandes Hayes ainsi que le mode GSM Telit GM-862. Pour les connexions d'appel entrant, le périphérique Telit doit être configuré en mode Données basculées sur le circuit.
- La propriété *Constructeur* permet de sélectionner le fabricant ou le modèle pour chaque configuration développée et stockée au sein de Crimson. Laisser ce paramètre sur l'option Générique permettra de personnaliser les paramètres liés aux chaînes d'initialisation. Veuillez consulter votre support technique pour connaître les paramètres spécifiques à un modèle de modem particulier.
- La propriété *Arrêt après inactivité de* permet de définir la durée où le maître n'envoie aucun paquet sur le lien PPP pour que la connexion soit terminée. Pour les connexions d'appel entrant, on suppose que le périphérique de connexion est convivial. Par conséquent, aucun effort n'est fourni pour filtrer les paquets facultatifs qui peuvent avoir pour conséquence que le lien reste actif sur de longues périodes. Notez que même si vous voulez une connexion permanente, vous devez entrer un délai d'attente approprié pour permettre de détecter les liens morts. Cela implique que les « connexions permanentes » peuvent être perdues à certains moments, mais comme le client rétablit immédiatement le lien, ce n'est pas un problème.
- La chaîne *Init Supplémentaire* est utilisée avec les liens non directs et fournit une série de commandes AT qui servent à initialiser le modem. Le préfixe initial AT

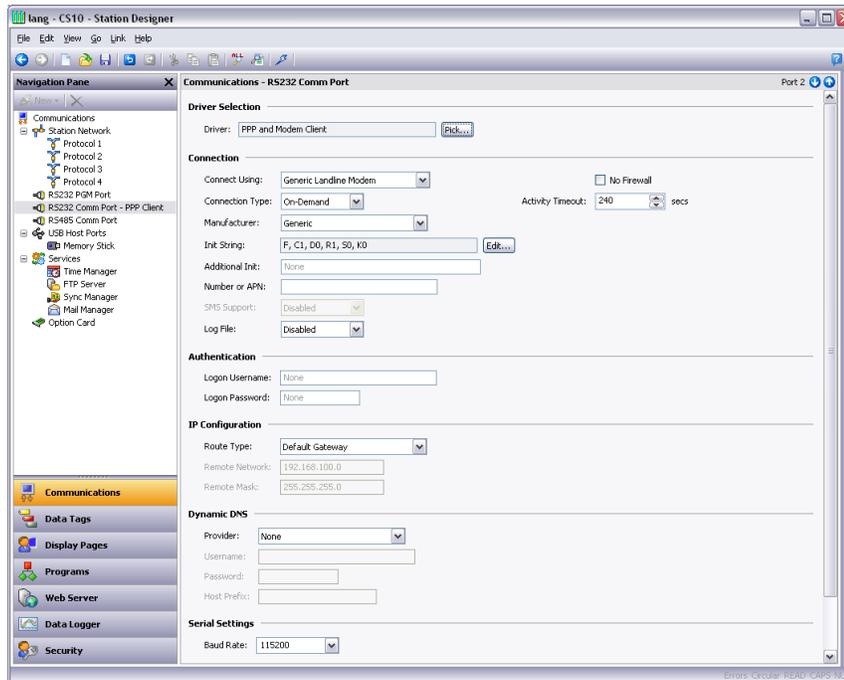
n'est pas requis. Vous pouvez associer plusieurs commandes simplement en les plaçant l'une après l'autre. La chaîne exacte requise pour votre modem dépend de son logiciel interne. Ainsi, si vous contactez le support technique pour obtenir une assistance, assurez-vous que vous disposez des informations exactes du modèle et de la marque.

- La propriété *Support SMS* permet d'activer la messagerie SMS (Short Message Service) lors de l'utilisation d'un modem GSM. Pour que la messagerie SMS fonctionne correctement, vous devez activer le transport SMS à l'aide de l'icône Mail qui se trouve dans la fenêtre Communications, tel que décrit dans ce manuel.
- Les propriétés *Nom d'Utilisateur* et *Mot de Passe* permettent de définir les informations d'identification que le client distant doit fournir afin d'être autorisé à se connecter à ce périphérique. Le nom d'utilisateur ne respecte pas la casse, mais le mot de passe la respecte. L'implémentation du protocole PPP de Crimson demande à son homologue d'utiliser l'authentification CHAP pour éviter la transmission ou la réception des mots de passe en clair, mais a recours au protocole PAP si le client distant ne prend pas en charge l'authentification CHAP.
- La propriété *Adresse locale* permet de définir l'adresse IP attribuée à des fins locale de la connexion. Ainsi, vous utiliserez cette adresse pour rentrer en communication par Ethernet ou Internet ou par un modem avec les CSMSTRSX ou GT ou Data Station Plus ou G3. Veuillez noter qu'elle ne doit pas être identique à l'adresse IP du port Ethernet du maître (CSMSTRSX ou GT ou Data Station Plus ou G3) car chaque interface IP physique doit posséder une adresse IP distincte. La valeur par défaut fonctionne dans la plupart des cas sauf si la conception de votre réseau exige que vous utilisiez un paramètre différent.
- La propriété *Adresse distante* permet de définir l'adresse IP attribuée à la connexion distante. Elle est utilisée avec la propriété *Masque distant* pour déterminer les paquets qui seront routés vers cette connexion. Pour la plupart des applications, un masque de 255.255.255.255 est utilisé, demandant alors à Crimson d'envoyer uniquement via cette interface les paquets à destination directe du client distant. En revanche, un masque de 0.0.0.0 permet de transférer au client distant tous les paquets qui ne correspondent pas expressément à une autre interface, vraisemblablement pour un transfert ultérieur vers l'hôte visé. Les masques intermédiaires peuvent être utilisés pour contrôler l'envoi exact des paquets.

### **AJOUT D'UNE CONNEXION D'APPEL SORTANT**

Les connexions d'appel sortant sont configurées de la même façon que ce que nous venons de décrire ci-dessus sauf que le pilote *Modem Client PPP* doit être sélectionné pour le port requis.

Les options de configuration de ce modem sont affichées ci-dessous...



Le modem est doté des propriétés suivantes qui sont distinctes des propriétés des connexions d'appel entrant...

- La propriété *Se connecter via* s'adresse aux connexions d'appel entrant avec en plus, la prise en charge des connexions GPRS via un modem GSM. Ces connexions diffèrent des connexions CSD car elles atteignent des vitesses bien supérieures et elles sont généralement chargées sur la base de la quantité de données transférées plutôt que sur la durée de la connexion. Ainsi, les connexions GPRS peuvent être configurées pour une connexion permanente sauf si un temps d'arrêt est nécessaire pour permettre le transfert des messages SMS.
- La propriété *Aucun pare-feu* permet de désactiver la protection par pare-feu qui est par ailleurs fournie par les connexions d'appel sortant. Cette protection empêche les connexions entrantes d'être établies sur cette interface et empêche le maître (CSMSTRSX ou GT ou Data Station Plus ou G3) d'envoyer certains paquets de diagnostic qui pourraient fournir aux pirates des informations sur le système ou être utilisés par les attaquants pour maintenir une connexion active en l'absence du transfert de données réel. Si vous êtes directement connecté à Internet par le biais de cette connexion, vous ne devez normalement pas désactiver le pare-feu. Le pare-feu doit être désactivé uniquement pour les connexions sur les réseaux d'entreprise ou sur les autres environnements contrôlés.
- La propriété *Type de connexion* permet d'indiquer si vous souhaitez que cette connexion soit conservée en permanence ou si vous souhaitez qu'elle soit établie automatiquement lorsqu'une tentative de transférer les données vers des hôtes qui peuvent être atteints via cette interface est effectuée. Si vous sélectionnez la

connexion à la demande, vous devez spécifier le délai d'attente précédant la fin du lien si aucun paquet n'a été transmis par le G3.

- Les propriétés *Nom d'Utilisateur* et *Mot de Passe* permettent de définir les informations d'identification qui sont envoyées au serveur distant lors de la tentative d'initialisation de cette connexion. Le nom d'utilisateur ne respecte pas la casse, mais le mot de passe la respecte. L'implémentation du protocole PPP de Crimson demande à son homologue d'utiliser l'authentification CHAP pour éviter la transmission ou la réception des mots de passe en clair, mais a recours au protocole PAP si le serveur distant ne prend pas en charge l'authentification CHAP.
- La propriété *Type de Routage* permet de définir les données qui sont transférées via cette interface. Pour les connexions à la demande, elle définit efficacement l'activation de la connexion. Si *Passerelle par défaut* est sélectionnée, tous les paquets qui ne correspondent pas à l'adresse et au masque réseau de la connexion Ethernet sont envoyés vers cette interface. Notez que dans ce mode, le port Ethernet doit disposer d'un paramètre de passerelle de 0.0.0.0, sinon il prendra tous les paquets et n'en laissera aucun afin d'activer le modem ! Si *Réseau spécifique* est sélectionné, vous devez fournir l'adresse et le masque réseau qui définissent le réseau sur lequel les paquets sont routés.

### **AJOUT D'UNE CONNEXION SMS**

Les connexions SMS sont utilisées lorsque la fonctionnalité de la messagerie écrite est requise, mais lorsque aucune connexion PPP d'appel entrant ou sortant n'est établie. Elles sont configurées comme nous venons de le décrire ci-dessus sauf que le périphérique *SMS via modem GSM* doit être sélectionné pour le port requis.

Les propriétés du périphérique sont un sous-ensemble de celles qui sont fournies pour les connexions d'appel entrant. La prise en charge de SMS est toujours active avec ce pilote, mais une fois encore, pour que la messagerie SMS fonctionne correctement, vous devez activer le transport SMS dans le gestionnaire de courriers.

### **TRAITEMENT DES MESSAGES SMS**

Lorsque la messagerie SMS est activée, Crimson demande au modem GSM de vérifier la présence de messages entrants ou sortants toutes les cinq secondes. Les messages entrants sont transférés au Gestionnaire de Messages qui les transférera le cas échéant aux autres utilisateurs en fonction de sa configuration. Notez que vous pouvez vérifier la présence de messages lorsque le modem est connecté sur une session CSD ou GPRS, ce qui vous permet d'éviter d'utiliser les connexions permanentes lorsque vous travaillez avec le service SMS. Notez également que si plusieurs modems GSM sont configurés, ils peuvent tous recevoir des messages, mais seul le deuxième modem sera utilisé pour l'envoi.

### **VERIFICATION DU STATUT DU MODEM**

Pour déboguer les connexions par modem, Crimson propose la fonction `GetInterfaceStatus()`. Cette fonction prend un seul argument qui est l'index numérique de l'interface requise. L'interface zéro est toujours l'interface de bouclage du maître. Ensuite

vient l'interface Ethernet, si elle est active, suivie des interfaces PPP. Si un système utilise un port Ethernet simple, par exemple, la première interface PPP aura un index de 1.

The function returns a string, which can be interpreted according to the following table...

<b>STATUT</b>	<b>SIGNIFICATION</b>
FERME	L'interface n'a pas encore été initialisée. Cet état se produit uniquement sur une courte période lors du démarrage du système.
INIT	Le modem est en cours d'initialisation. Si la connexion reste sur cet état, c'est qu'il existe probablement des erreurs dans les chaînes d'initialisation qui sont envoyées au modem.
INACTIF	Le lien est inactif. Les modems GSM renvoient un nombre à la fin de la chaîne pour indiquer la force du signal. Le tableau suivant décrit la façon d'interpréter ces valeurs.
SMS	Le modem envoie des messages SMS ou demande au modem de vérifier si de nouveaux messages SMS sont disponibles. Si la messagerie SMS est activée pour un modem, cet état s'affiche pendant une courte période toutes les cinq secondes.
CONNEXION	Le modem est en train d'établir une connexion. Cet état s'affiche en général uniquement pour les connexions clientes et indique qu'un appel est effectué.
ECOUTE	Le modem attend un appel. Cet état s'affiche uniquement pour les connexions serveur. Notez que les modems GSM renvoient également un état INACTIF lorsqu'ils attendent un appel afin d'indiquer la force du signal.
RÉPONSE	Le modem répond à un appel et essaie de négocier le débit en bauds de la connexion. Cet état s'affiche uniquement pour les connexions serveur. Si la connexion est établie, le modem entre l'état CONNECTE.
CONNECTÉ	Le modem a établi une connexion. Cet état persiste pendant une courte durée, le temps que le processus de négociation LCP commence après un petit délai.
NEG LCP	La connexion est en train de négocier les options LCP. Ce processus décide d'un groupe de paramètres du protocole de lien qui sont acceptables pour le client et le serveur.
AUTH	La connexion effectue le processus d'authentification pour garantir que les informations d'identification appropriées de l'utilisateur sont utilisées.
NEG IPCP	La connexion est en train de négocier les options IPCP. Ce processus décide d'un groupe de paramètres du protocole réseau qui sont acceptables pour le client et le serveur.
ACTIF	La connexion est active et les données IP peuvent être échangées.

STATUT	SIGNIFICATION
FIN DE LA COMMUNICATION	Le modem est en cours de déconnexion. Cet état existe uniquement pendant une courte durée avant que le modem ne revienne sur INACTIF.

Les valeurs de la force du signal renvoyées par les modems GSM ont la signification suivante...

VALEUR	FORCE DU SIGNAL
0	-113dBm ou moins
1	-111dBm.
2-30	-109dBm à -52dBm par paliers de 2dBm.
31	-51dBm ou plus.
99	La force du signal ne peut pas être déterminée.

Les telephones portables interprètent généralement ces valeurs comme décrit ci-dessous...

VALEUR	FORCE	NOMBRE DE BARS
5 ou moins	-103dBm ou moins	Une
6 à 9.	-101dBm à -95dBm	Deux
10 à 14.	-93dBm à -85dBm	Trois
15 ou plus.	-83dBm ou plus.	Quatre

## RESOLUTION DES PROBLEMES DE COMMUNICATION MODEMS

Les pilotes de modem différents fournissent un *Fichier journal* de propriété pour ouvrir une session d'échange avec le modem via un fichier de la carte CompactFlash. Ce fichier est utilisé pour le débogage en cours de configuration du modem ou en tentant de trouver les options de configuration appropriée. Assurez-vous de désactiver cette fonctionnalité, une fois la séquence de configuration établie.

## UTILISATION D'INTERFACES MULTIPLES

Crimson supporte jusqu'à deux connexions modem indépendantes. Lorsqu'elles sont combinées avec les ports Ethernet un ou deux permis par le dispositif cible, cela donne un total allant jusqu'à quatre interfaces IP distinctes, qui fonctionneront selon les paramètres de configuration définis pour chaque connexion. Cette section décrit comment ces interfaces multiples interagissent, et comment Crimson va décider où envoyer chaque paquet de données.

### SELECTION DE L'INTERFACE

Chaque interface possède une adresse IP et un masque de réseau, qui sont utilisés pour décider de transmettre les paquets à cette interface. Par exemple, si une interface Ethernet est configurée avec une adresse IP de 192.168.1.0 et un masque réseau 255.255.255.0, tous les paquets pour les adresses IP commençant par 192.168.1 seront envoyés à cette interface. De même, si une connexion en modem demande a une adresse IP distante de 192.168.2.2 et un

masque de réseau de 255.255.255.255, l'envoi d' un paquet à l'adresse 192.168.2.2 se traduira par l'établissement de la connexion

Notez que ce mécanisme ne va jamais envoyer un paquet à une interface unique. Cela implique que les interfaces doivent avoir des adresses réseaux distinctes, telles que définis par leur adresse IP et leur masque de réseau. Si vous violez cette exigence, les paquets ne seront pas routés vers la deuxième interface avec cette adresse de réseau, et de la communication sur ce port va échouer. Par exemple, vous ne devez pas configurer un port Ethernet comme 192.168.100.1 et l'autre comme 192.168.100.2, car les paquets pour le réseau 192.168.100.0 ne seront envoyés au premier port.

#### ROUTE PAR DEFAUT

En outre, une interface unique peut aussi définir une route par défaut, qui sera utilisée pour gérer les paquets qui ne correspondent pas spécifiquement à toute autre interface. La méthode utilisée pour configurer l'itinéraire varie selon le type d'interface est indiquée dans le tableau ci-dessous...

<b>INTERFACE</b>	<b>POUR DEFINER LA ROUTE PAR DEFAUT</b>
Ethernet	Entrez une valeur non nulle pour la propriété de la <i>Passerelle</i> .
Appel entrant	Entrez 0.0.0.0 pour le <i>Masque de distance</i> .
Appel sortant	Sélectionnez la passerelle par défaut pour la propriété <i>Type d'itinéraire</i>

Là encore, une seule interface peut définir un itinéraire par défaut. Par exemple, un opérateur peut être connecté à un certain nombre de périphériques Ethernet en utilisant une adresse IP 192.168.1.0 et un masque réseau 255.255.255.0, sans passerelle définie. Une connexion par modem sur demande peut être configuré pour accéder à un fournisseur de services Internet afin d'envoyer des emails d'alarme. Son *Type de route* est réglé par la passerelle par défaut, ce qui en fait la route de tous les paquets pour les adresses IP qui ne correspondent pas au réseau définis pour le port Ethernet. Le serveur SMTP est configuré en tant que 24.104.0.39, résultant en une connexion dial-out lors d'une tentative est faite pour envoyer un message.

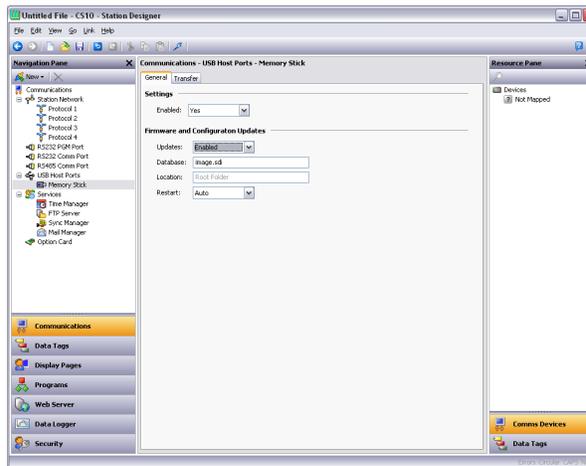
# UTILISATION DE L'HÔTE USB

Si votre périphérique cible possède un ou plusieurs ports USB hôtes, l'icône correspondante dans la catégorie des communications, peut être utilisée pour configurer les appareils pris en charge. Les versions actuelles de Crimson supportent uniquement les périphériques USB.

## SUPPORT CARTE MEMOIRE

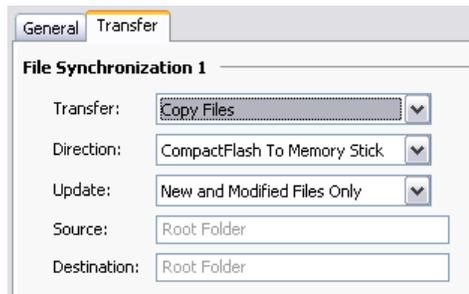
Les dispositifs USB sont configurés via l'icône Memory Stick...

### PROPRIETES GENERALES



- La propriété *Activer* est utilisée pour activer ou désactiver globalement le support carte mémoire
- La propriété *Mises à jour* est utilisée pour configurer le transfert automatique du nouveau firmware et de la base de données pour le répertoire racine de la carte CompactFlash.
- La propriété *Base de données* définit le nom de l'image de base de données copiée dans le fichier `image.ci3` sur la carte CompactFlash. Ce paramètre permet à plusieurs dossiers d'être placés sur une clé unique, chaque dispositif Crimson copiant le fichier qui est adapté à sa propre application.
- La propriété *Emplacement* spécifie l'emplacement sur la Carte Mémoire où l'image de base peut être localisée.
- La propriété *Redémarrage* est utilisé pour indiquer si un redémarrage automatique doit être effectuée une fois que le fichier a été copié. L'activation de cette propriété permet la prise en charge immédiate des informations de l'image de base de données.

## PROPRIETES DE TRANSFERT



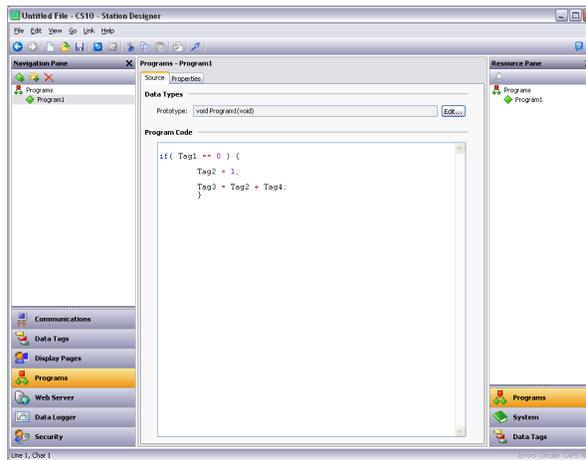
The screenshot shows a software interface with two tabs: 'General' and 'Transfer'. The 'Transfer' tab is active. Below the tabs is a section titled 'File Synchronization 1'. It contains five settings:

- Transfer:** A dropdown menu with 'Copy Files' selected.
- Direction:** A dropdown menu with 'CompactFlash To Memory Stick' selected.
- Update:** A dropdown menu with 'New and Modified Files Only' selected.
- Source:** A text input field containing 'Root Folder'.
- Destination:** A text input field containing 'Root Folder'.

- La propriété *Transfert* pour chaque groupe de synchronisation définit la fonction qui doit être effectuée. L'information peut soit être copiée ou déplacée, et l'opération peut être simplement appliqué sur les fichiers dans le dossier spécifié, ou également à des sous-dossiers et à leur contenu de façon récursive.
- La propriété *Direction* spécifie la direction du transfert.
- La propriété *Mise à jour* est utilisée pour indiquer si les fichiers qui semblent être déjà présents sur l'appareil cible doivent être copiés dans tous les cas, ou si seuls les fichiers nouveaux et modifiés doivent être transférés. Crimson utilise l'heure du fichier-tampon et la taille afin de décider si le dossier devrait être traité.
- Les propriétés *Source* et *Destination* sont utilisées pour indiquer les dossiers sur la source et dispositifs cible où les fichiers doivent être situés.

# UTILISATION DE LA CATÉGORIE PROGRAMMER

Les sections précédentes de ce manuel décrivent l'utilisation des actions pour effectuer toutes sortes d'opérations une fois que vous avez appuyé sur des touches ou modifié des étiquettes de données. Si vous devez effectuer une action qui est trop complexe pour s'adapter sur une ligne unique ou qui exige une logique de prise de décision plus complexe, vous la catégorie Programmer pour créer et manipuler des programmes.



## LISTE DES PROGRAMMES

La liste des programmes du Panneau de navigation est un moyen classique de navigation qui peut être utilisé pour créer, supprimer, renommer et organiser les programmes. Notez que les programmes peuvent être regroupés en dossiers, et que l'icône de chaque programme peut afficher trois états : vert, indiquant un programme qui a été traduit et validé, jaune, indiquant un programme qui a été édité mais pas encore traduit, ou rouge, indiquant un programme qui contient une ou plusieurs erreurs.

## USAGE DES PROGRAMMES

Vous pouvez trouver tous les éléments qui font référence à un programme donné par un clic droit sur cet élément dans le Panneau de navigation puis en sélectionnant la commande Rechercher. Les éléments résultant seront placés sur la liste globale Résultats de la recherche, et peuvent être consultés par le biais de la combinaison de touches **F4** et **MAJ + F4**. La liste elle-même peut être affichée ou masquée en appuyant sur **F8**.

## EDITION DES PROGRAMMES

Pour modifier un programme, il suffit d'éditer le texte du programme en utilisant l'onglet source affichée dans le Panneau d'édition. Vous remarquerez que l'icône du programme vire au jaune dès que vous commencez à taper, vous indiquant que vous avez apporté des modifications qui doivent encore être traduites. Vous noterez également que l'éditeur de programme Crimson effectue une coloration de la syntaxe, une auto-indentation et une variété

d'autres caractéristiques appropriées pour un éditeur de code. Les options de l'éditeur peuvent être configurées par un clic droit sur le Panneau d'édition et en sélectionnant la commande appropriée dans le menu contextuel.

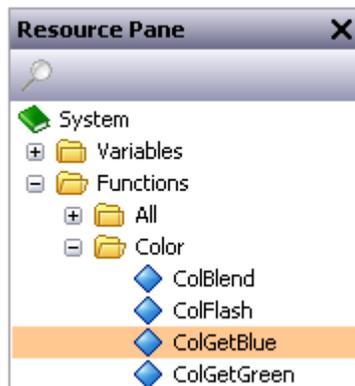
Lorsque vous avez fini d'écrire votre programme, appuyez sur la combinaison **CTRL + T** ou sélectionnez le bouton Traduire de la barre d'outils. Le programme va alors vérifier les erreurs. Si une erreur est trouvée, une boîte de dialogue s'affiche et l'icône du programme devient rouge. Le curseur se déplace également vers la position de l'erreur. Si aucune erreur n'est détectée, un son de carillon est joué et l'icône du programme devient verte, indiquant que le programme a été traduit dans une forme appropriée pour l'exécution au sein de l'appareil cible.

## AIDE

Lors du travail dans le Panneau d'édition, un raccourci est disponible pour fournir de l'aide sur les fonctions du système. Placez votre curseur à l'intérieur ou à la fin du nom de fonction et appuyez sur la touche **F1** pour afficher des informations sur le fonctionnement de la fonction, les arguments et type de retour. Vous pouvez également appuyer sur **F1** après avoir tapé le nom d'une fonction afin d'accéder aux mêmes informations.

## PANNEAU DE RESSOURCES

Le panneau de ressources affiché par l'éditeur de programme contient une variété d'articles qui peuvent être glissés dans votre code. Les Tags de données et les catégories des programmes sont suffisamment explicites et donnent rapidement accès à des points correspondants dans votre base de données, permettant d'insérer le nom de l'élément dans votre éditeur. La catégorie système permet d'accéder à la bibliothèque des variables Crimson...



Comme vous pouvez le constater, les variables et les fonctions sont regroupées en catégories. Quand une fonction est sélectionnée, son type de retour et les types d'arguments sont indiqués sur la barre d'état. La suppression d'une fonction dans votre code entre le texte approprié, et place le curseur dans le texte entre parenthèses après le nom de fonction, permettant ainsi de saisir les arguments nécessaires.

## TYPES DE DONNEES PROGRAMME

Le champ situé au-dessus de l'éditeur de logiciel peut être utilisé pour modifier les types de données du programme...

	Type	Name
1:	Integer	Value1
2:	Integer	Value2
3:	None	Param3
4:	None	Param4
5:	None	Param5
6:	None	Param6

- La propriété *Type de données* est utilisé pour indiquer si ce programme devrait se borner à effectuer une série d'actions, ou si elle va effectuer un calcul et retourner la valeur de ce calcul à l'appelant. Programmes que les valeurs de retour ne peut par définition être gérées dans l'arrière-plan.
- La section des biens *Paramètres définissent* jusqu'à six paramètres que le programme va accepter. Chaque paramètre a un nom et un type de données. Dans cet exemple, le programme accepte deux paramètres, le `Value1` premier nommé et le `Value2` deuxième nommée, et tous deux étant signé 32 bits entiers.

Valeurs de retour et les paramètres de passage est examinée plus en détail ci-dessous.

## PROPRIETES PROGRAMME

Le deuxième onglet de l'éditeur définit l'exécution du programme environnement...

- La propriété *En Arrière-Plan* permet d'indiquer si Crimson doit attendre que le programme se termine avant de continuer le traitement de n'importe quelle tâche appelée qui a appelé le programme. Par exemple, si cette propriété est définie sur Non, si vous exécutez un programme après avoir appuyé sur une touche, une pause se produit dans les mises à jour de l'affichage jusqu'à ce que le programme se termine. (Comme la majorité des programmes mettent très peu de temps à s'exécuter, vous ne le noterez peut-être même pas.) Si cette propriété est définie sur Oui, les mises à jour de l'affichage continuent immédiatement et le programme s'exécute en arrière-plan, avec une priorité inférieure. Seul un programme en arrière-plan s'exécute à la fois. Par conséquent, les requêtes suivantes sont mises en file d'attente en vue d'une exécution ultérieure. Notez également que les programmes qui renvoient des valeurs ne peuvent pas être

exécutés en arrière-plan car leur valeur de retour ne serait alors pas disponible pour que l'appelant puisse l'utiliser !

- Les propriétés *Données externes* et *Timeout* permettent de contrôler l'interaction du programme avec l'infrastructure de communication de Crimson en ce qui concerne les éléments de données externes auxquels le programme fait référence. Rappelez-vous que Crimson lit uniquement les données lorsqu'elles sont utilisées. Cette propriété permet de contrôler l'interprétation exacte de cette règle en ce qui concerne les programmes...

<b>MODE</b>	<b>BEHAVIOR</b>
Lire si references	Les données externes utilisées par le programme sont ajoutées à l'analyse des communications dès que le programme est référencé. Si le programme est référencé par une page d'affichage, les données sont lues lorsque cette page s'affiche. Si le programme est référencé par une action globale ou un déclencheur, les données sont toujours lues. C'est là le mode par défaut, qui est acceptable pour tous les programmes sauf ceux qui utilisent de grandes quantités de données externes.
Toujours lire	Les données externes utilisées par le programme sont toujours lues, que le programme soit référencé ou non. Ainsi, le programme est toujours prêt à s'exécuter et l'opérateur ne voit pas le message « NOT READY » qui peut s'afficher lorsque le programme est référencé pour la première fois. L'inconvénient de ce mode réside dans le fait que les performances de communications peuvent être réduites si de grandes quantités de données sont référencées par le programme.
Lire si exécutées	Les données externes utilisées par le programme sont uniquement lues lorsque le programme est appelé. Le programme attend que la période définie dans la propriété <i>Timeout</i> de ces données soit disponible. Si les données ne peuvent être lues (peut-être parce qu'un périphérique est hors ligne), le programme ne s'exécute pas. Ce mode est généralement utilisé avec les programmes référencés de façon globale, qui consomment de grande quantités de données qui ralentiraient par ailleurs l'analyse des communications.
Lire mais executer tout de même	Les données externes sont gérées comme pour le mode <i>Toujours lire</i> , mais le programme s'exécute si les données ont été correctement lues ou non. L'opérateur ne voit donc jamais le message « NOT READY », mais si un périphérique est hors ligne, il n'existe aucune garantie que les éléments de données du programme contiennent des données valides.

## AJOUT DE COMMENTAIRES

Vous pouvez ajouter des commentaires de deux façons à vos programmes. Tout d'abord, vous pouvez utiliser la séquence `//` pour présenter un commentaire qui se poursuit dans le reste de la ligne en cours. Ensuite, vous pouvez utiliser la séquence `/*` pour présenter un commentaire d'une ou plusieurs lignes. Ce commentaire se poursuit jusqu'à ce que la séquence `*/` apparaisse. L'exemple ci-dessous montre les deux styles de commentaires...

```
// Commentaire sur une ligne

/* Ligne 1 du commentaire
   Ligne 2 du commentaire
   Ligne 3 du commentaire */
```

Vous pouvez également placer un commentaire sur une seule ligne à la fin d'une ligne contenant du code..

## RETOUR DE VALEURS

Comme nous venons de l'expliquer ci-dessus, les programmes peuvent retourner des valeurs. Ce type de programmes peuvent être appelés par d'autres programmes ou par des expressions, quel que soit l'endroit dans la base de données. Par exemple, si vous souhaitez effectuer un décodage complexe particulier sur plusieurs conditions relatives à un moteur, puis retourner une valeur afin d'indiquer l'état actuel, vous devez créer un programme qui renvoie un entier tel que celui-ci...

```
if( MotorRunning )
    return 1;
else {
    if( MotorTooHot )
        return 2;
    if( MotorTooCold )
        return 3;
    return 0;
}
```

Vous devez ensuite configurer une formule multi-états pour appeler ce programme, puis utiliser l'onglet Format de cette étiquette pour définir les noms des différents états. L'appel est effectué en définissant la propriété Valeur de l'étiquette sur `Programme()`, où `Programme` représente le nom du programme en question. Les parenthèses permettent d'indiquer un appel de fonction et ne doivent pas être oubliées.

### GARE !

Notez que vous devez être très attentif lorsque vous utilisez des programmes pour renvoyer des valeurs. En particulier, vous devez éviter de diffuser en boucle sur de trop longues périodes ou d'effectuer des actions illogiques dans le contexte dans lequel la fonction est appelée. Par exemple, si le fragment de code ci-dessus a appelé la fonction `GotoPage` pour modifier la page, l'affichage est modifié chaque fois que le programme est appelé. Imaginez ce qui pourrait vous arriver si vous avez par exemple essayé d'enregistrer des données à partir

de l'étiquette associée et que vous vous êtes rendu compte que cela n'est pas une bonne idée ! Voilà pourquoi vous devez conserver les programmes qui renvoient des valeurs simples et toujours prendre en compte le contexte dans lequel ils sont exécutés. En cas de doute, évitez de faire autre chose que des mathématiques simples et des instructions `if`.

## PASSAGE D'ARGUMENTS

Comme nous l'avons également expliqué ci-dessus, le programme peut accepter les arguments. Par exemple, supposez que vous souhaitez écrire un programme appelé `FindMean` pour prendre la moyenne des deux valeurs. Vous pouvez configurer le programme pour qu'il accepte deux arguments d'entiers, `a` et `b`, comme l'exemple fourni le montre lors de la définition de l'objectif de la propriété `Arguments`. Vous pouvez également configurer le programme pour qu'il renvoie une valeur entière. Le code du programme serait alors défini comme...

```
return (a+b)/2;
```

Une fois que vous avez créé et compilé ce programme, vous pouvez saisir une expression telle que `FindMean(Tag1, Tag2)` pour l'appeler avec les arguments appropriés. Dans ce cas, l'expression est égale à la moyenne de `Tag1` et `Tag2`.

## CONSEILS ET ASTUCES POUR LA PROGRAMMATION

Les sections ci-dessous fournissent une vue d'ensemble des constructions de programmation que Crimson prend en charge. La syntaxe de base utilisée est celle du langage de programmation C. Notez que l'objectif consiste à essayer de vous apprendre à devenir un programmeur ou à maîtriser les subtilités du langage C. Ces sujets dépassent le périmètre du présent manuel. Nous tenterons plutôt de vous fournir une rapide vue d'ensemble des fonctions disponibles afin que l'utilisateur intéressé puisse approfondir son expérience.

### ACTIONS MULTIPLES

Le type de programme le plus simple comprend une liste d'actions et chaque action occupe une ligne et est suivie d'un point-virgule. Toutes les différentes actions définies dans la section `Ecriture d'actions` sont disponibles pour être utilisées. Ces programmes simples sont en règle générale utilisés lorsque l'association des actions en une seule définition d'actions s'avère illisible.

L'exemple ci-dessous définit plusieurs variables, puis modifie la page d'affichage...

```
Motor1 = 0;  
Motor2 = 1;  
Motor3 = 0;  
  
GotoPage(Page1);
```

Les actions sont exécutées dans l'ordre et le programme revient vers l'appelant.

### INSTRUCTIONS IF

Ce type d'instruction est utilisé dans un programme afin de prendre une décision. La construction se compose d'une instruction `if` avec une condition entre parenthèses, suivie

d'une action (ou de plusieurs actions) qui est exécutée si la condition est vraie. Si plusieurs actions sont spécifiées, vous devez placer chacune d'entre elles sur une ligne distincte et utiliser des accolades pour regrouper les instructions. Vous pouvez utiliser une clause `else` facultative pour exécuter du code si la condition est fausse.

L'exemple ci-dessous montre une instruction `if` avec une action unique...

```
if( TankFull )
    StartPump = 1;
```

L'exemple ci-dessous montre une instruction `if` avec deux actions...

```
if( TankEmpty ) {
    StartPump = 0;
    OpenValue = 1;
}
```

L'exemple ci-dessous montre une instruction `if` avec une clause `else`...

```
if( MotorHot )
    StartFan = 1;
else
    StartFan = 0;
```

Notez qu'il est très important de ne pas oublier de placer les accolades autour des groupes d'actions à exécuter dans la partie `if` ou `else` de l'instruction. Si vous omettez les accolades, Crimson interprétera sans doute mal les actions exactes dont vous souhaitez dépendre lors de la condition `if`. Même si nous vous conseillons d'ajouter des sauts de ligne entre les actions, ils ne servent pas à comprendre ce qui est inclus ou pas dans l'instruction conditionnelle.

## INSTRUCTIONS SWITCH

Une instruction `switch` permet de comparer une valeur entière à un nombre de constants possibles et d'effectuer une action qui repose sur la valeur correspondante. La syntaxe exacte prend en charge plusieurs options au-delà de celles qui sont présentées dans l'exemple ci-dessous, mais pour la majorité des applications, cette forme simple est acceptable.

Cet exemple démarre un moteur sélectionné par la valeur dans l'étiquette `MotorIndex...`

```
switch( MotorIndex ) {  
  
    case 1:  
        MotorA = 1;  
        break;  
    case 2:  
    case 3:  
        MotorB = 1;  
        break;  
    case 4:  
        MotorC = 1;  
        break;  
    default:  
        MotorD = 1;  
        break;  
}
```

Une valeur de 1 démarre le moteur A, une valeur de 2 ou 3 démarre le moteur B et une valeur de 4 démarre le moteur C. Toute valeur qui n'est pas explicitement répertoriée démarre le moteur D. Concernant la syntaxe, vous devez noter les choses suivantes : l'utilisation des accolades autour des instructions `case`, l'utilisation de `break` pour conclure chaque bloc conditionnel, l'utilisation de deux instructions séquentielles `case` pour correspondre à plusieurs valeurs et l'utilisation de l'instruction facultative `default` pour indiquer une action à effectuer si la valeur ne correspond à aucune des valeurs spécifiées dans l'expression de contrôle. (Si cette syntaxe semble trop intimidante, vous pouvez utiliser une série d'instructions `if` à sa place pour obtenir les mêmes résultats, mais avec des performances et une lisibilité relativement inférieures.)

### VARIABLES LOCALES

Certains programmes utilisent des variables pour stocker les résultats intermédiaires ou pour contrôler l'une des différentes constructions de boucles décrites ci-dessous. Plutôt que de définir une étiquette pour qu'elle contienne ces valeurs, vous pouvez déclarer ce qu'on appelle des variables locales à l'aide de la syntaxe décrite ci-dessous...

```
int    a;           // Declare local integer 'a'  
float  b;           // Declare local real   'b'  
cstring c;         // Declare local string 'c'
```

Vous pouvez initialiser les variables locales lorsqu'elles sont déclarées en faisant suivre le nom de la variable de `:=` et de la valeur à attribuer. Les variables qui ne sont pas initialisées de cette façon sont définies sur zéro ou sur une chaîne vide, le cas échéant.

Notez que les variables locales sont réellement locales dans leur portée et leur durée de vie. C'est-à-dire qu'elles ne peuvent pas être référencées en dehors du programme et qu'elles ne conservent pas leur valeur entre les appels de fonctions. Si une fonction est appelée de façon récursive, chaque appel possède ses propres variables.

### CREATIONS DE BOUCLES

Vous pouvez utiliser les trois différentes constructions de boucles pour réaliser une section de code donnée alors qu'une certaine condition est vraie. La boucle `while` teste sa condition

avant l'exécution du code alors que la boucle `do` teste la condition après. La boucle `for` est un moyen plus rapide de définir une boucle `while` en vous permettant de combiner trois éléments communs en une instruction.

Notez que vous devez être attentive lorsque vous utilisez des boucles dans vos programmes car vous pouvez faire des erreurs de programmation et une boucle ne se terminerait ainsi jamais. Selon la situation dans laquelle le programme est appelé, cela peut sérieusement perturber l'activité de l'interface utilisateur du maître ou de ses communications. Les boucles qui se répètent de trop nombreuses fois peuvent également provoquer des problèmes de performances au sous-système qui les a appelés.

### LA BOUCLE WHILE

Ce type de boucle répète l'action qui la suit alors que la condition de l'instruction `while` reste vraie. Si la condition n'est jamais vraie, l'action n'est jamais exécutée et la boucle n'effectue aucune opération au-delà de l'évaluation de la condition de contrôle. Si vous souhaitez que la boucle comprenne plusieurs actions, assurez-vous que vous avez entouré les instructions d'accolades, comme avec l'instruction `if`. L'exemple ci-dessous initialise une paire de variables locales, puis utilise la première variable pour diffuser en boucle dans le contenu d'un tableau, totalisant les dix premiers éléments et renvoyant la valeur totale à l'appelant...

```
int i=0, t=0;

while( i < 10 ) {
    t = t + Data[i];
    i = i + 1;
}

return t;
```

L'exemple ci-dessous montre le même programme, mais réécrit dans une forme compressée. Comme l'instruction de la boucle contrôle maintenant une seule action, les accolades ont été omises...

```
int i=0, t=0;

while( i < 10 )
    t += Data[i++];

return t;
```

### LA BOUCLE FOR

Notez que la boucle `while` affichée ci-dessus dispose de quatre éléments...

1. L'initialisation de la variable de contrôle de la boucle.
2. L'évaluation d'un test pour vérifier si la boucle doit continuer.
3. L'exécution de l'action que la boucle doit effectuer.
4. La modification apportée à la variable du contrôle.

La boucle `for` permet d'associer les éléments 1, 2 et 4 dans une seule instruction de telle façon que l'action qui suit l'instruction doit seulement implémenter l'élément 3. Le résultat de cette syntaxe est assez semblable à celui de la boucle `FOR-NEXT` qui se trouve dans `BASIC` et d'autres langages similaires.

En utilisant cette instruction, vous pouvez réécrire l'exemple ci-dessous sous la forme...

```
int i, t;

for( i=t=0; i<10; i++ )
    t += Data[i];

return t;
```

Notez que l'instruction `for` contient trois éléments distincts, chacun étant séparé par un pointvirgule.

Le premier élément est l'étape d'initialisation qui est effectuée une fois que la boucle a commencé. Le deuxième élément est la condition qui est testée au début de chaque itération de boucle pour vérifier si la boucle doit continuer. Le dernier élément est l'étape d'induction qui permet d'apporter des modifications à la variable de contrôle afin de déplacer la boucle vers sa prochaine itération. Une nouvelle fois, n'oubliez pas que si vous souhaitez inclure plusieurs actions dans la boucle, ajoutez-les entre des accolades !

#### LA BOUCLE DO

Ce type de boucle est semblable à la boucle `while` sauf que la condition est testée à la fin de la boucle. C'est-à-dire que la boucle s'exécute toujours au moins une fois.

L'exemple ci-dessous montre l'exemple précédent, mais réécrit pour utiliser une boucle `do...`

```
int i=0, t=0;

do {
    t += Data[i];
} while( ++i < 10 );

return t;
```

#### CONTROLE DES BOUCLES

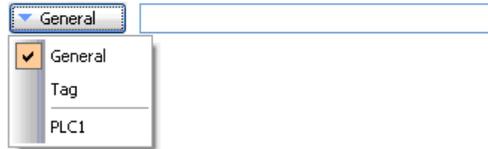
Vous pouvez utiliser deux autres instructions dans les boucles. L'instruction `break` permet de terminer la boucle par anticipation alors que l'instruction `continue` permet de reprendre au la boucle à son début sans atteindre la fin de celle-ci. Pour être logiques, ces instructions doivent être utilisées avec des instructions `if` pour que leur exécution soit conditionnelle.

L'exemple ci-dessous montre une boucle qui s'est terminée par anticipation lorsqu'un autre programme était vrai...

```
for( i=0; i<10; i++ ) {
    if( LoopAbort() )
        break;
    LoopBody();
}
```

# ECRITURE D'EXPRESSIONS

On sait depuis les premières sections de ce manuel que de nombreux domaines au sein de Crimson sont configurés comme ce qu'on appelle des propriétés d'expression. Vous se souviendrez également que ces champs sont édités au moyen d'un élément d'interface utilisateur similaire à celui indiqué ci-dessous...



Dans de nombreuses situations, vous savez quela configuration de ces biens est soit égale à la valeur d'une balise soit au contenu d'un registre dans un dispositif de communication à distance. Dans ces cas, il vous sera possible soit de glisser-déposer à partir de ressources le de panneau, soit de cliquer sur l'option appropriée sur le menu déroulant, puis de sélectionner l'élément de la boîte de dialogue qui en résulte.

Il y aura des situations plus problématiques, bien que, lorsque vous voulez créer une propriété dépendant d'une combinaison plus complexe d'éléments de données, les mathématiques vous aideront probablement. De telles éventualités sont traitées via ce qu'on appelle les expressions, qui peuvent être entrées dans la propriété de zone d'édition à chaque fois que le mode Général est sélectionné via le menu déroulant.

## VALEURS DE DONNEES

Toutes les expressions contiennent au moins une valeur de données. Les expressions les plus simples font donc référence à des constantes simples, balises simples ou simples registres PLC. Si vous entrez l'une des deux dernières options, Crimson permettra de simplifier le processus d'édition en changeant automatiquement le mode de propriété selon le cas. Par exemple, si vous entrez un nom de balise en mode général, Crimson bascule en mode Tag, et affiche le nom de la balise dans le champ de sélection.

### CONSTANTES

Les constantes représentent - ce qui n'est pas surprenant – des numéros constants ou des chaînes.

#### CONSTANTES ENTIER

Les constantes entières représentent un nombre simple 32 bits signé. Il peut être saisi en décimal, binaire, octal ou hexadécimal, au besoin. Les exemples ci-dessous montrent le même nombre inscrits dans les quatre bases nombre différent...

BASE	EXEMPLE
Décimale	123
Binaire	0b1111011
Octale	0173
Héxadécimale	0x7B

Les suffixes « U » et « L » supporté par les versions antérieures du logiciel ne sont pas utilisés.

#### CONSTANTES CARACTERE

Les constantes de caractères représentent un seul caractère Unicode, codé dans des nombres 16 bits inférieurs à 32 bits signés. Un caractère constant comporte un seul caractère entouré de guillemets simples, tels que 'A', et peut être utilisé pour représenter une valeur de 65. Certains autres caractères non imprimables ou irréprésentables peuvent être encodés en utilisant ce qu'on appelle des séquences d'échappement, dont chacun e est intriduite par un slash inversé...

SEQUENCE	VALEUR	ASCII
<code>\a</code>	Hex 0x07, Decimal 7	BEL
<code>\t</code>	Hex 0x09, Decimal 9	TAB
<code>\n</code>	Hex 0x0A, Decimal 10	LF
<code>\f</code>	Hex 0x0C, Decimal 12	FF
<code>\r</code>	Hex 0x0D, Decimal 13	CR
<code>\e</code>	Hex 0x1B, Decimal 27	ESC
<code>\xnn</code>	Valeur hex représentée par <i>nn</i> .	-
<code>\unnnn</code>	Valeur hex représentée par <i>nnnn</i> .	-
<code>\nnn</code>	Valeur octale représentée par <i>nnn</i> .	-
<code>\\</code>	Simple slash inversé	-
<code>\'</code>	Simple apostrophe	-
<code>\"</code>	Double apostrophe	-

#### CONSTANTES LOGIQUES

Les constantes logiques représentent une valeur 1 ou 0 qui est utilisée pour indiquer la vérité une expression sous forme oui-non. Un exemple de quelque chose qui peut être attribué à une logique constante est une balise qui représente une sortie numérique dans un automate. Les constantes logiques peuvent être comptabilisées comme de simples 1 ou 0, ou par l'utilisation des mots-clés `true` ou `false`.

#### CONSTANTES DE VIRGULE FLOTTANTE

Les constantes à virgules flottante représentent une valeur flottante de 32-bits. Elles sont représentées, comme on pouvait s'y attendre, par la partie entière suivie par un point décimal unique, et enfin par la partie fractionnaire. La notation scientifique est également soutenue en spécifiant une valeur pour la mantisse, suivie par un « E » et un exposant.

#### CONSTANTES DE CHAINES

Les constantes de chaînes représentent des séquences de caractères. Elles comprennent les caractères représentés, placés entre guillemets doubles. Par exemple, la chaîne "ABCD" représente une chaîne de quatre caractères, comprenant les valeurs 65, 66, 67 et 68. (En fait, cinq mots de 16 bits sont utilisés pour stocker la chaîne, avec une valeur nulle ajoutée comme terminateur.)

### VALEURS DE TAGS

La valeur d'un tag est représentée dans une expression par le nom de balise. Les tags qui sont organisés en dossiers sont représentés par un chemin de chaque paire d'éléments séparés par un point. Une balise nommée PV dans un dossier nommé boucle serait ainsi appelé `Loop.PV`. Notez que les majuscules et minuscules sont considérées comme équivalentes pour la fin de la balise requise. Une fois qu'une expression a été inscrite, toute modification apportée au nom de la balise va modifier toutes les expressions qui lui font référence.

### PROPRIETES DES TAGS

Les tags de données possèdent certaines propriétés qui peut être consulté en faisant suivre un nom de balise d'un point puis du nom de la propriété requise. Les propriétés suivantes sont définies...

PROPRIETE	DESCRIPTION	TYPE DE DONNEES
Name	Nom du tag	Chaîne
AsText	Valeur formatée du tag sous forme texte	Chaîne
Label	Propriétés de la légende du tag	Chaîne
Desc	Propriétés de description du tag	Chaîne
Prefix	Préfixe défini par le format du tag	Chaîne
Units	Unités définies par le format du tag	Chaîne
SP	Propriété du point de oparamétrage du tag	Similaire au tag
Min	Limite d'entré la plus basse du tag	Similaire au tag
Max	Limite d'entré la plus haute du tag	Similaire au tag
Fore	Couleur d'avant-plan du tag	Entier
Back	Couleur d'arriète-plan du tag	Entier

### PROPRIETES DES PAGES

Les pages d'affichage possèdent également certaines propriétés qui peuvent être consultées de la même manière...

PROPRIÉTÉ	DESCRIPTION	TYPE DE DONNÉES
Name	Nom de la page.	Chaîne.
Label	Page de propriété étiquette.	Chaîne.
Desc	Description des propriétés de la page.	Chaîne.

### REFERENCES COMM.

Les références aux registres situés dans les dispositifs de communications maîtres peuvent être entrées dans une expression au moyen d'une syntaxe comportant un crochet ouvrant, le nom de registre, et enfin un crochet fermant. Un nom de périphérique facultatif peut être précédé du nom du registre et séparé par un point. Le nom de périphérique n'est pas nécessaire pour désigner l'appareil dans une base de données.

Des exemples de cette syntaxe sont affichés ci-dessous...

EXEMPLE	SIGNIFICATION
[D100]	Registre D100 sur premier périphérique.
[AB.N7:0]	Registre N7:0 sur périphérique AB.
[FX.D100]	Registre D100 sur périphérique FX.

## UN PEU DE MATHS

Comme mentionné ci-dessus, les expressions contiennent souvent plus d'une valeur de données, ces dernières étant combinées mathématiquement. La plus simple de ces expressions peut ajouter une paire de valeurs, alors qu'une expression plus complexe pourrait obtenir la moyenne des trois valeurs. Ces opérations sont effectuées en utilisant la syntaxe que vous avez pu voir dans des applications comme Excel. Les exemples ci-dessous montrent les opérations de base qui peuvent être effectuées...

OPERATION	PRIORITE	EXEMPLE
Addition	Groupe 4	Tag1 + Tag2
Soustraction	Groupe 4	Tag1 - Tag2
Multiplication	Groupe 3	Tag1 * Tag2
Division	Groupe 3	Tag1 / Tag2
Remainder	Groupe 3	Tag1 % Tag2

Bien que les exemples soient présentés avec des espaces entourant les opérateurs, ces derniers ne sont pas nécessaires.

## PRIORITE OPERATEUR

Vous aurez remarqué la colonne Priorité dans le tableau ci-dessus. Comme vous vous en souvenez sans doute depuis vos cours d'algèbre, lorsque plusieurs opérateurs sont utilisés ensemble, ces derniers sont évalués dans un ordre défini. Par exemple, la multiplication est toujours évaluée avant l'addition. Crimson implémente cette commande par le biais de ce qu'on appelle les priorités de l'opérateur, chaque opérateur étant placé dans un groupe. Sauf indication contraire dans le texte, les opérateurs au sein d'un groupe sont évalués de gauche à droite. L'ordre par défaut de l'évaluation peut être outrepassé en utilisant des parenthèses.

## CONVERSION DE TYPE

Normalement, Crimson décidera automatiquement quand il faut passer de l'évaluation d'une expression en nombres entier à une évaluation en virgule flottante. Par exemple, si vous divisez un nombre entier par une valeur en virgule flottante, l'entier sera converti en virgule flottante avant que la division ne soit effectuée. Cependant, il y aura des situations où vous dans lesquelles vous souhaitez forcer une conversion.

Par exemple, supposons que vous additionnez trois nombres entiers qui représentent les niveaux de trois réservoirs, puis divisez total par le nombre de citernes pour obtenir le niveau moyen. Si vous utilisez une expression telle que  $(\text{Tank1} + \text{Tank2} + \text{Tank3}) / 3$  alors le résultat eut ne pas être aussi précis que souhaité, la division ayant lieu en utilisant des entiers et la moyenne ne contenant pas de décimales. Pour forcer Crimson à évaluer le résultat en

utilisant des virgules flottantes, la technique la plus simple consiste à changer les 3 en 3.0, ce qui contraint Crimson à convertir le montant en virgule flottante avant d'effectuer la division. Une technique un peu plus complexe consiste à utiliser la syntaxe telles que `float (Tank1 + Tank2 + Tank3) / 3.`

Les type casts peuvent également être utilisés pour convertir une valeur en virgule flottante en valeur entière, afin d'arrondir délibérément une valeur avant de la placer dans un registre PLC. Par exemple, l'expression `int (cos (Theta) * 100)` va calculer le cosinus d'un angle. Il faut multiplier cette valeur par 100 en utilisant les mathématiques à virgule flottante avant de la convertir en un entier, interrompant tout chiffre placé après la virgule.

### COMPARAISON DE VALEURS

Vous pourrez comparer aussi souvent que vous le souhaitez la valeur d'une des données avec celle d'une autre, et prendre une décision basée sur le résultat. Par exemple, vous pouvez souhaiter de définir un tag drapeau pour indiquer qu'un réservoir dépasse une valeur particulière, alors vous devrez utiliser une instruction `if` dans un programme d'exécution du code lorsque le moteur atteint sa vitesse désirée. Les opérateurs de comparaison suivants sont fournis...

OPERATEUR	PRIORITE	EXEMPLE
Egal à	Groupe 7	<b>Données == 100</b>
Pas égal à	Groupe 7	<b>Données != 100</b>
Supérieur à	Groupe 6	<b>Données &gt; 100</b>
Supérieur ou égal à	Groupe 6	<b>Données &gt;= 100</b>
Inférieur à	Groupe 6	<b>Données &lt; 100</b>
Inférieur ou égal à	Groupe 6	<b>Données &lt;= 100</b>

Chaque opérateur produit une valeur de 0 ou 1, selon l'état du test. Les opérateurs peuvent être utilisés sur des entiers, des valeurs flottantes ou des chaînes. Si les chaînes sont comparées, la comparaison est insensible à la casse : ainsi "abc" est considéré comme égal à "ABC".

### TEST DES BITS

Crimson vous permet de tester la valeur d'un bit dans une valeur de données en utilisant l'opérateur de sélection bits, ce qui est représenté par un point unique. Le côté gauche de l'opérateur devrait être la valeur dans lequel le bit est à tester, et le côté droit doit être une expression qui indique le nombre de bits à tester. Cette valeur devrait être comprise entre 0 et 31. Le résultat de l'opérateur est égal à 0 ou 1 selon la valeur du bit en question.

OPERATEUR	PRIORITE	EXEMPLE
Sélection de bit	Groupe 1	<b>Input . 2</b>

L'exemple ci-dessus présente un bit 2 essais (à savoir le bit d'une valeur de 4) dans la balise indiquée.

Si vous souhaitez tester un peu étant égal à zéro, vous pouvez utiliser l'opérateur logique NOT...

OPERATEUR	PRIORITE	EXEMPLE
Logique NOT	Groupe 2	<b>! Input . 2</b>

Cet exemple est égal à 1 si le bit 2 de la balise indiquée est égale à 0, et vice versa.

## CONDITIONS MULTIPLES

Si vous souhaitez définir une expression qui est vraie si un certain nombre de conditions *sont toutes* remplies, vous pouvez utiliser l'opérateur logique AND. De même, si vous souhaitez définir une expression qui est vraie si *n'importe laquelle* d'un certain nombre de conditions est remplie, vous pouvez utiliser l'opérateur logique OR. Les exemples ci-dessous montrent chaque opérateur...

OPERATEUR	PRIORITE	EXEMPLE
Logique AND	Groupe 11	<b>A&gt;10 &amp;&amp; B&gt;10</b>
Logique OR	Groupe 12	<b>A&gt;10    B&gt;10</b>

L'opérateur logique ET produit une valeur de 1 si et seulement si les expressions sur la gauche et côté droit sont vraies, alors que l'opérateur OU logique produit une valeur de 1 si une seule expression est vraie. Notez que contrairement aux opérateurs de bits visés ailleurs dans le présent article, les opérateurs logiques arrêtent l'évaluation une fois qu'ils connaissent la réponse. Cela signifie que dans l'exemple ci-dessus pour ET logique, le côté droit de l'exploitant ne sera évalué que si A est supérieur à 10.

## CHOIX DES VALEURS

Vous pourrez trouver des situations où vous voudrez choisir entre deux valeurs, qu'il s'agisse d'entiers, de nombres flottants ou de chaînes de valeurs de point-dépendant de la valeur d'une condition. Par exemple, vous souhaitez mettre en place une vitesse égale à 500 tr / min ou 2000 tr / min pour un moteur basé sur un tag drapeau. Cette opération peut être effectuée en utilisant l'opérateur ? : qui est unique en ce qu'il prend trois arguments, comme le montre l'exemple ci-dessous...

OPERATEUR	PRIORITE	EXEMPLE
Sélection	Groupe 13	<b>Fast ? 2000 : 500</b>

Cet exemple sera évalué à 2000 si *rapide* est considéré comme Vrai, 500 autrement. L'opérateur peut être considérée comme équivalente à la fonction SI dans les applications telles que Microsoft Excel.

## MANIPULATION DES BITS

Crimson offre aussi aux opérateurs la possibilité d'effectuer des opérations qui ne traitent pas les entiers comme des valeurs numériques, mais plutôt comme des séquences de bits. Ces opérateurs sont appelés opérateurs de bits.

### AND, OR ET XOR

Ces trois opérateurs de bits produisent chacun un résultat dans lequel chaque bit est défini comme étant égal aux bits correspondants dans les valeurs sur l'opérateur à gauche et côté droit...

OPERATEUR	PRIORITE	EXEMPLE
Bitwise AND	Groupe 8	<code>Data &amp; Mask</code>
Bitwise OR	Groupe 9	<code>Data   Mask</code>
Bitwise XOR	Groupe 10	<code>Data ^ Mask</code>

Le tableau ci-dessous montre les tables de vérité associées...

A	B	A & B	A   B	A ^ B
0	0	0	0	0
0	1	0	1	1
1	0	0	1	1
1	1	1	1	0

### BASCULE OPERATEURS

Crimson fournit également aux opérateurs la possibilité de transférer un nombre entier de *n* bits vers la gauche ou la droite...

OPERATEUR	PRIORITE	EXEMPLE
Shift Gauche	Groupe 5	<code>Data &lt;&lt; 2</code>
Shift Droite	Groupe 5	<code>Data &gt;&gt; 2</code>

Chaque exemple déplace de deux bits la fonction `Données` dans la direction indiquée.

### COMMANDE NOT

Enfin, Crimson fournit un opérateur Commande NOT permettant d'inverser le sens des bits d'une valeur...

OPERATEUR	PRIORITE	EXEMPLE
Commande NOT	Groupe 2	<code>~Mask</code>

Cet exemple génère une valeur où chaque bit est égal à l'inverse de sa valeur en `Mask`.

### INDEXATION FLECHES

Les éléments situés à l'intérieur d'une balise tableau peuvent être sélectionnés à la suite du nom du tableau avec des crochets qui contiennent une expression d'indexation. Cette expression doit être comprise entre 0 et un, moins le nombre d'éléments dans le tableau. Si vous créez un tableau à 10 éléments, par exemple, le premier élément aura le nom `Nom [0]` et le dernier le `Nom [9]`.

## INDEXATION CHAINES

Les crochets peuvent également être utilisés pour sélectionner les caractères d'une chaîne. Par exemple, si vous avez une balise appelée `texte` qui contient la chaîne "ABCD", l'expression de `Texte [0]` renverra une valeur de 65, ce qui est égal à la valeur Unicode du premier caractère. Les valeurs d'indice au-delà de la fin de la chaîne devront toujours renvoyer zéro.

## AJOUT DE CHAINES

Ainsi que l'ajout de numéros, l'opérateur d'addition peut être utilisé pour concaténer des chaînes. Ainsi, l'expression "`AB`" + "`CD`" est équivalente à "ABCD". Vous pouvez également utiliser l'opérateur d'addition pour ajouter un entier en chaîne, auquel cas un caractère unique égal à la représentation Unicode de l'entier est ajoutée aux données dans la chaîne.

## APPEL PROGRAMMES

Les programmes retournant des valeurs peuvent être appelés via des expressions suivant le nom du programme avec une paire de parenthèses. Par exemple, `Program1() * 10` appellera le programme associé, et multipliera la valeur de retour de 10. Évidemment, le type de retour pour `Program1` doit être défini en entier ou en virgule flottante pour cela ait un sens.

## UTILISATION DES FONCTIONS

Crimson fournit un certain nombre de fonctions prédéfinies qui peuvent être utilisées pour accéder aux informations du système, ou pour effectuer des opérations mathématiques courantes. Ces fonctions sont définies en détail dans la référence de la fonction. Elles sont invoquées en utilisant une syntaxe similaire à celle des programmes, avec des arguments de fonction entourés de parenthèses. Par exemple, `cos(0)` appelle la fonction cosinus avec un argument de 0, retournant une valeur de +1,0.

## RESUME PRIORITES

Le tableau ci-dessous montre la priorité de tous les opérateurs définis dans la présente section...

GROUPE	OPERATEURS
Groupe 1	.
Groupe 2	! ~
Groupe 3	* / %
Groupe 4	+ -
Groupe 5	<< >>
Groupe 6	< > <= >=
Groupe 7	== !=
Groupe 8	&
Groupe 9	
Groupe 10	^
Groupe 11	&&

<b>GROUPE</b>	<b>OPÉRATEURS</b>
Groupe 12	
Groupe 13	? :

Les opérateurs dont les numéros sont les moins élevés sont appliqués en premier.



# ECRITURE DES ACTIONS

Alors que des expressions définissent des valeurs, les actions vous permettent de définir ce que vous souhaitez qu'il se produise lorsqu'un événement apparaît. La grande majorité des actions situées dans une base de données seront liées à des interactions avec des primitives ou avec le clavier. Etant donné que Crimson fournit une méthode simple de définir les actions couramment utilisés pour ces articles, vous serez souvent en mesure d'éviter d'écrire les actions à la main. Des actions sont nécessaires, mais vous pouvez utiliser des déclencheurs, écrire des programmes, ou utiliser une clé ou une primitive en mode utilisateur défini.

## CHANGER DE PAGE

Pour créer une action qui modifie la page affichée sur l'écran du panneau, utilisez la syntaxe `GoToPage (Nom)`, où `Nom` est le nom de l'affichage de la page en question. La page en cours sera supprimée, et la nouvelle page sera affichée à sa place.

## CHANGER LES VALEURS NUMERIQUES

Crimson propose plusieurs façons de changer les valeurs des données.

### ASSIGNEMENT SIMPLE

Pour créer une action qui affecte une nouvelle valeur à un tag ou à un registre dans un dispositif de communication, utilisez la syntaxe `Données = Valeur`, où les `Données` sont l'élément devant être modifié, et la `valeur` la valeur à attribuer. Notez que la `Valeur` peut ne pas être une constante, mais n'importe quelle expression valide de type correct. Reportez-vous à la section précédente pour plus de détails sur la façon d'écrire des expressions. Par exemple, un code tel que `[N7: 0] = Tank1 + Tank2` peut être utilisé pour ajouter deux niveaux de réservoirs et de stocker la quantité totale directement dans un registre PLC.

### ASSIGNEMENT ELEMENT

Pour créer une action qui fixe une valeur de données égale à sa valeur actuelle, combinée avec une autre valeur par le biais de l'un des opérateurs définis dans la section précédente, utilisez la syntaxe `Données op = Valeur`, Lorsque `Données` sont le tag à modifier, `Valeur` est la valeur à être utilisée par l'opérateur, et `op` est l'un des opérateurs disponibles. Par exemple, le code `Tag + = 10` augmentera le `Tag` de 10, tandis que `Tag* = 10` le multipliera par 10.

### INCREMENTATION ET DECREMENTATION

Pour créer une action qui augmente la valeur de données de un, utilisez la syntaxe `Données + +`. Pour créer une action qui diminue un tag de un, utilisez la syntaxe `Données - -`. Notez que les opérateurs `+ +` ou `- -` peuvent être placés avant ou après la valeur des données en question. Dans le premier cas, la valeur de l'expression représentée par `des + + Données` est égale à la valeur des `Données` après incrémentation. Dans ce dernier cas, l'expression est égale à la valeur *avant* changement.

## MODIFICATION DE VALEURS DE BITS

Pour changer un bit au sein d'un tag, utilisez la syntaxe `Data.Bit = 1` ou `Data.Bit = 0` pour définir ou effacer au besoin le bit, lorsque les `Données` représentent est la base zéro du nombre de bits. Notez encore que la valeur située sur le côté droit de l'opérateur `=` peut être une expression si désiré, de telle sorte que dans un exemple comme `Données1 = (Niveau > 10)` il est possible d'utiliser cet opérateur pour définir ou effacer un peu selon que le niveau de réserve dépasse ou non la valeur prédéfinie.

## UTILISATION DES PROGRAMMES

Les programmes peuvent être invoqués dans des actions en faisant suivre le nom du programme d'une paire de parenthèses. Par exemple, `Program1()` va appeler le programme associé. Le programme s'exécute dans l'avant-plan ou l'arrière-plan tel que défini par les propriétés du programme.

## UTILISATION DES FONCTIONS

Crimson fournit un certain nombre de fonctions prédéfinies qui peuvent être utilisées pour effectuer différentes opérations. Ces fonctions sont définies en détail dans la référence de la fonction. Elles sont invoquées en utilisant une syntaxe similaire à celle des programmes, les arguments de la fonction étant entourés de parenthèses. Par exemple, `SetLanguage(1)` va définir la langue d'administration sur 1.

## PRIORITE OPERATEUR

Tous les opérateurs d'affectation tombent dans le groupe 14. En d'autres termes, ils seront évalués après tous les autres opérateurs dans une action. Ils sont également uniques en ce qu'ils se groupent de gauche à droite. Cela signifie qu'un code tel que `Tag1 = Tag2 = Tag3 = 0` peut être utilisé pour effacer toutes les trois balises à la fois.