



## **LoriotPro V5.00 « Extended Edition » System Expert et Scripts**

### **Introduction**

### **TOME 1**

**V1.00**  
**Par Ludovic Lecointe**



**Copyright © 2005-2008 LUTEUS SARL. All rights reserved. This documentation is copyrighted by LUTEUS SARL. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying or otherwise, without the prior express written permission of LUTEUS SARL**



## TABLE DES MATIERES

<b>1</b>	<b>Pré requis .....</b>	<b>11</b>
<b>1.1</b>	<b>Conventions de lecture .....</b>	<b>12</b>
1.1.1.1	Remarque .....	12
1.1.1.2	Portion de code .....	12
1.1.1.3	Commentaire dans le code .....	12
1.1.1.4	Signe d'attention.....	13
1.1.1.5	Type des variables et syntaxe.....	13
1.1.1.6	Variable 'ip'.....	13
1.1.1.7	Variable 'oid'.....	13
1.1.1.8	Variable 'uid'.....	14
1.1.1.9	Variable 'path'.....	14
1.1.1.10	Variable 'array' .....	14
<b>2</b>	<b>Introduction.....</b>	<b>15</b>
<b>3</b>	<b>Introduction au modèle SNMP .....</b>	<b>16</b>
<b>3.1</b>	<b>Limitation du protocole SNMP .....</b>	<b>16</b>
<b>3.2</b>	<b>Modules dédiés de LoriotPro .....</b>	<b>17</b>
3.2.1	Les plugins attachés aux équipements.....	17
3.2.2	Les plugins de service .....	18
3.2.3	Les plugins directs.....	19
<b>3.3</b>	<b>Fonction classique d'un NMS.....</b>	<b>20</b>
<b>3.4</b>	<b>Limitation d'un NMS classique.....</b>	<b>21</b>
<b>3.5</b>	<b>Exploitation d'un infrastructure communicante .....</b>	<b>22</b>
<b>3.6</b>	<b>Constat concernant le management d'une infrastructure communicante</b>	<b>23</b>
<b>3.7</b>	<b>Nécessité de l'intégration d'une couche programmable .....</b>	<b>23</b>
3.7.1	Calcul de pourcentage d'utilisation d'un disque dur .....	23
3.7.1.1	Phase 1 : recherche documentaire.....	24
3.7.1.2	Phase 2 : trouver l'index du disque C .....	26
3.7.1.3	Phase 3 : Collecter les valeurs .....	28
3.7.1.4	Phase 4 : faire le calcul .....	28
3.7.1.5	Phase 5 : afficher le résultat.....	30
3.7.2	Conclusion.....	30
<b>4</b>	<b>Traitemet efficace de l'information .....</b>	<b>33</b>
<b>4.1</b>	<b>La solution LoriotPro V5 .....</b>	<b>33</b>
4.1.1	Spectre d'expertise de LoriotPro V5 .....	34
<b>4.2</b>	<b>Le Système Expert de LoriotPro V5.....</b>	<b>34</b>
4.2.1	Composant principaux du Système Expert.....	35
<b>4.3</b>	<b>Présentation des modules formant le système Expert .....</b>	<b>36</b>
4.3.1	Interpréteur LUA.....	37
4.3.1.1	Plus d'information sur LUA.....	37
4.3.2	L'environnement de développement intégré.....	38
4.3.3	Le module IP Expert .....	38
4.3.4	Le module d'audit multi-tâche.....	40
4.3.5	Objets SNMP Virtuel.....	44
4.3.6	Les ActiveView .....	45
4.3.7	Graphe et courbe.....	48
4.3.8	Le module de configuration multiple « bulk configuration » .....	49
4.3.9	Le Report Center.....	50

---

4.3.10	Le Module IP Scanner .....	51
4.3.11	Les Filtres d'Events et de Traps scriptés .....	53
<b>5</b>	<b>Le Langage LUA .....</b>	<b>55</b>
<b>5.1</b>	<b>Intégration de LUA dans LoriotPro .....</b>	<b>55</b>
5.1.1	Variables globales .....	56
5.1.1.1	Variables en entrée du script.....	56
5.1.1.2	Variables en sortie du script .....	56
5.1.2	Exemple .....	57
<b>6</b>	<b>Librairies LUA de LoriotPro .....</b>	<b>59</b>
<b>6.1</b>	<b>Introduction.....</b>	<b>59</b>
<b>6.2</b>	<b>Restriction d'usage des entiers 64bits .....</b>	<b>59</b>
<b>6.3</b>	<b>Fichier de définition de variables système.....</b>	<b>63</b>
<b>6.4</b>	<b>LoriotPro Kernel Librairie (lp).....</b>	<b>67</b>
<b>6.5</b>	<b>LoriotPro ActiveView Librairie (lpav) .....</b>	<b>70</b>
<b>6.6</b>	<b>LoriotPro Wizard Librairie (lpw) .....</b>	<b>71</b>
<b>6.7</b>	<b>LoriotPro SLA (lpsla) Librairie.....</b>	<b>73</b>
<b>7</b>	<b>Exemple de code sous forme de fonction LUA.....</b>	<b>76</b>
7.1.1	Gestion de périodes .....	76
7.1.1.1	function LP_GetAbsolutePeriodeMin.....	76
7.1.1.2	function LP_GetAbsolutePeriodeHour.....	76
7.1.1.3	function LP_GetAbsolutePeriodeDay .....	77
7.1.1.4	function LP_GetPeriodeMinute.....	77
7.1.1.5	function LP_GetPeriodeHour.....	77
7.1.1.6	function LP_ComputeUpTime .....	77
7.1.2	Analyse de ressource SNMP .....	78
7.1.2.1	function LP_AuditStorage.....	78
7.1.2.2	function LP_AuditSystem .....	79
7.1.2.3	function LP_AuditSoftwareInstalled.....	79
7.1.2.4	function LP_AuditDriverInstalled .....	80
7.1.2.5	function LP_AuditRouteTable.....	80
7.1.2.6	function LP_AuditDot3Stat .....	81
7.1.2.7	function LP_AuditCiscoCPU.....	81
<b>8</b>	<b>Environnement de développement intégré .....</b>	<b>83</b>
<b>8.1</b>	<b>Introduction.....</b>	<b>83</b>
<b>8.2</b>	<b>Lancement de l'éditeur .....</b>	<b>83</b>
<b>8.3</b>	<b>Zone d'édition du script .....</b>	<b>84</b>
8.3.1	Coloration .....	85
8.3.2	Complétion .....	86
8.3.3	Tooltips .....	86
8.3.4	Numerotation des lignes .....	87
8.3.5	Gestion des encapsulations de code.....	87
8.3.6	Sélection de texte .....	88
8.3.7	Taille des caractères .....	88
8.3.7.1	Options - Copier / coller 'undo redo' .....	89
8.3.8	Search Replace .....	90
8.3.9	Mode multi-pages .....	91
8.3.10	Menu files .....	92
8.3.10.1	Options - New.....	92
8.3.10.2	Options - New Wizard .....	93
8.3.10.3	Options - Open .....	93

---

8.3.10.4	Options - Open Wizard .....	94
8.3.10.5	Options - Open IP Expert/Scanner File .....	95
8.3.10.6	Options - Open Audit File .....	95
8.3.10.7	Options - Open Report File .....	96
8.3.10.8	Options - Save .....	97
8.3.10.9	Options - Save As .....	97
8.3.10.10	Options - Print .....	98
8.3.10.11	Options - Page Setup .....	99
8.3.11	Menu Edit .....	100
8.3.11.1	Options - Clear Result Box .....	101
8.3.11.2	Options – Make Selection Uppercase .....	102
8.3.11.3	Options – Make Selection Lowercase .....	102
8.3.11.4	Options – Block Comment or Uncomment .....	102
8.3.12	Menu View .....	103
8.3.12.1	Options - ActiveView Box .....	103
8.3.12.2	Options - Trace Output Box .....	104
8.3.12.3	Options – White Space .....	105
8.3.12.4	Options – End Of File .....	105
8.3.12.5	Options – Indentation Guides .....	105
8.3.13	Menu Insert .....	105
8.3.13.1	Options - Insert SNMP OID .....	106
8.3.13.2	Options - Insert IP Host .....	109
8.3.13.3	Options - Insert Directory UID Object .....	109
8.3.13.4	Options - Insert Event Filter UID .....	110
8.3.13.5	Options - Insert Trap Filter UID .....	111
8.3.13.6	Options - Insert Audit Reference Number .....	111
8.3.13.7	Options - Insert IPExpert Reference Number .....	112
8.3.13.8	Options - Insert Event Reference Number .....	113
8.3.13.9	Options - Insert Color Reference .....	114
8.3.13.10	Options - Insert Path/file .....	115
8.3.13.11	Options - Menu BreakPoint .....	115
8.3.13.12	Options - Add a BreakPoint .....	116
8.3.13.13	Options - Delete BreakPoint .....	117
8.3.13.14	Options - Delete All BreakPoint .....	117
8.3.13.15	Options - Find Next BreakPoint .....	117
8.3.13.16	Options - Find Previous BreakPoint .....	117
8.3.14	Menu Search .....	117
8.3.14.1	Options - Find .....	117
8.3.14.2	Options - Find Next .....	118
8.3.14.3	Options - Replace .....	118
8.3.15	Menu Options .....	118
8.3.15.1	Options - Display Line Numbers .....	119
8.3.15.2	Options - Display Selection/Bookmark Margin .....	119
8.3.15.3	Options - Display folding Margin .....	119
8.3.15.4	Options - Set Global LUA Debugging .....	120
8.3.15.5	Options – Line End Characters .....	121
8.3.15.6	Options – Convert Line End Characters .....	121
8.3.16	Menu Compiler .....	122
8.3.16.1	Options - Run Script .....	123
8.3.16.2	Options - Stop Script .....	124

---

8.3.16.3	Options - Kill Running Script .....	124
8.3.16.4	Options - Trace call in code.....	124
8.3.16.5	Options - Run Report .....	125
8.3.17	Menu Help .....	127
8.3.17.1	Options - LUA Reference manual.....	127
8.3.17.2	Options - LUA Scripting documentation .....	128
<b>8.4</b>	<b>Zone d'édition des variables .....</b>	<b>129</b>
8.4.1	Menu Value .....	132
8.4.1.1	Options - Insert Value.....	132
8.4.1.2	Options - Clear All Value .....	133
<b>8.5</b>	<b>Fenêtre des messages systèmes .....</b>	<b>134</b>
<b>8.6</b>	<b>Fenêtre d'affichage des graphiques (ActiveView) .....</b>	<b>134</b>
<b>9</b>	<b>Le module IP Expert .....</b>	<b>136</b>
<b>9.1</b>	<b>Introduction.....</b>	<b>136</b>
<b>9.2</b>	<b>Logique de fonctionnement du module IP Expert .....</b>	<b>136</b>
<b>9.3</b>	<b>Découverte de topologie réseau et des ressources .....</b>	<b>136</b>
9.3.1	Introduction.....	136
9.3.2	Lancement du Wizard de découverte .....	137
9.3.3	Collecte local de donnée .....	139
9.3.4	Lancement des modules de découvertes .....	140
9.3.4.1	Module Discover.....	140
9.3.4.2	Module IP Expert.....	142
9.3.4.2.1	Options - Run IP Expert on Directory Networks .....	144
9.3.4.2.2	Options - Run IP Expert on Directory Hosts Object.....	144
9.3.4.2.3	Options - Assign Host parameters from IP Expert Informations	144
9.3.4.3	Paramètres du module IP Expert .....	144
9.3.4.4	Fenêtre - Scan Option .....	145
9.3.4.5	Options - Scan TCP .....	146
9.3.4.5.1	Script IP Expert et stratégie associées.....	146
9.3.4.6	Stratégie IP Expert associé au réseaux .....	147
9.3.4.6.1	Options - Min Scan Interval.....	149
9.3.4.6.2	Options - Event Generated .....	150
9.3.4.7	Stratégie IP Expert associé aux équipements.....	151
9.3.4.8	cycle de découverte .....	153
<b>9.4</b>	<b>Génération de Scripts pour le module IP Expert ou IP Scanner .....</b>	<b>154</b>
9.4.1	Principe de lancement des scripts IP Expert .....	157
9.4.2	Structure des fichiers scanmod.lua et scan.lua .....	158
9.4.3	Structure d'un fichier de script IP Expert .....	161
9.4.3.1	Entête d'un fichier ipscanner-x.lua .....	161
9.4.4	Wizard de création d'un fichier de script IP Expert .....	164
9.4.4.1	Options - Reference Number for your script.....	166
9.4.4.2	Options - Title for this Expert/Scanner Script .....	166
9.4.4.3	Options - Comment associer à votre script.....	166
9.4.4.4	Options - Number of function desired.....	166
9.4.4.5	Options - TCP port list used by this script .....	167
9.4.4.6	Options - Line parameters .....	167
9.4.4.7	Exemple de script.....	167
9.4.4.7.1	zone d'exécution du script.....	171
9.4.4.7.2	Zone des fonctions du script .....	171
9.4.4.8	Sauvegarde du script IP Expert .....	174

---

9.4.4.9	Visualisation du nouveau script dans les stratégies .....	175
9.4.4.9.1	Test du script avec IP Scanner .....	178
<b>10</b>	<b>Le module d'Audit.....</b>	<b>183</b>
<b>10.1</b>	<b>Introduction.....</b>	<b>183</b>
10.1.1	Quelques exemples.....	183
<b>10.2</b>	<b>Console du module d'audit.....</b>	<b>184</b>
<b>10.3</b>	<b>Insertion manuelle d'un script d'audit.....</b>	<b>185</b>
10.3.1	Boîte de dialogue de sélection du type d'audit .....	187
10.3.1.1	Options - Polling Interval .....	188
10.3.1.2	Options - Event Sent .....	189
10.3.1.3	Options - At level .....	189
10.3.1.4	Options - If status .....	190
10.3.1.5	Options - SLA collect.....	190
10.3.1.6	Options - Enable process .....	191
10.3.1.7	Options - polling Type (TCP Connect).....	191
10.3.1.8	Options - Polling Port .....	192
10.3.1.8.1.1	Exemple.....	192
10.3.1.9	Options - Polling Type (Audit process).....	193
10.3.1.10	Options - Audit Ref .....	194
10.3.1.11	Options - Parameters.....	195
10.3.1.12	Options - Generate Audio .....	196
10.3.1.13	Options - Generate Reporting Options .....	196
10.3.1.14	Options - Generate Graph Options .....	196
10.3.2	Exemples.....	196
<b>10.4</b>	<b>Insertion automatique via un script .....</b>	<b>203</b>
<b>10.5</b>	<b>Génération de Scripts pour le module d'Audit .....</b>	<b>205</b>
10.5.1	Principe de lancement des scripts d'audit .....	206
10.5.2	Structure d'un fichier de scripts d'audits .....	209
10.5.2.1	Entête d'un fichier ipaudit.lua .....	209
10.5.3	Wizard de création d'un fichier de scripts d'audits.....	210
10.5.3.1	Options - Reference Number for your audit.....	212
10.5.3.2	Options - Title for this Audit Script .....	212
10.5.3.3	Options - Comment associe to your script.....	212
10.5.3.4	Options - Line parameters .....	212
10.5.3.5	Options - This audit generate data for graph .....	212
10.5.3.6	Exemple de script.....	212
<b>11</b>	<b>Objet SNMP virtuel.....</b>	<b>224</b>
11.1.1	Passage de paramètres .....	224
11.1.2	Objet snmp virtuel par défaut .....	233
11.1.3	Exception d'utilisation des objets SNMP virtuels .....	238
<b>12</b>	<b>ActiveView .....</b>	<b>241</b>
<b>12.1</b>	<b>Introduction.....</b>	<b>241</b>
<b>12.2</b>	<b>Principe d'utilisation des scripts .....</b>	<b>241</b>
<b>12.3</b>	<b>Types de scripts utilisables dans les ActiveViews.....</b>	<b>241</b>
<b>12.4</b>	<b>Usage direct d'objets SNMP virtuels .....</b>	<b>242</b>
<b>12.5</b>	<b>Calculette scientifique.....</b>	<b>243</b>
<b>12.6</b>	<b>Lancement scheduler de script LUA.....</b>	<b>249</b>
<b>12.7</b>	<b>Script utilisé avec la fonction DoubleClick .....</b>	<b>251</b>
<b>12.8</b>	<b>Script associé au menu contextuel d'un objet graphique .....</b>	<b>252</b>
<b>12.9</b>	<b>Utilisation directe de scripts dans une ActiveView .....</b>	<b>256</b>

---

12.10 Utilisation de Script d'initialisation pour les ActiveView Modal .....	259
12.11 ShortCut associant une ActiveView Modal et un Script LUA.....	261
<b>13 Graphe et Courbe.....</b>	<b>272</b>
13.1 Introduction.....	272
13.2 Introduction à RRD .....	273
13.3 Implémentation de RRD dans LoriotPro.....	277
13.4 Licence RRD .....	278
<b>13.5 Introduction à RRD Collector .....</b>	<b>278</b>
13.5.1 Principe d'exécution des scripts .....	280
13.5.2 Exemple de script – Disk Usage.....	280
13.5.3 Lancement de RRD Collector .....	281
13.5.3.1 Insertion du plugin RRD Collector .....	281
13.5.3.2 Configuration de la partie Script du plugin.....	283
13.5.3.2.1 Option – Insert.....	283
13.5.3.2.1.1 Option – Script Name .....	284
13.5.3.2.1.2 Option – LUA Script File .....	284
13.5.3.2.1.3 Option – Script description.....	285
13.5.3.2.2 Zone – Script Input arguments .....	285
13.5.3.2.2.1 Option – Host Name .....	285
13.5.3.2.2.2 Option – IP Address .....	286
13.5.3.2.2.3 Option – Arg1 (SNMP oid).....	286
13.5.3.2.2.4 Option – Arg2 (SNMP index) .....	286
13.5.3.2.3 Zone – Script test.....	286
13.5.3.2.3.1 Zone – Type .....	287
13.5.3.2.4 Option – Delete .....	287
13.5.3.2.5 Option – Move up.....	288
13.5.3.2.6 Option – Move down .....	288
13.5.3.2.7 Option – Clear all .....	288
13.5.3.2.8 Option – Graph style .....	288
13.5.3.2.9 Option – Y axis legend .....	288
13.5.3.2.10 Option – Polling Interval .....	288
13.5.3.2.11 Option – Initialize DataBase .....	288
13.5.3.2.12 Option – Enable Logging.....	289
13.5.3.2.13 Option – View Log .....	289
13.5.3.2.14 Option – Advanced options .....	290
13.5.3.3 Lancement du plugin .....	290
13.5.3.4 Visualisation des graphes .....	291
13.5.3.5 Visualisation des graphes avec les fenêtres volantes .....	293
<b>14 Le module de configuration multiple – Bulk configuration .....</b>	<b>296</b>
14.1 Introduction.....	296
14.2 Sélection multiple d'équipement.....	296
14.2.1.1 Création de nouveau script de type « bulk » .....	300
14.2.1.1.1 Exemple d'un script de type bulk.....	301
<b>15 Report Center .....</b>	<b>303</b>
15.1 Introduction.....	303
15.2 Configuration .....	304
15.2.1 Création d'une branche .....	304
15.2.2 Exploitation du script d'exemple .....	308
<b>16 Le Module IP Scanner .....</b>	<b>311</b>
16.1 Introduction.....	311

---

<b>16.2 Lancement d'IP Scanner associé à un équipement .....</b>	<b>311</b>
<b>16.3 Lancement d'IP Scanner associé à un réseau .....</b>	<b>312</b>
16.3.1 Configuration du module IP Scanner.....	313
16.3.2 La zone de configuration .....	314
16.3.2.1 Options - IP Start Address.....	314
16.3.2.2 Options - IP Stop Address.....	314
16.3.2.3 Options - Scan All Host in Directory .....	314
16.3.2.4 Options - Clear Tree.....	314
16.3.2.5 Options - Clear List Box .....	314
16.3.3 Section - Add / Update Hosts Directory.....	314
16.3.3.1 Options - AddUpdate Entry .....	314
16.3.3.2 Options - DNS Resolve .....	315
16.3.3.3 Options - Add to existing network.....	315
16.3.3.4 Options - Icmp Polling (if respond to request) .....	315
16.3.3.5 Options - SNMP Polling (if respond to request).....	315
16.3.3.6 Options - Enable Discover Scanning (if next-hop).....	315
16.3.3.7 Options - New Polling Time .....	315
16.3.4 Scan Option.....	315
16.3.4.1 Option – IP (ping) Scan .....	316
16.3.4.2 Option – Snmp Scan .....	316
16.3.4.3 Option – Check SnmpV2c .....	316
16.3.4.4 Option – Snmp CommunityRO .....	316
16.3.4.5 Option – Scan TCP Port Range .....	317
16.3.4.6 Option – Scan TCP Port List .....	317
16.3.4.7 Option – Advanced Scan / Audit Using LUA Script .....	318
16.3.4.7.1 Script IP Expert et stratégie associée .....	318
<b>17 Filtre d'Events et de Trap scriptés.....</b>	<b>320</b>
<b>17.1 Introduction.....</b>	<b>320</b>
<b>17.2 La gestion d'événements .....</b>	<b>320</b>
<b>17.3 Types généraux d'événements .....</b>	<b>320</b>
<b>17.4 Filtre d'événement LoriotPro Scripté.....</b>	<b>322</b>
17.4.1 Wizard d'insertion d'un Event LoriotPro.....	322
17.4.1.1 Exécution du filtre.....	327
17.4.2 Wizard d'insertion d'un Filtre de Trap SNMP .....	329
<b>17.5 Lancement du service 'Event / Trap Script Queue manager' .....</b>	<b>332</b>
17.5.1.1 Option – Set Timer .....	334
17.5.1.2 Option – Stop .....	335
17.5.1.3 Option – Start .....	335
<b>17.6 Génération de Script pour le module Event / Trap Script Queue manager' .....</b>	<b>335</b>
17.6.1 Structure des fichiers dynamiques .....	336
17.6.2 Variables associées aux fichiers d'Event LoriotPro .....	337
17.6.2.1 Exemple de fichier d'Event généré .....	337
17.6.3 Variables associées aux fichiers de TRAP SNMP .....	338
17.6.3.1 Exemple d'un fichier de Trap SNMP généré .....	338
17.6.4 Wizard de creation d'un fichier de script de filtre d'Event .....	339
17.6.4.1 Filtre d'Event LoriotPro .....	341
17.6.4.2 Variables d'Event .....	342
17.6.5 Wizard de création d'un fichier de script de filtre de TRAP SNMP .....	342
17.6.5.1 Filtre de Trap SNMP.....	342

---

17.6.5.2	Variables de Trap SNMP .....	343
17.6.6	Mise au point des scripts .....	344
<b>18</b>	<b>Utilisation direct des scripts .....</b>	<b>346</b>
<b>18.1</b>	<b>Introduction.....</b>	<b>346</b>
<b>18.2</b>	<b>A partir des menus du logiciel .....</b>	<b>346</b>
18.2.1.1	Menu contextuel d'un équipement.....	346
18.2.1.2	Menu principal.....	351
<b>18.3</b>	<b>A partir d'un Shortcut d'équipement .....</b>	<b>352</b>
<b>19</b>	<b>ANNEXES.....</b>	<b>354</b>
<b>19.1</b>	<b>Intégration du SLA dans LUA.....</b>	<b>354</b>
19.1.1	Introduction.....	354
<b>19.2</b>	<b>LoriotPro (Ipsla) Librairie.....</b>	<b>354</b>
<b>19.3</b>	<b>Tableau de définition de la librairie.....</b>	<b>354</b>
<b>19.4</b>	<b>Architecture des répertoires de collectes des SLA.....</b>	<b>355</b>
19.4.1	Analyse des fichiers.....	359
19.4.1.1	Codage du nom des fichiers.....	359
19.4.1.2	Contenus des fichiers.....	360
19.4.1.3	Détermination de l'intervalle de polling.....	361
19.4.1.4	Trou de collecte.....	362
19.4.1.5	Arrêt du logiciel.....	364
19.4.1.6	Arrêt global du polling des équipements .....	364
19.4.1.7	Arrêt d'un type de polling sur deux .....	365
19.4.2	SLA Conclusion.....	366
<b>20</b>	<b>Librairies externe LUA.....</b>	<b>369</b>
<b>20.1</b>	<b>Exemple LUA_ZIP .....</b>	<b>369</b>
<b>20.2</b>	<b>Exemple LUA_ODBC .....</b>	<b>374</b>
<b>20.3</b>	<b>Exemple LUA_COM .....</b>	<b>377</b>
20.3.1	Exemple d'utilisation de WMI .....	378
<b>20.4</b>	<b>Exemple LUA_SOCKET.....</b>	<b>381</b>
<b>20.5</b>	<b>Projet Lua_Wizard_80 .....</b>	<b>383</b>
<b>21</b>	<b>Matrice de compatibilité des librairies LUA de Loriotpro .....</b>	<b>386</b>
<b>21.1</b>	<b>License .....</b>	<b>387</b>
<b>21.2</b>	<b>Lua 5.0 license .....</b>	<b>387</b>
<b>21.3</b>	<b>Lua 4.0 license .....</b>	<b>388</b>
<b>21.4</b>	<b>LoriotPro End User License Agreement.....</b>	<b>388</b>

## 1 Pré requis

Ce document s'adresse à des utilisateurs possédant la connaissance des mécanismes SNMP et ayant des notions avancées de programmation. Une lecture préalable des documents relatifs au langage LUA est nécessaire pour comprendre la syntaxe utilisée et les mécanismes du langage. Le langage utilise une syntaxe et des termes anglo-saxons, une connaissance de la langue anglaise est donc très souhaitable. Une connaissance approfondie de l'utilisation du logiciel LoriotPro est indispensable.

Quelques liens pour vous permettre de mieux appréhender le contenu de cet ouvrage :

Les liens de l'éditeur du logiciel LoriotPro ou vous pourrez trouver des exemples et toute la documentation en ligne.

<http://www.loriotpro.com>  
<http://www.luteus.fr>  
<http://www.luteus.biz>

Un accès direct à la documentation du logiciel.

[http://www.luteus.biz/Download/LoriotPro\\_Doc/V4/LoriotPro\\_Documentation.htm](http://www.luteus.biz/Download/LoriotPro_Doc/V4/LoriotPro_Documentation.htm)  
[http://www.loriotpro.com/Products/On-line\\_DocumentationV3\\_ActiveView/ActiveView\\_Documentation.htm](http://www.loriotpro.com/Products/On-line_DocumentationV3_ActiveView/ActiveView_Documentation.htm)

Un accès aux principaux sites traitant de LUA. La documentation de base du langage LUA est mise en ligne sur le site <http://www.lua.org>

<http://www.lua.org>  
<http://lua-users.org/wiki/LuaLinks>  
<http://www.keplerproject.org/>  
<http://luaforge.net/>

...

Des portails sur le protocole SNMP

<http://www.snmplink.org/>  
<http://www.snmpworld.com/>

Des portails partenaires de LUTEUS

[http://www.loriotpro.com/Partners/PartnersList\\_EN.php](http://www.loriotpro.com/Partners/PartnersList_EN.php)

## 1.1 Conventions de lecture

### 1.1.1.1 Remarque

Malgré toute l'attention qui a été portée dans la réalisation de ce document, il est possible que des erreurs ou fautes se soit glissées. Nous vous prions par avance de nous en excuser et de bien vouloir nous en faire part à l'adresse Email suivante :

[support@loriotpro.com](mailto:support@loriotpro.com)

### 1.1.1.2 Portion de code

Les portions de codes sont encadrées :

```
dofile(lp.GetPath().."/config/script/loriotinit.lua");
dofile(lp.GetPath().."/config/script/1-clock-function.lua");

lp_value = 0;
lp_buffer ="error";
```

### 1.1.1.3 Commentaire dans le code

Les commentaires standard insérés dans le code sont en caractères verts.

```
-----  
-- start program  
-----  
  
dofile(lp.GetPath().."/config/script/loriotinit.lua");
dofile(lp.GetPath().."/config/script/1-clock-function.lua");

lp_value = 0;
lp_buffer ="error";
-- test if clock is already include in the map
nb=lpav.FindRef(1,1,"center","a");
if (nb~=0) then
error("Clock center object found, abort\n");
end
```

Des commandes et/ou fonctions peuvent être écrites en bleu ou rouge pour attirer votre attention sur certaines parties du code.

```
lp_value = 0;
lp_buffer ="error";
-- test if clock is already include in the map
nb=lpav.FindRef(1,1,"center","a");
if (nb~=0) then
error("Clock center object found, abort\n");
end
```

Dans le texte :

Les références à des menus ou des chemins de fichier sont en ***gras/Italique***.

#### 1.1.1.4 Signe d'attention

Ce signe vous avertit d'un problème possible ou d'une information importante.

**Attention** 

#### 1.1.1.5 Type des variables et syntaxe

Tous les variables écrites entre simple guillemet sont des chaînes de caractères.

##### Exemple

'caption' 'ip' 'oid'

Les variables sans simple guillemet, sauf information contraire sont des valeurs de type double (LUA number).

##### Exemple

value result

#### 1.1.1.6 Variable 'ip'

Toutes références à la variable 'ip' impliquent que cette adresse soit définie dans la Directory et soit correctement configurée. Toutes les fonctions LUA des librairies LUTEUS utilisent de façon transparente les paramètres des équipements représentés par leur adresse 'ip' respective. Si vous utilisez une fonction de requête SNMP, la fonction utilisera les paramètres SNMP de l'équipement tels que définis dans la Directory. La librairie offre plusieurs fonctions pour vérifier si une adresse IP est valide (présente dans la Directory)

**Attention** 

Il n'est pas possible de réaliser des fonctions de requêtes SNMP sur des équipements dont l'adresse 'ip' n'est pas dans la Directory.

#### 1.1.1.7 Variable 'oid'

Toutes références à un nom d'objet SNMP 'oid' impliquent que cet objet existe dans la data base des MIBs compilés. La librairie offre plusieurs fonctions pour vérifier la réalité d'un nom SNMP. En général, il s'agit d'un nom d'objet SNMP en forme non dotted, par exemple **sysname**, **syslocation**...

#### **1.1.1.8 Variable ‘uid’**

Toutes références à la variable ‘uid’ représentent un unique identifiant de type entier 64bits codé dans une chaîne de caractères. Logiquement tous les éléments de LoriotPro possèdent un ‘uid’ qui permette à certaines fonctions de réaliser des opérations sur ces objets.

#### **1.1.1.9 Variable ‘path’**

Les variables de type ‘path’ ou ‘file’ qui contiennent des caractères ‘\’ doivent être modifiées pour que le caractère ‘\’ soit doublé : ‘\\’ .

#### **Exemple**

**‘C:\\myfile.txt’**

#### **1.1.1.10 Variable ‘array’**

Les variables de type ‘array’ sont des tableaux LUA. Les fonctions des librairies LUTEUS vous demandent de fournir une chaîne de caractères qui sera utilisée par la fonction pour créer le tableau. Une fois le tableau créé, vous devrez l'utiliser dans votre code comme une tableau classique LUA (sans les guillemets simples).

## 2 Introduction

La version 4 de LoriotPro Extended Edition intégrait la capacité de réaliser des scripts par l'intermédiaire d'un langage interprété, intégré au logiciel. Cet interface programmable a permis la réalisation de solutions de supervision, de monitoring, de management d'infrastructures complexes débordant largement le cadre de la supervision ou du « monitoring » classique. Des solutions de géo-localisations graphiques, de contrôles de processus complexes et d'autres ont vu le jour. Nous pouvons même dire que chaque client a réalisé grâce à cette option du logiciel une solution particulière propre à son environnement. Fort de ce succès nous avons analysé les points fort et les faiblesses de la version 4 concernant cette interface. Le résultat de cette analyse s'est concrétisé dans la version 5 par une intégration encore plus performante de cette fonction et la capacité d'exploiter de façon beaucoup plus simple les scripts développés. Une infrastructure de système expert a été conçu au cœur du logiciel pour vous permettre de mieux exploiter la puissance des scripts. Le format des scripts a été standardisé pour permettre leur emploi automatique à travers des modules graphiques de plus haut niveau. Un nouveau module d'audit intégré au noyau du logiciel et exploitant les scripts a ainsi vu le jour et est contrôlable graphiquement ou via les scripts eux mêmes. Un module « IP Expert » permet une découverte et une auto-configuration des paramètres des équipements. Il est maintenant possible de lancer automatiquement un script sur réception d'un événement (Event, TRAP SNMP) en exploitant ses paramètres...

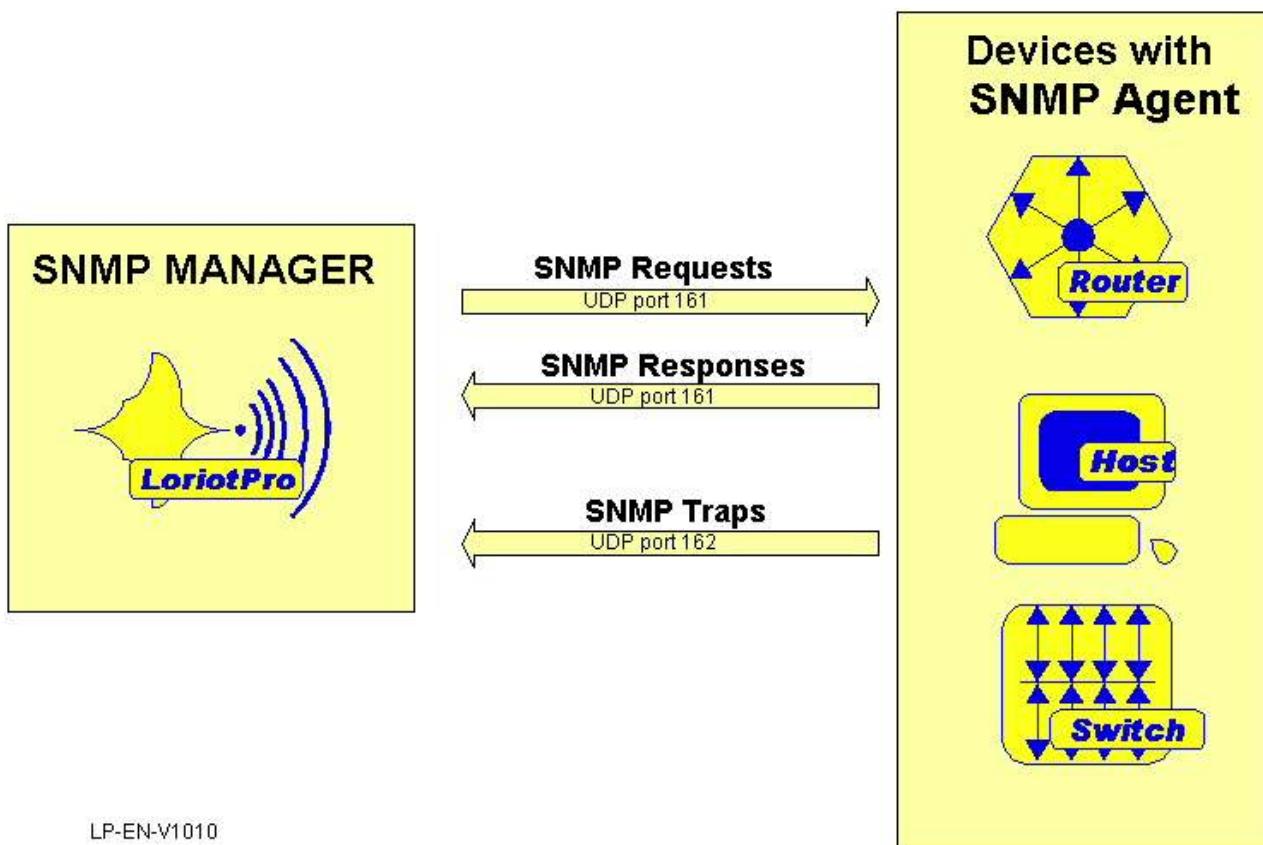
Nous espérons sincèrement que cette version 5 de l'Extended Edition de LoriotPro répondra à vos besoins. Depuis 2002 nos objectifs sont clairement affichés sur notre site [www.lriotpro.com](http://www.lriotpro.com) et sont, nous l'espérons, pleinement validés par cette nouvelle version.

Les missions de Luteus sont (entre autres):

- Aider les administrateurs de réseaux et de systèmes dans leurs tâches quotidiennes pour garantir aux utilisateurs disponibilité et performance des accès à l'information.
- De créer des logiciels puissants et conviviaux visant à diminuer les temps de prise en main et à faciliter leur exploitation.
- De donner de l'autonomie aux administrateurs et de fournir de l'information visuelle synthétique et/ou détaillée.
- De fournir des produits modulaires pour répondre à des besoins identifiés avec des prix optimisés.

### 3 Introduction au modèle SNMP

Le modèle d'administration SNMP présente bien des avantages mais se limite à la collecte de variables fournies par un agent intégré à l'équipement à administrer. Le manager demande le contenu d'une variable via le protocole SNMP à l'agent qui lui répond en fournissant le contenu de la variable via le même protocole ou bien « NULL » si l'agent ne supporte pas cette variable. Le protocole SNMP n'est qu'un mécanisme de stockage et d'échange de paramètres entre deux équipements hiérarchisés, le manager et l'agent.



#### 3.1 Limitation du protocole SNMP

Le protocole SNMP n'intègre pas la notion de traitement mathématique ou logique des variables entre elles. Ce travail est laissé à la charge du logiciel d'administration, du 'Manager'. Un exemple simple peut illustrer notre propos ; si nous désirons connaître le pourcentage d'utilisation du disque dur C:/ de notre agent, un serveur Microsoft par exemple, l'administrateur doit pouvoir collecter ce paramètre via SNMP, malheureusement ce paramètre n'existe pas. Les seules valeurs standard existantes sont contenues dans la RFC 'rfc2790.mib', **HOST-RESOURCES-MIB** et ne fournissent pas directement la réponse à la question, de plus ces valeurs sont indexées par rapport au nombre de disques de stockage de l'agent. Un traitement va

donc être indispensable. Ce premier constat apporte quelques pistes de travail et une méthode structurée de résolution, il nous faudra dans l'ordre :

1. Connaître les objets SNMP permettant la collecte (recherche documentaire)
2. Trouver l'index dans la table **Hrstorageentry** du disque C :
3. Collecter les valeurs du disque C : permettant le calcul
4. Faire le calcul
5. Afficher le résultat

Ce raisonnement de travail s'applique à tous les cas de figure où l'agent SNMP d'un équipement n'est pas capable de fournir un résultat direct. Toute autre logique que l'analyse d'une valeur brute, un nombre d'octets passés en entrée sur un équipement avec l'objet SNMP **ipinreceives.0** par exemple, ou bien un statut de réponse à un ping, demande un traitement de l'information pour corrélérer ou agréger des valeurs.

## 3.2 Modules dédiés de LoriotPro

Pour faire face à ce problème d'analyse des valeurs SNMP collectées, LoriotPro intègre un nombre important de fonctions, modules et Plugins assurant de base les logiques de traitement les plus courantes pour ces valeurs collectées. En plus du nombre important de modules intégrés au noyau du logiciel, il existe trois sortes de plugins qui répondent à des besoins distincts.

### 3.2.1 Les plugins attachés aux équipements

Ces plugins sont attachés dans la Directory aux équipements (hosts) et réalisent des tâches dédiées. Ils possèdent tous une interface graphique conviviale pour faciliter la tâche de l'administrateur. Consultez la documentation générale du logiciel pour plus d'information concernant ces modules.

Nom	Fonction
Active View	Traitement graphique des variables SNMP, et de bien d'autres fonctions.
Bulk TCP Poller	Traitement de résultats de requête TCP.
Bulk Threshold Control	Traitement de résultats de requête SNMP simple.
Cisco IOS Configuration Surveyor	Traitement dédié des variables SNMP Cisco contrôlant les sauvegardes de configuration.
Cisco ISDN Call Collector	Traitement dédié des variables SNMP Cisco gérant l'historique des Calls.
Events Counter	Module d'agrégation des événements.
HTTP Graphics MIB II Box	Exploitation graphique des variables SNMP permettant la supervision d'un équipement.
Interface Monitor	Analyse des variables SNMP associées aux interfaces d'un équipement.
Linear Graph	Graphe de variables SNMP simples.
Grapher	Graphe de variables SNMP simples.
VuMeter Graph	Graphe d'une variable SNMP simple.
Report Scheduler	Génération de rapport exploitant des collectes

	de variables SNMP.
Poll TCP Service	Traitemet de résultats de requête TCP.
Poll URL Link Statistic	Traitemet de résultats de requête HTTP.
ProcessSurveyor	Traitemet dédié de variables SNMP associées aux processus d'un équipement.
SLA ReportCenter	Affichage des résultats de temps de réponse des collectes.
RRD Graph Collector	Graphe de variables SNMP et autres complexes.
SNMP Table Audit	Analyse d'un tableau de variables SNMP.
Spanning Tree Bridge Map	Analyse et exploitation graphique des variables SNMP associées à un bridge supportant le protocole Spanning-Tree
Traps Filter Counter	Module d'agrégation des TRAP SNMP.
Trend View	Graphe de variables SNMP complexes en tendance.

### 3.2.2 Les plugins de service

Ces plugins sont des services qui tournent en dehors de la Directory et permettent par exemple l'utilisation d'un serveur Web. Ils possèdent tous une interface graphique conviviale pour faciliter la tâche de l'administrateur. Consultez la documentation générale du logiciel pour plus d'information concernant ces modules.

Nom	Fonction
HTTP ISAPI Server	Un serveur Web pour une visualisation distante du logiciel.
Syslog Collector Manager	Un module de visualisation des Syslog reçus par le logiciel.
Dynamic DNS To IP	Un module de 'management' des associations d'adresses IP et de noms d'équipement.
Internetwork Path Route	Un module graphique exploitant les variables SNMP de plusieurs équipements pour visualiser le chemin pris par un paquet IP dans un réseau d'un point à un autre.
Statistic DashBoard	Un module d'affichage de statistiques.
NetFlowCollector Service	Un module de collecte de variable Netflow.
MIB Object Description Browser	Un module d'aide à la gestion des variables SNMP.
Netbios Name Resolver	Un module de résolution de noms Netbios.
Cache monitor	Un module d'optimisation des noms de variables SNMP.
Event Delayed Correlator	Un module de gestion des Events reçus.
Event / Trap Scripter	Un module de gestion des Events ou TRAPs SNMP reçus avec des fonctions de scripts associés.
RRD manager	Un gestionnaire de base de données RRD qui

	permet la génération de rapport et de graphes.
EMAIL Event Scheduler	Un module permettant l'émission de EMAIL.
TraceRoute Discover	Un module graphique de type TraceRoute intégrant un analyseur temporel.
Transparent Windows Manager	Une module de gestion de l'interface graphique du logiciel.
TrapSimulator	Un module permettant la simulation de réception de TRAPs SNMP pour analyser et réaliser des filtres cohérents.
Text To Speech	Un module de conversion de texte vers la voix pour des alertes interactives.
User Task	Un module de gestion de l'interface graphique du logiciel.

### 3.2.3 Les plugins directs

Ces plugins correspondent à des outils que l'administrateur utilise à sa convenance de façon unitaire. Ils possèdent tous une interface graphique conviviale pour faciliter la tâche de l'administrateur. Consultez la documentation générale du logiciel pour plus d'information concernant ces modules.

Nom	Fonction
Active View Box	Un module de représentation graphique d'un équipement sous la forme d'une face avant virtuelle de l'équipement.
Bulk Host Configuration	Un module de configuration par lot des équipements.
Cisco IOS Configuration Requester	Un module pour télécharger des configurations dans des équipements Cisco.
Cisco ISDN stats	Un module de gestion des collectes SNMP des historiques de Calls des routeurs Cisco.
Syslog Collector Messages Browser	Un module d'affichage des Syslogs reçus par le logiciel.
Events Filter Counter	Un module d'agrégation et de statistiques concernant les filtres d'Events LoriotPro.
Events Browser	Un module de sélection et d'affichage des Events reçus par le logiciel.
Host Properties	Un module de configuration des paramètres (LoriotPro) d'un équipement.
HTTP Report Manager	Un module de configuration des droits utilisateurs sur les rapports exécutables via le serveur Web de LoriotPro.
LoriotPro Pluging Sample : SetSnmpValue	Un module d'exemple pour modifier une valeur SNMP sur un équipement.
HTTP User Name Manager	Un module de configuration des droits utilisateurs sur le serveur Web de LoriotPro.
Whois Requester	Un module de requête Whois.

Malgré le nombre important d'outils disponibles en standard avec le logiciel, l'expérience montre que les besoins d'un administrateur réseau sont très gourmands en fonctionnalités non standard. La version 5 du logiciel LoriotPro a donc été repensée pour fournir à l'administrateur réseau une plus grande puissance de traitement de l'information tout en alliant une plus grande facilité d'utilisation du produit.

### 3.3 Fonction classique d'un NMS

Ce chapitre fait un état des lieux des fonctionnalités offertes par des logiciels classiques de supervision de réseaux. (**Networks Management System**)

Les MMS standard proposent en général la découverte de l'infrastructure par des pollings force brute (scan) en ICMP/SNMP ou TCP. Certains utiliseront même des outils logiciel du domaine public tel que NMAP. (<http://nmap.org/>)

Les statuts des équipements seront ensuite affichés dans des MAPs graphiques de Niveau 3 (routage avec interconnexion) ou similaires et des listes, quelquefois sous forme d'arbres structurés. Le statut affiché d'un équipement sera le reflet d'une réponse basique de type oui/non.

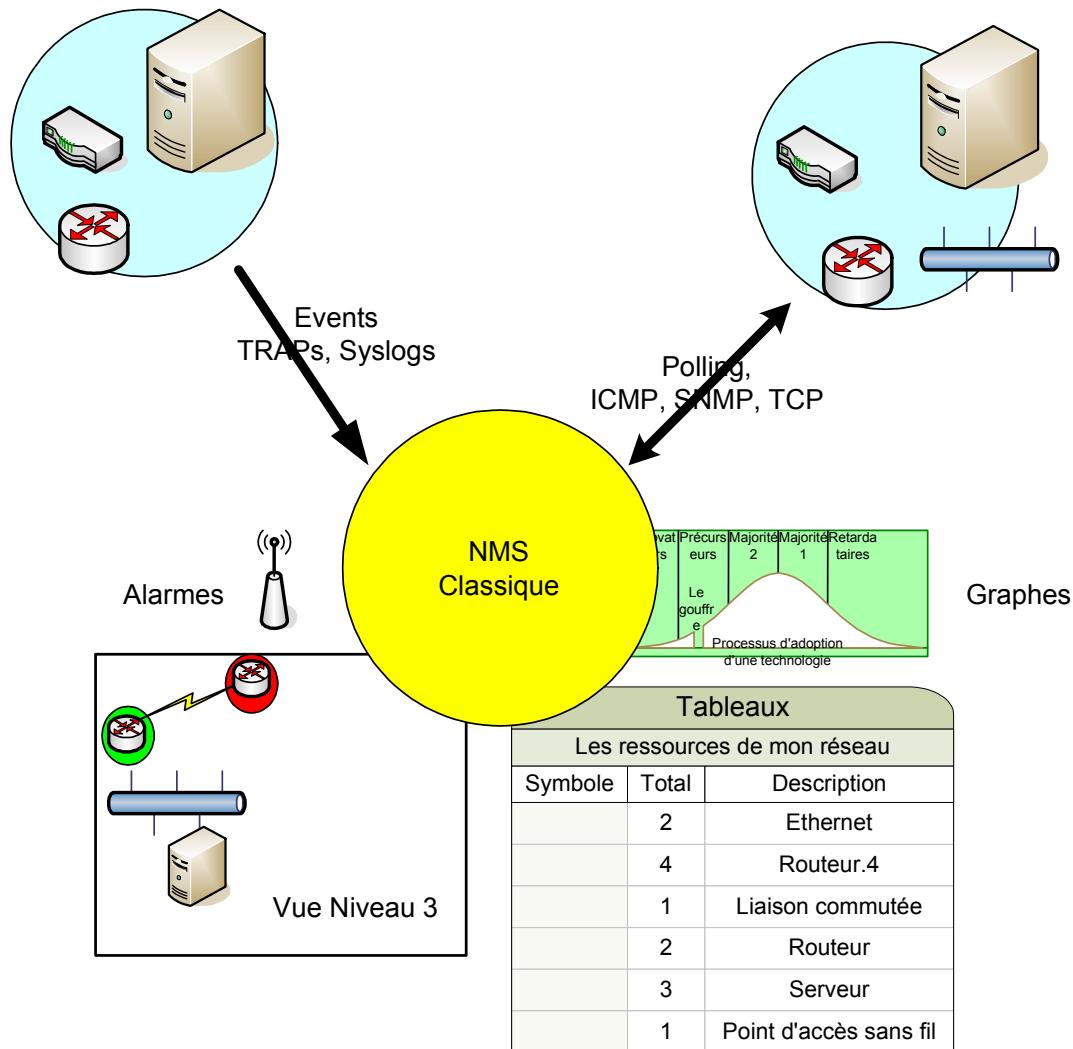
Il sera possible de collecter des valeurs via SNMP pour réaliser des tests simples ou bien grapher ces valeurs.

Les Syslogs et les TRAPs SNMP seront stockés et un filtrage sera parfois possible.

Les temps de réponse aux requêtes seront stockés pour un calcul de SLA simple.

Pour réaliser des opérations plus complexes il faudra faire appel à des logiciels spécialisés souvent très coûteux et limités à un constructeur et quelques noeuds. En général le logiciel sera peu souple et offrira des fonctions toutes faites ne correspondant pas forcément aux besoins du client (on fera avec !).

Le schéma suivant représente le spectre couvert par un NMS classique.



### 3.4 Limitation d'un NMS classique

Dans ce chapitre, certaines des limitations des NMS classiques seront démontrées par des questions et des besoins récurrents d'administrateur ou de DSI.

Un Scan d'équipements ne fournit que des informations de type réseau.

- Le processus notepad.exe tourne-t-il sur cet équipement ?
- Ma table oracle ORACLEBASE1 est-elle saturée ?
- Le patch Microsoft K00001 est-il installé sur les serveurs de mon entreprise ?
- Les équipements du service de comptabilité sont-ils opérants ?

Une MAP graphique de niveau 3 ou 2 ne fournit pas une logique organisationnelle ou de type « Dashboard »

- Sur le network 192.168.1.0 quels sont les PC des Supports ?
- Où sont les imprimantes sur mon MAN ?
- Comment réduire la MAP à un ou deux indicateurs graphiques ?

- Comment positionner les camions frigorifiques de mon entreprise dynamiquement sur une MAP graphique avec une carte de France ?
- Si la porte du couloir est ouverte, je veux voir une porte ouverte sur la MAP et recevoir un EMAIL !!

Il faut parfois faire des calculs avec les objets SNMP pour obtenir une valeur et faire un graphe ou un test

- $(\text{hrstorageused.2} / \text{hrstoragesize.2}) * 100 = \text{Usage \%}$
- Comment afficher la capacité d'espace disque restant sur les serveurs du 1er étage ?

Un TRAP ou un Syslog est générique (Un Link down contient des paramètres )

- Comment faire la différence entre la rupture d'un lien important et la déconnexion d'un PC au commutateur ?
- Comment interpréter un message texte sans logique apparente d'un syslog ?

Le calcul de SLA n'est pas forcément assigné à des équipements uniques .

- Comment afficher le SLA des SLA de mes serveurs HTTP du service formation sur la dernière semaine.

Etc....

### **3.5 Exploitation d'un infrastructure communicante**

Acquérir la connaissance de son infrastructure globale est le premier pas vers la maîtrise de son environnement. Pouvoir exploiter cette infrastructure est le deuxième pas.

Un réseau de PME/PMI possédant des sites distants, des agences ou des usines compte rapidement plusieurs centaines voir plusieurs milliers d'équipements communicants devant être supervisés. Il est possible de configurer les logiciels manuellement mais dans le cas de réseaux de communication importants il est indispensable de faire appel à des modules d'auto découverte et d'auto configuration. Les processus de découvertes sont complexes car certains équipements peuvent être cachés ou difficilement accessibles ; de plus que faire des équipements découverts ?

Un nombre important de questions et de difficultés se pose à l'administrateur.

- L'équipement est équipé d'un firewall qui le protège des scans ; comment le détecter ?
- Quelles sont la community et la version SNMP utilisées par cet équipement ?
- Comment être sûr du type de cet équipement pour l'auto configurer (dans l'annuaire).
- L'équipement doit-il être pollé ?

- L'équipement fait-il partie de l'entreprise ?
- Doit-on auditer les processus découvert sur l'équipement ?
- Une supervision des espaces disques doit-elle être réalisée sur cet équipement ?
- L'équipement est un routeur ; doit-on grapher tous ses interfaces ?
- Doit-on auditer les ports de ce commutateur ?
- Quel intervalle de polling doit être positionnée sur cet équipement ?
- Visiblement cet équipement a un service FTP et/ou autre actif ; est-il légal ?
- Cet équipement n'a pas la dernière version de patch installée ; faut-il envoyer une alerte ?
- Cet équipement est une imprimante ; faut-il auditer les bacs à feuilles ou les cartouches d'encre ?
- Cet équipement est un CALL Manager ; faut-il auditer les téléphones attachés à sa configuration ?
- ...

### **3.6 Constat concernant le management d'une infrastructure communicante**

- **CHAQUE CLIENT A UN BESOIN DIFFERENT !!**
- **AUCUN PRODUIT DEDIE NE REPOND A 100% AU BESOIN D'UN CLIENT.**
- **IL FAUDRAIT UN PRODUIT PAR CLIENT.**

### **3.7 Nécessité de l'intégration d'une couche programmable**

Dans notre introduction, nous avons montré que le simple fait de vouloir afficher le pourcentage d'utilisation d'un disque dur nous oblige à réaliser une série de requêtes et un calcul. Nous allons détailler les différentes phases nécessaires pour réaliser ce travail afin de vous permettre de bien appréhender la complexité du problème posé et essayer de démontrer la nécessité de disposer d'une couche programmable intégrée au logiciel.

#### **3.7.1 Calcul de pourcentage d'utilisation d'un disque dur**

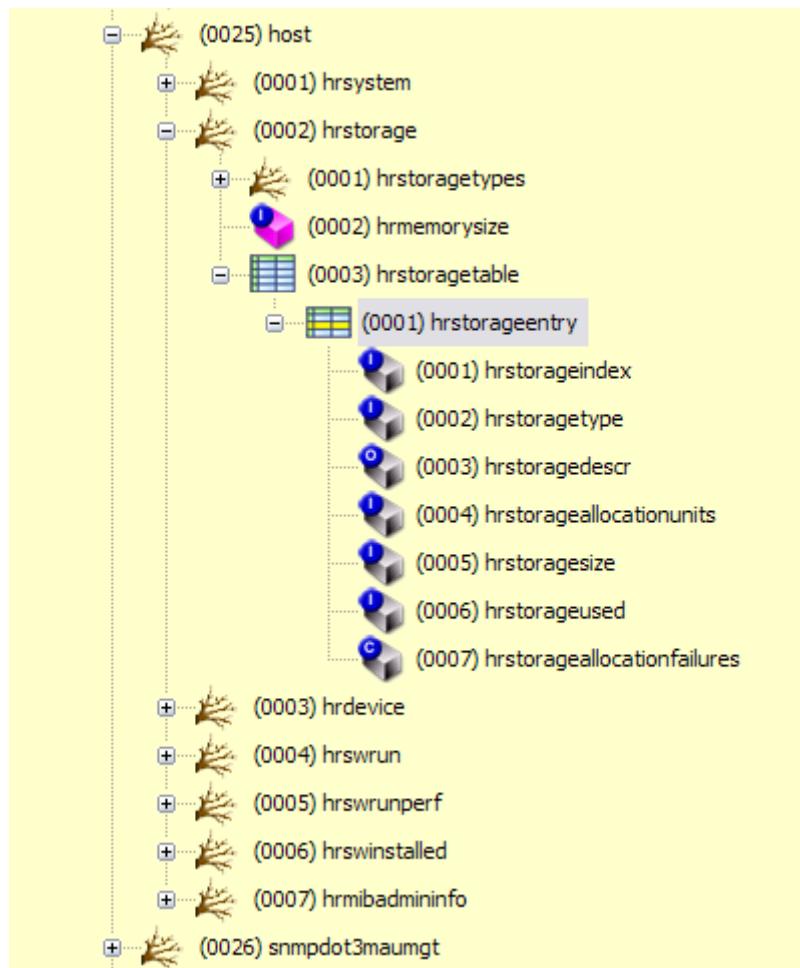
Si nous désirons connaître le pourcentage d'utilisation du disque dur **C:/** de notre agent, un serveur Microsoft par exemple, l'administrateur doit pouvoir collecter ce paramètre via SNMP ; malheureusement, ce paramètre n'existe pas. Les seules valeurs standard existantes sont contenues dans la RFC 2790 (**bin/mins/rfc2790.mib**), **HOST-RESOURCES-MIB** et ne fournissent pas directement la réponse à la question ; de plus ces valeurs sont indexées par rapport au nombre de disques de stockage de l'agent. Un traitement va donc être indispensable. Nous retrouvons les cinq phases décrites dans un chapitre précédent.

1. Connaître les objets SNMP permettant la collecte (ceci implique une recherche documentaire)
2. Trouver l'index dans la table **Hrstorageentry** du disque C :

3. Collecter les valeurs du disque C:/ permettant le calcul
4. Faire le calcul
5. Afficher le résultat

### 3.7.1.1 Phase 1 : recherche documentaire

La phase 1 qui consiste à rechercher les objets SNMP à collecter pour réaliser le traitement, est certainement la plus complexe car elle oblige à une étude des fichiers de MIB et des documents du constructeur. Dans notre exemple nous utilisons les variables fournies par la rfc2790 qui est un document standard simple. Pour de plus amples détails reportez-vous au fichier rfc2790.mib se trouvant dans le répertoire mibs du logiciel ainsi qu'à la documentation du logiciel LoriotPro qui contient un chapitre d'introduction au protocole SNMP. L'étude de la RFC2790 nous apprend qu'il existe une table de variables maintenue par l'agent, **Hrstorageentry**, qui peut nous fournir les informations souhaitées.



Affichage de la table Hrstorageentry à partir de LoriotPro

### Description des objets SNMP de la table

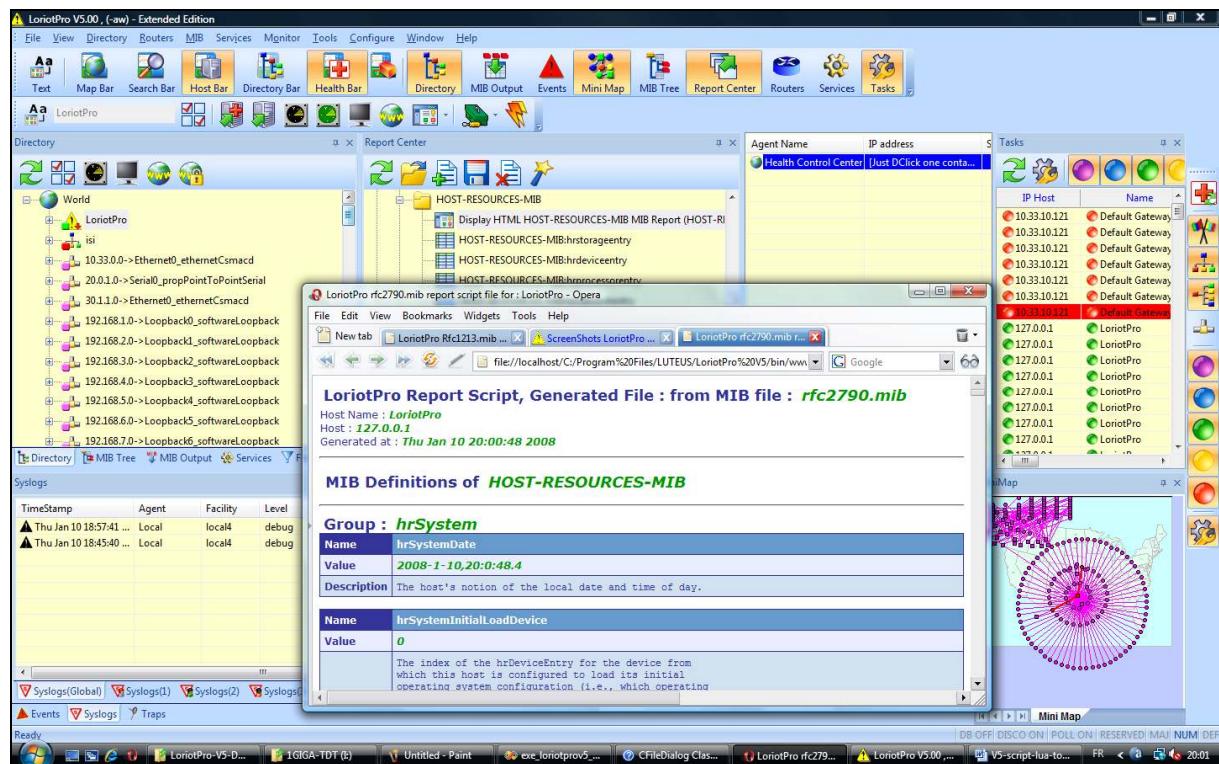
LoriotPro V5.00 Extended Edition – Supervision et gestion SNMP par scripts LUA  
 Introduction au System Expert et aux scripts

<b>Objet SNMP</b>	<b>Description</b>
Hrstorageentry OID LEN (10) : 1.3.6.1.2.1.25.2.3.1.	“A (conceptual) entry for one logical storage area on the host. As an example, an instance of the hrStorageType object might be named hrStorageType.3”
Hrstorageindex OID LEN (11) : 1.3.6.1.2.1.25.2.3.1.1.	“A unique value for each logical storage area contained by the host. »
Hrstoragetype OID LEN (11) : 1.3.6.1.2.1.25.2.3.1.2.	“The type of storage represented by this entry.”
Hrstoragedescr OID LEN (11) : 1.3.6.1.2.1.25.2.3.1.3.	“A description of the type and instance of the storage described by this entry.”
Hrstorageallocationunits OID LEN (11) : 1.3.6.1.2.1.25.2.3.1.4.	“The size, in bytes, of the data objects allocated from this pool. If this entry is monitoring sectors, blocks, buffers, or packets, for example, this number will commonly be greater than one. Otherwise this number will typically be one. »
Hrstoragesize OID LEN (11) : 1.3.6.1.2.1.25.2.3.1.5.	“The size of the storage represented by this entry, in units of hrStorageAllocationUnits. This object is writable to allow remote configuration of the size of the storage area in those cases where such an operation makes sense and is possible on the underlying system. For example, the amount of main memory allocated to a buffer pool might be modified or the amount of disk space allocated to virtual memory might be modified.”
Hrstorageused OID LEN (11) : 1.3.6.1.2.1.25.2.3.1.6.	“The amount of the storage represented by this entry that is allocated, in units of hrStorageAllocationUnits. »
Hrstorageallocationfailures OID LEN (11) : 1.3.6.1.2.1.25.2.3.1.7.	“The number of requests for storage represented by this entry that could not be honored due to not enough storage. It should be noted that as this object has a SYNTAX of Counter32, that it does not have a defined initial value. However, it is recommended that this object be initialized to zero, even though management stations must not depend on such an initialization.”

Il est aussi possible d'utiliser le « Report Center » pour vous aider dans la recherche documentaire :

# LoriotPro V5.00 Extended Edition – Supervision et gestion SNMP par scripts LUA

## Introduction au System Expert et aux scripts

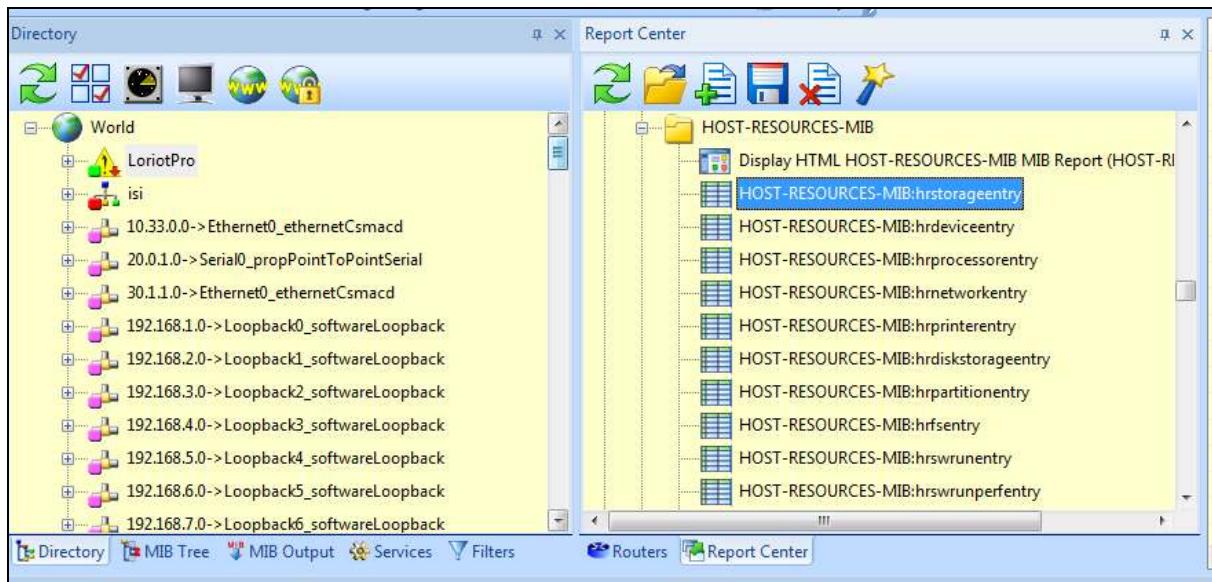


Le pourcentage d'utilisation de notre disque dur C : sera obtenu en collectant le nombre de blocs utilisés du disque (**hrstorageused**) puis en le divisant par le nombre total de blocs (**hrstoragesize**) puis en le multipliant par 100 pour avoir un résultat en pourcentage.

Si nous réalisons une collecte manuelle de la table **Hrstorageentry** à partir de LoriotPro, nous pouvons faire le calcul manuellement.

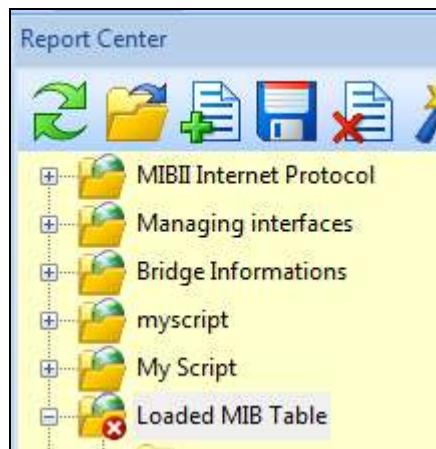
### 3.7.1.2 Phase 2 : trouver l'index du disque C :

Nous pouvons utiliser les ressources de LoriotPro pour trouver l'index du disque C:/.

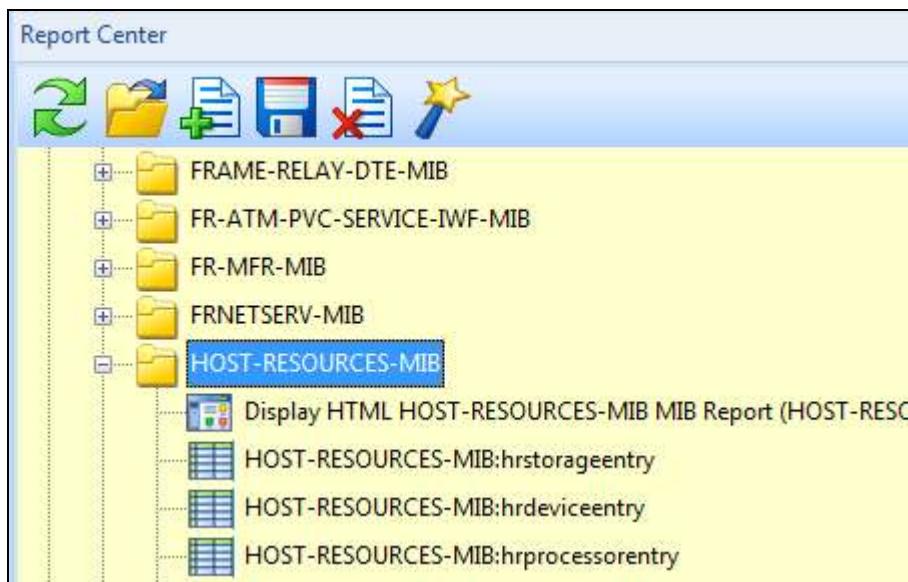


A partir du Report Center double cliquez sur le champ :

### Loaded MIB Table



Puis sur sur **HOST-RESSOURCES-MIB**



Et enfin **HOST-RESOURCES-MIB :hrstorageentry**

LoriotPro contacte l'équipement par défaut préalablement sélectionné dans la Directory et affiche le contenu de la table.

Dans notre exemple le disque C:/ possède l'index 2

Mib Table [hrstorageentry] for 127.0.0.1/LoriotPro								
hrstorageindex	hrStorageIndex	hrStorageType	hrStorageDescr	hrStorageAllocationUnits	hrStorageSize	hrStorageUsed	hrStorageAllocationFailures	hrStorageAllocationFailures
1	1	hrstorageremovabledisk	A:\	0	0	0	0	0
2	2	hrstoragefixeddisk	C:\ Label:BOOT Serial Number c4cefa82	4096	19454707	18284132	0	0
3	3	hrstoragefixeddisk	D:\ Label:BACKUP Serial Number 30d40c9f	4096	18432571	6497184	0	0
4	4	hrstoragefixeddisk	E:\ Label:RECOVER Serial Number 34d5fa2d	4096	1182471	996154	0	0
5	5	hrstoragecompactdisc	F:\	0	0	0	0	0
6	6	hrstoragecompactdisc	G:\	0	0	0	0	0
7	7	hrstorageremovabledisk	H:\	0	0	0	0	0
8	8	hrstorageremovabledisk	I:\	0	0	0	0	0
9	9	hrstorageremovabledisk	J:\	0	0	0	0	0
10	10	hrstorageremovabledisk	K:\	0	0	0	0	0
11	11	hrstoragevirtualmemory	Virtual Memory	65536	19971	7147	0	0
12	12	hrstorageram	Physical Memory	65536	8183	5537	0	0

### 3.7.1.3 Phase 3 : Collecter les valeurs

Dans cet exemple, il va falloir collecter 2 valeurs pour réaliser notre calcul. Nous pouvons lire le tableau de valeur que nous avons manuellement collecté.

**hrstorageused.2 = 18284132**

**hrstoragesize.2 = 19454707**

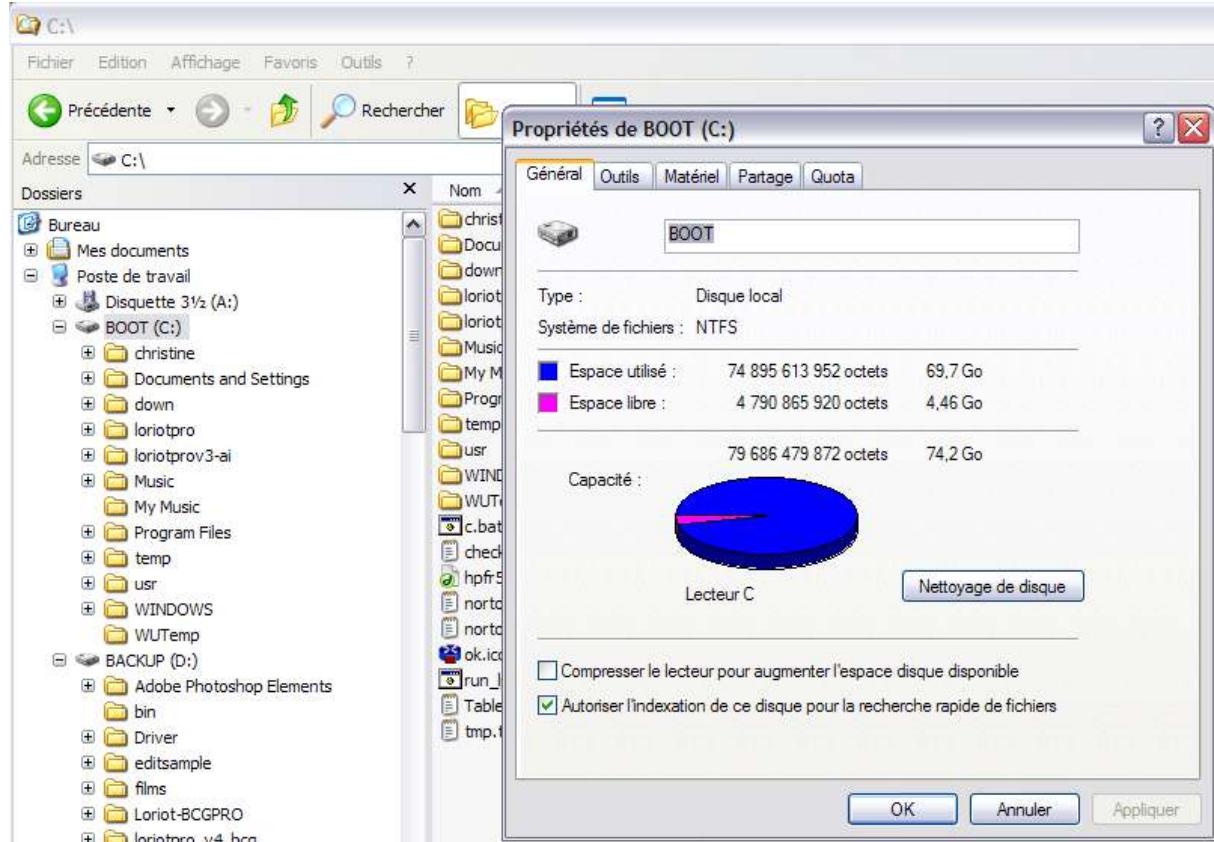
### 3.7.1.4 Phase 4 : faire le calcul

Nous devons appliquer la formule suivante :

$$(hrstorageused.2 / hrstoragesize.2) *100 = Usage \%$$

Disque	Résultat
C :	(18284132/19454707)*100=93.9%

Si nous regardons les propriétés du disque à partir de l'interface de l'explorer :



Nous avons une information qui nous confirme le résultat, ici la capacité du disque est en GigaOctet.

Disque	Résultat
C :	(69,7/74,2)*100=93.9%

Si nous voulons connaître via SNMP la capacité restante du disque C : il faut collecter la variable **hrstorageallocationunits.2** et la multiplier par **hrstoragesize.2** pour avoir le taille maximun du disque.

Soit en respectant l'indexation du disque C : « .2 ».

**hrstorageallocationunits.2 \* (hrstoragesize.2 – hrstorageused.2) =**

**4096 \* (19454707 - 18284132 ) = 4794675200 octets**

que nous pouvons transformer en GigaOctet :

**1024(kilo)\*1024(mega)\*1024(giga) = 1073741824**

(4794675200)/1073741824 = 4,4653 GigaOctets

### 3.7.1.5 Phase 5 : afficher le résultat

A ce niveau de la documentation nous n'avons pas encore abordé les capacités offertes par l'intégration de LUA dans LoriotPro. Ce sujet sera abordé plus en détail dans la suite du document. Le tableau suivant résume les différentes possibilités d'affichage en fonction de l'environnement utilisé pour lancer un script.

Environnements	Type d'affichages
Objet snmp virtuel Collecte SNMP standard/plugin	Tous les affichages LoriotPro existants Graphiques (mrtg, trendview, etc...) Texte Tableau
ActiveView via objet snmp virtuel calcul direct Script	Texte Graphique (objet de la map)
Script lancer à partir des menus	Texte
ShortCut d'un host	Texte Graphique (ActiveView Modal)
IP Expert	Texte (arbre d'affichage)
IP Scanner	Texte (arbre d'affichage)
Bulk Configuration	Texte
IP Audit	Texte, graph
Filtre d'Events et de Traps scriptés	Programmable

La suite du document présente en détail les capacités d'affichage des résultats en fonction de l'environnement utilisé pour lancer un script.

### 3.7.2 Conclusion

Cet exemple simple nous montre que les variables SNMP utilisées en direct ne nous permettent pas toujours d'obtenir le résultat que nous souhaitons et un calcul mathématique et/ou une analyse est très souvent nécessaire.

Un autre exemple simple est celui des séquences, en effet certaines variables SNMP retournent une valeur entière qui correspond en réalité à un statut. Il peut alors être nécessaire en fonction de la valeur renvoyée de collecter une autre variable. Dans ce cas une opération de tri ou de choix multiple doit être réalisée.

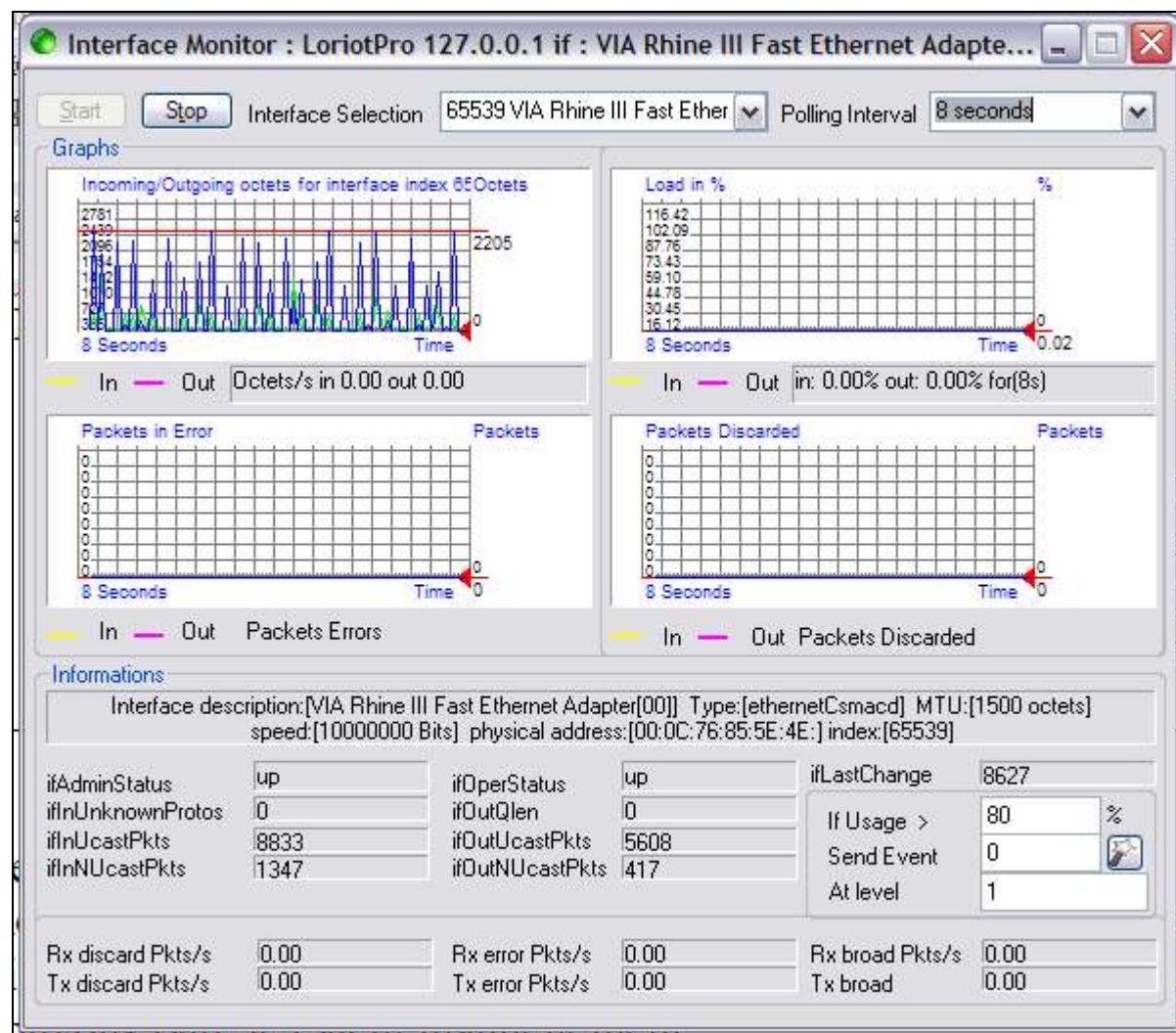
L'objet **sysServices** de la RFC1213 retourne une valeur qui est codée, un script ou un traitement est indispensable pour pouvoir interpréter le résultat de la collecte.

### RFC1213-MIB

sysServices OBJECT-TYPE
-------------------------

```
SYNTAX  INTEGER (0..127)
ACCESS  read-only
STATUS   mandatory
DESCRIPTION
    "A value which indicates the set of services that
    this entity primarily offers.
    The value is a sum. This sum initially takes the
    value zero. Then, for each layer, L, in the range
    1 through 7, that this node performs transactions
    for, 2 raised to (L - 1) is added to the sum. For
    example, a node which performs primarily routing
    functions would have a value of 4 (2^(3-1)). In
    contrast, a node which is a host offering
    application services would have a value of 72
    (2^(4-1) + 2^(7-1)). Note that in the context of
    the Internet suite of protocols, values should be
    calculated accordingly:
        layer  functionality
            1  physical (e.g., repeaters)
            2  datalink/subnetwork (e.g., bridges)
            3  internet (e.g., IP gateways)
            4  end-to-end (e.g., IP hosts)
            7  applications (e.g., mail relays)
    For systems including OSI protocols, layers 5 and
    6 may also be counted."
::= { system 7 }
```

En standard LoriotPro réalise un certain nombre d'opérations mathématiques ou des choix sur des variables standard, le plugin « Interface Monitor » est un bon exemple :



Malheureusement, en standard, LoriotPro ne peut pas répondre à l'ensemble des besoins qui sont infinis ; de plus il est possible de charger à partir du compilateur de MIB de LoriotPro de nouvelles listes de variables (fichier de MIB). Il faut donc pouvoir disposer d'une interface puissante permettant des traitements de toute sorte sur les nouvelles variables collectées et s'intégrant de façon simple à l'existant. Le reste du document va nous montrer comment répondre avec l'aide de scripts aux 5 phases nécessaires à l'intégration de collectes complexes sur les agents SNMP.

## 4 Traitement efficace de l'information

Après ces différents constats concernant l'existant, une étude à été entreprise pour définir les d'objectifs à remplir par la version 5 de LoriotPro pour faire face aux différents problèmes soulevés.

L'objectif principal de LoriotPro est de fournir à l'administrateur la capacité de moniturer, auditer et visualiser graphiquement tout type d'équipement et de système quel qu'en soit l'éditeur ou le constructeur du niveau 1 à 7 de la couche OSI en autorisant la corrélation et/ou l'agrégation de l'information.

LoriotPro doit être évolutif, et permettre de couvrir les besoins simples des petites entreprises mais aussi les besoins complexes d'opérateurs de télécommunication ou de services.

LoriotPro doit pouvoir sortir du cadre classique du monitoring, de la supervision ou du contrôle de processus et permettre un niveau de configuration le plus libre possible du logiciel.

LoriotPro doit supporter des logiques de sécurité informatique à travers la supervision, le monitoring et la collecte d'événements et leurs éventuelles corrélations (Extended Edition).

LoriotPro doit intégrer des fonctions avancées d'auto configuration d'expertises et d'audits automatisés pour simplifier la tâche des exploitants (Extended Edition).

### 4.1 La solution LoriotPro V5

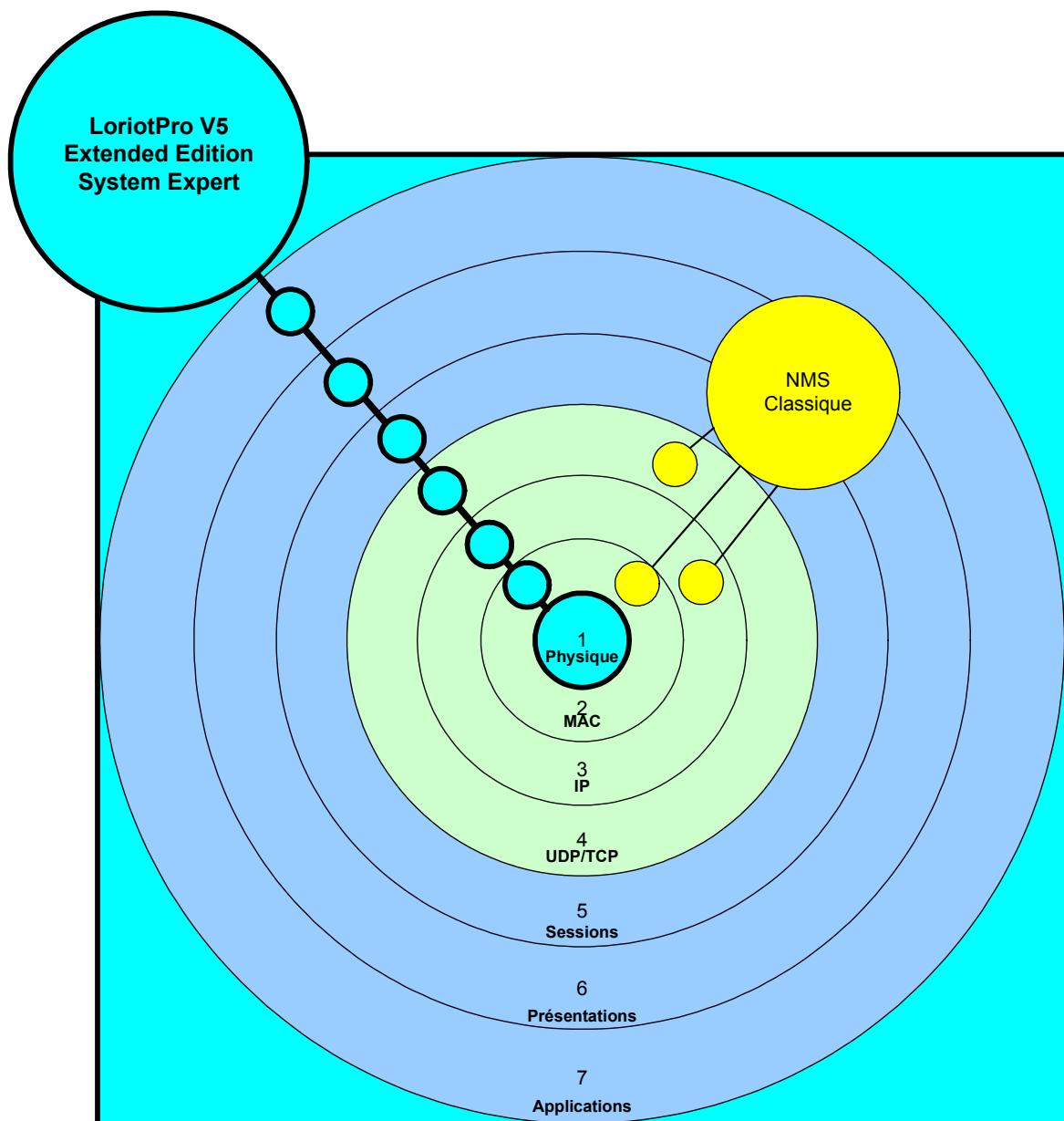
A l'issue de notre étude, notre service de développement a élaboré de nouveaux modules pour faire face au mieux aux besoins de traitement de l'information. LoriotPro V5 intègre bien sûr toutes les fonctions de base d'un NMS classique et bien plus, il n'est pas limité à un constructeur mais conçu pour communiquer avec le plus grand nombre d'équipements possibles.

LoriotPro V5 « extended Edition » intègre un système expert ouvert (Open Source) multi modules, basé sur des scripts permettant de réaliser tout type de réponses de supervisions, monitorings, audits, auto configurations, corrélations complexes, représentations graphiques dynamiques, etc...

LoriotPro V5 intègre un environnement de développement et plusieurs couches programmables allant du simple script interprété à la réalisation de plugins et/ou dll permettant d'enrichir le produit de modules dédiés écrits sur mesure.

#### 4.1.1 Spectre d'expertise de LoriotPro V5

LoriotPro V5 Extended Edition Permet de couvrir des besoins de la couche 1 à 7 du modèle OSI et ajoute une couche supplémentaire qui est la corrélation des 7 premières couches. La couche 8, couche implicite du modèle OSI est l'utilisateur, l'administrateur ou l'Expert réseau. LoriotPro V5 a la vocation d'être cet Expert réseau. La Version 5 du logiciel a été réécrite de manière à fournir à l'administrateur un système expert entièrement modifiable basé sur des scripts Open Source répartis sous forme de modules au sein du logiciel. Ce document a pour objectif de démontrer les capacités étendues du logiciel et d'expliquer son mode de fonctionnement et de configuration.



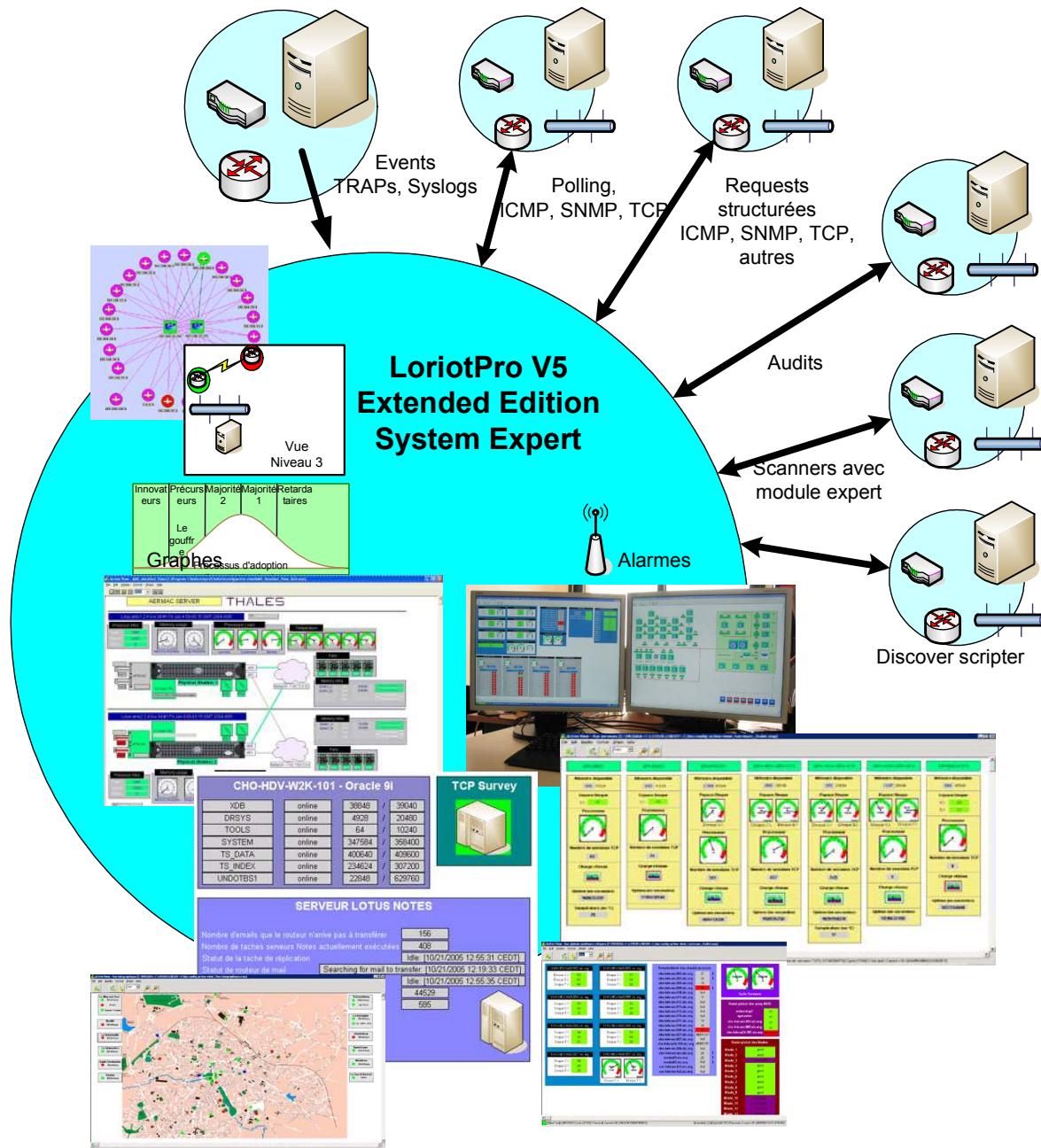
#### 4.2 Le Système Expert de LoriotPro V5

Le système expert de LoriotPro V5 « Extented Edition » est composé de modules interdépendants ayant la capacité à travers l'utilisation de scripts Open Sources d'exploiter les ressources du logiciel et les résultats des différentes collectes réalisées sur les équipements de l'infrastructure communicante. Tous les modules composant le système expert disposent d'un accès aux paramètres du noyau du logiciel et peuvent jouer des scripts.

#### 4.2.1 Composant principaux du Système Expert

- Un moteur d'interprétation de script LUA V5.1.3
- Un environnement de création de script graphique évolué
- Un module d'audit multitâche
- Un module IP Expert pour la découverte et la configuration
- Des MAPs graphiques entièrement paramétrables (Activeview)
- Des modules de graphes multi courbes « RRD Collector / manager»
- Un module de configuration de masse « bulk configuration »
- Tous les modules et fonctions de LoriotPro de type NMS peuvent utiliser des objets SNMP virtuels scriptés.
- Le report Center permet la gestion graphique et ergonomique des scripts et jobs.
- Un module IPSCANNER pour la découverte et la configuration
- Un module associé à la réception d'événets ou de traps permettant de réaliser des scripts.

Le schéma suivant représente symboliquement les capacités offertes par les différents modules du système expert.



**Attention** 💣

Seule l'« Extended Edition » de la version 5 du logiciel offre des fonctions de scripting indispensables pour des opérations d'audits, d'expertise et d'auto configuration avancées.

#### 4.3 Présentation des modules formant le système Expert

Il s'agit ici simplement d'une présentation rapide des éléments du système expert de LoriotPro, chaque élément fera l'objet d'un chapitre détaillé.

#### 4.3.1 Interpréteur LUA

LoriotPro intègre un moteur d'interprétation de script LUA en version V5.1.3 et implémente plusieurs bibliothèques spécialement écrites pour réaliser les tâches attendues.

LUA est un langage de scripts encapsulés dans LoriotPro depuis la version 4 « Extended Edition ». LUA existe depuis 1993 et fait l'objet d'un nombre important de communications ; il a été choisi pour LoriotPro car il permet de façon simple de créer de nouvelles fonctions (mots de langage). Il est de type interprété ce qui simplifie la réalisation de scripts car aucune phase de compilation n'est nécessaire pour mettre en œuvre une solution.

##### *Where does Lua come from?*

Lua is designed and implemented by a [team](#) at [PUC-Rio](#), the Pontifical Catholic University of Rio de Janeiro in Brazil. Lua was born and raised at [Tecgraf](#), the Computer Graphics Technology Group of PUC-Rio, and is now housed at [Lablua](#). Both Tecgraf and Lablua are laboratories of the [Department of Computer Science](#) of PUC-Rio.

Les librairies de fonctions spécifiques au management de réseaux implémentées par LUTEUS contiennent plus de 200 fonctions avancées réparties dans les domaines de la communication et de la gestion graphique des objets d'ActiveView. Le logiciel répond à la norme des modules LUA 5.1 et vous pouvez trouver très facilement sur Internet des modules de librairies complémentaires.

- Un sdk est disponible pour ajouter des fonctions de bas niveau aux librairies existantes (dll C++)
- Depuis la première implémentation de LUA dans la version 4 de LoriotPro, le nombre de fonctions a doublé pour répondre aux besoins des administrateurs.

##### 4.3.1.1 Plus d'information sur LUA

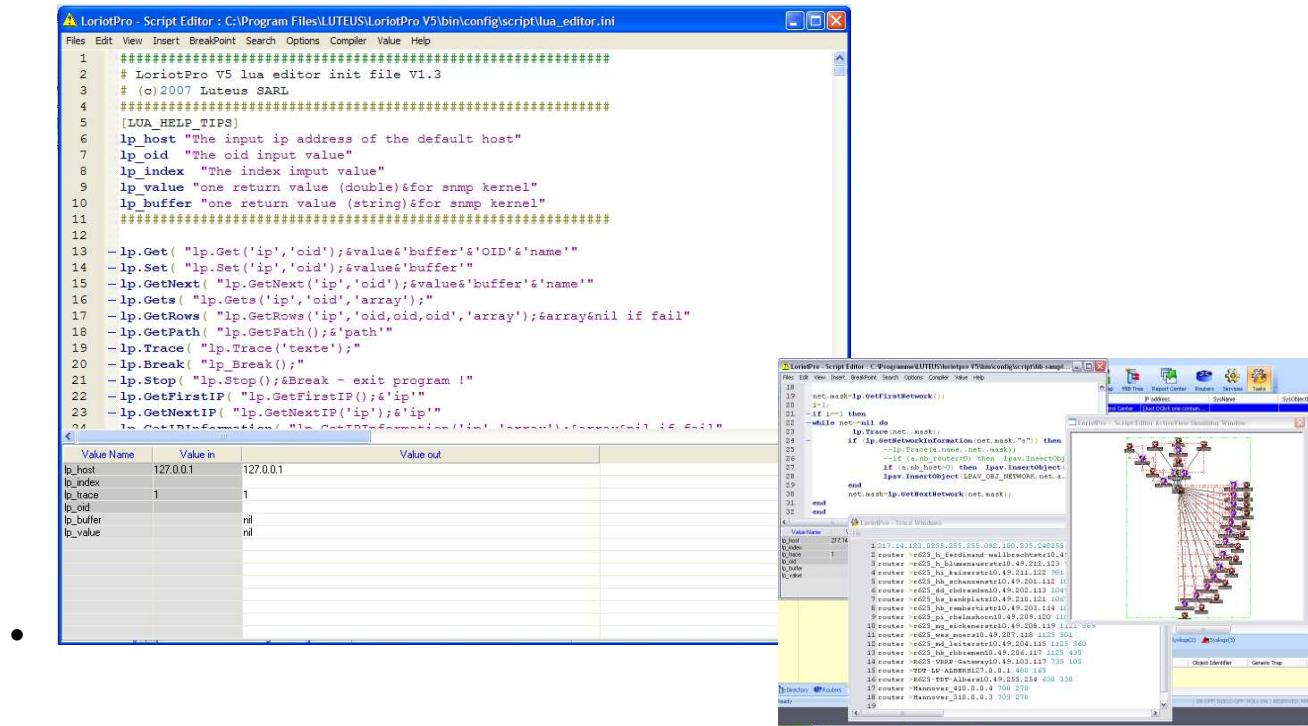
- <http://www.lua.org/about.html>
- <http://www.lua.org/news.html>
- <http://luaforge.net/>

### 4.3.2 L'environnement de développement intégré

LoriotPro intègre un environnement graphique évolué de création de scripts, constitué d'un éditeur associé à un ensemble de modules d'aide à la création (Wizard). L'éditeur de scripts intégrés répond aux normes classiques des environnements modernes avec un ensemble de wizards permettant la création et la maintenance des scripts dans les meilleures conditions.

Quelques unes des fonctions de l'environnement de développement :

- Editeur graphique et contextuel avec coloration des sources.
- Interpréteur avec une gestion des erreurs de codage.
- Editeur graphique avec des aides contextuelles (Wizards) pour faciliter la réalisation des scripts.
- Fenêtre de simulation d'ActiveView
- Break Point, pour le debugging.
- Complession, aide à la syntaxe etc...
- ...



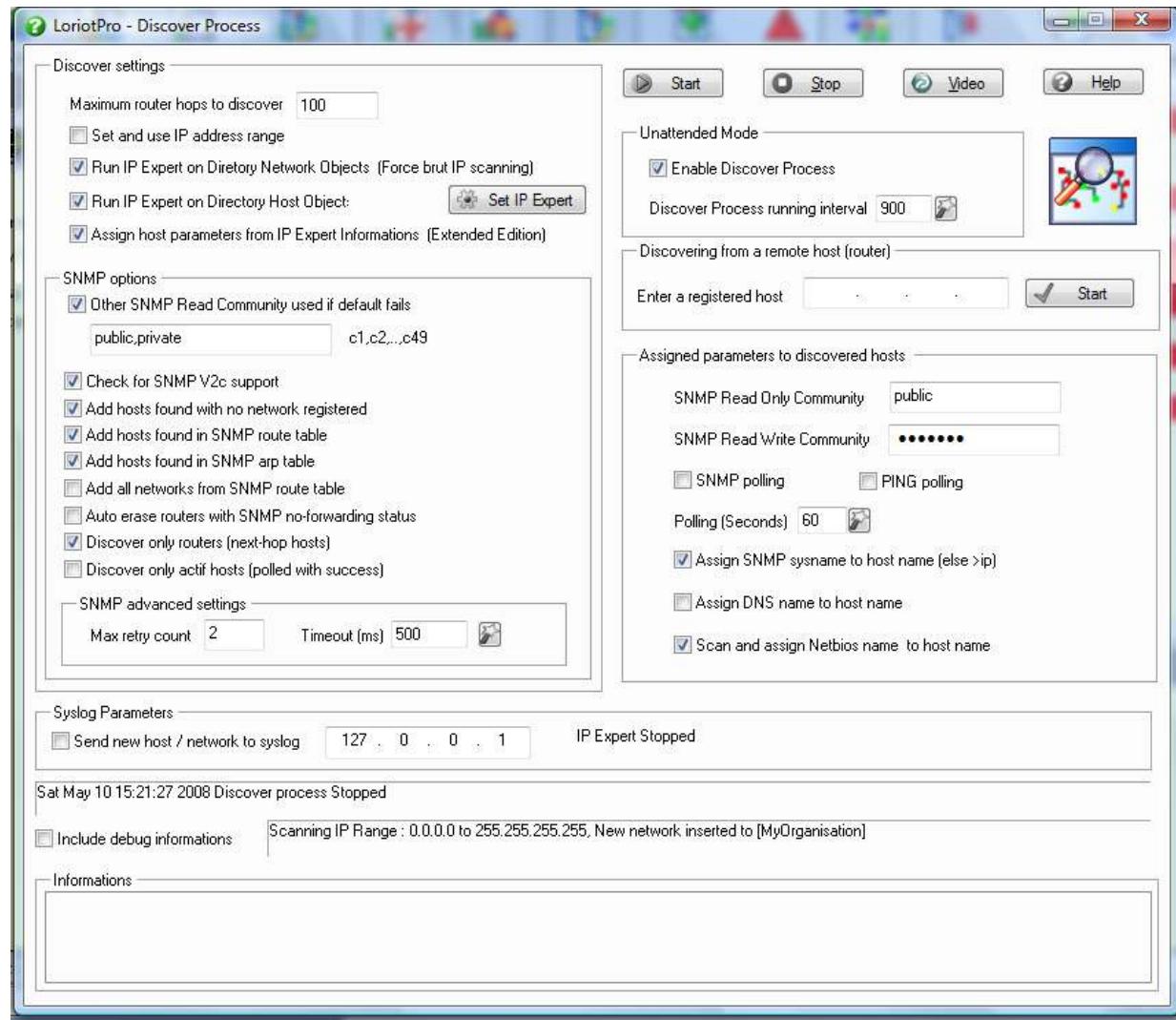
### 4.3.3 Le module IP Expert

LoriotPro intègre un module IP Expert pour la découverte et la configuration des équipements de la Directory. Ce module intègre des stratégies de découverte et d'auto configuration (script) paramétrable. Il peut y avoir des stratégies par network ou par équipement. Ce module intégré sous forme d'un module noyau du logiciel est contrôlé par le module Discover déjà présent dans la version 4 et les précédentes. Le module Discover réalise une découverte furtive du réseau via un algorithme de requêtes SNMP. Il fournit ensuite au module IP Expert les informations des réseaux et

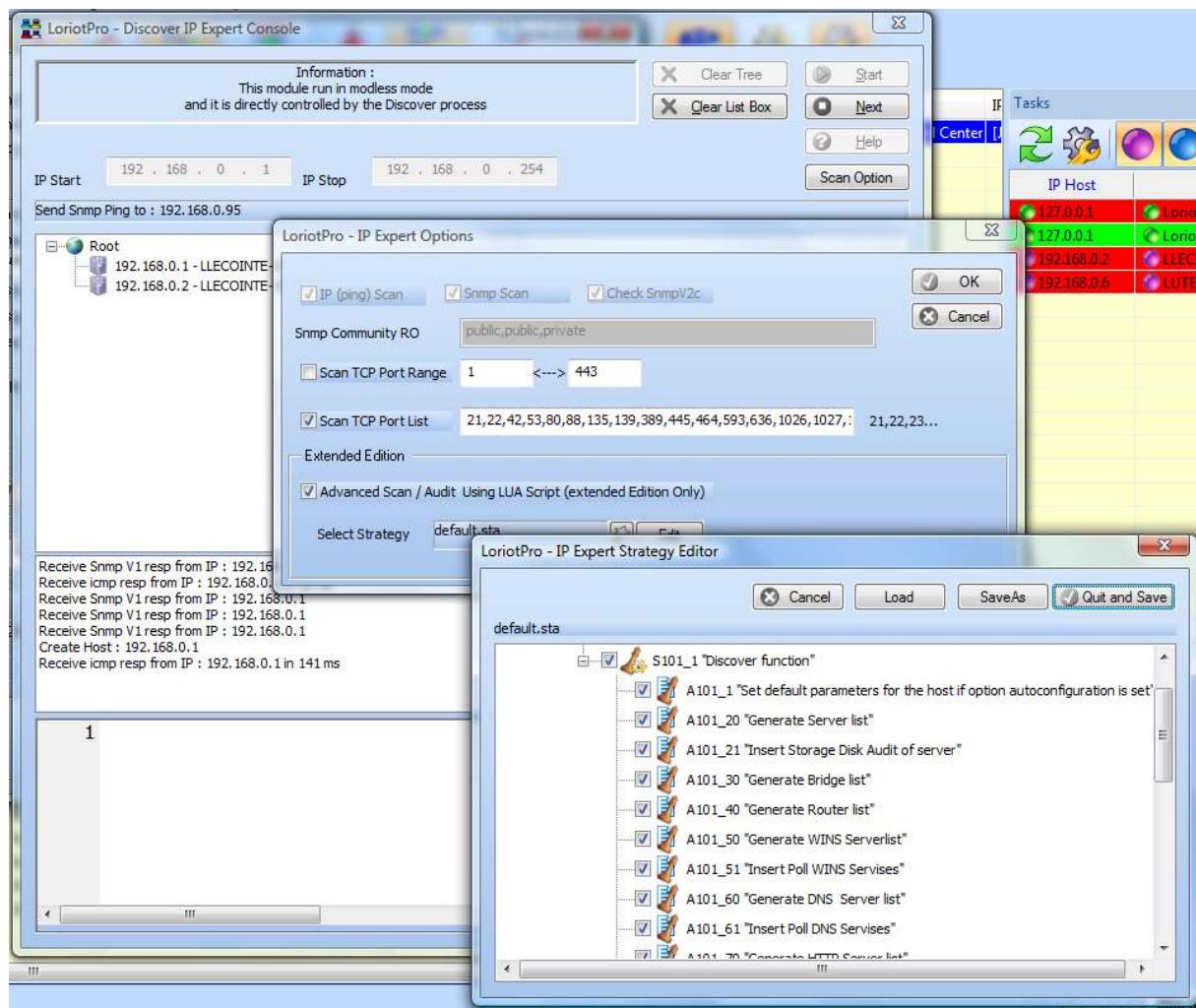
## LoriotPro V5.00 Extended Edition – Supervision et gestion SNMP par scripts LUA

### Introduction au System Expert et aux scripts

équipements détectés. Le module IP Expert réalise alors un scan approfondi des ressources découvertes de façon scriptée en suivant une stratégie configurable. Les scripts de scan du module IP Expert sont architecturés et il est possible d'augmenter leur nombre en utilisant l'environnement de développement et ses Wizards.



La fenêtre de paramétrage du module Discover.

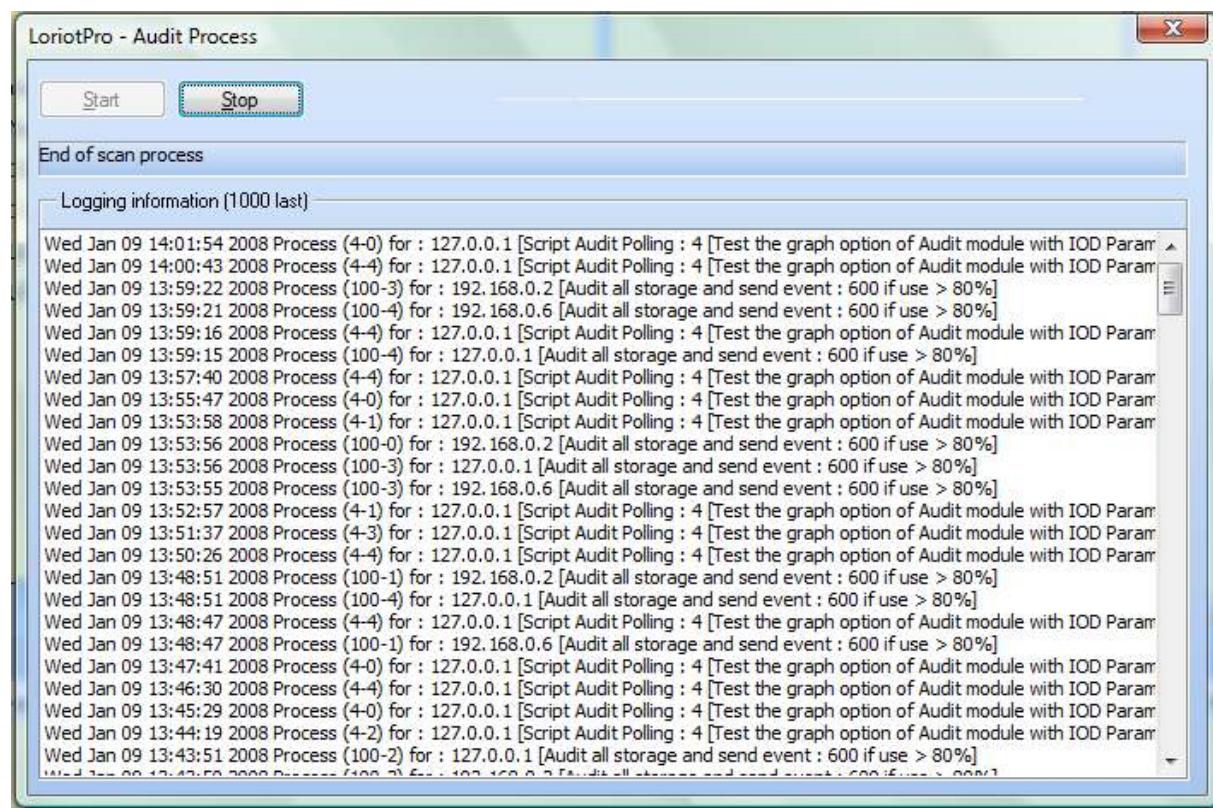


Les fenêtres de paramétrage du module IP Expert.

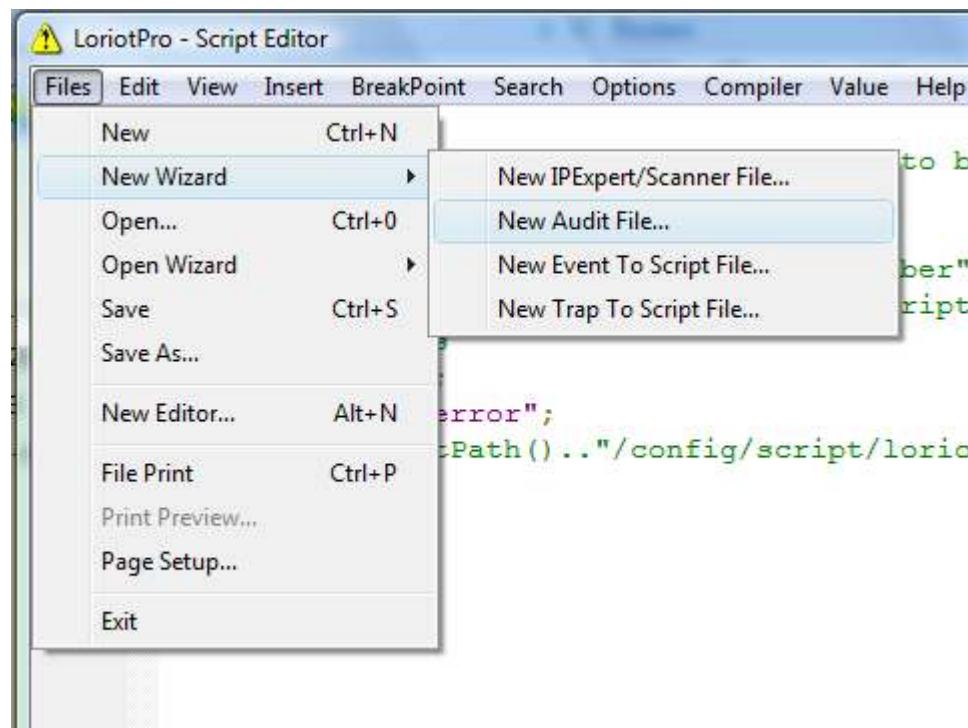
#### 4.3.4 Le module d'audit multi-tâche

Ce module intégré au noyau de LoritoPro V5 fait tourner des scripts d'audits attachés aux équipements de la Directory. Il est multitâche et permet d'auditer de façon rapide un environnement communicant.

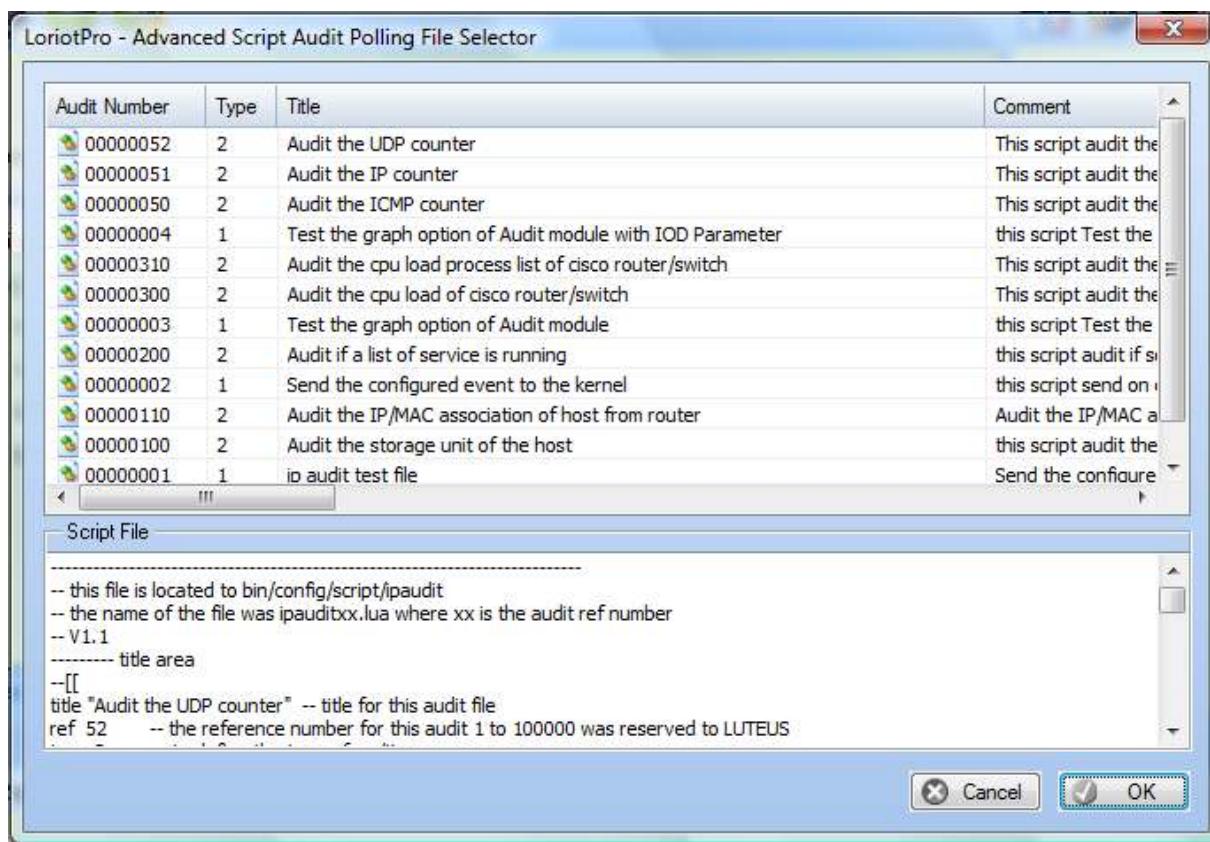
Les scripts d'audits peuvent être auto positionnés et/ou configurés par un autre script (IP Expert, IP SCANNER ou autre). On peut faire exécuter à intervalles réguliers tout type de traitements, (configurations, alarmes, audits, expertises...) . Il est possible de réaliser une génération de SLA sur la durée d'exécution du script et de grapher une information générée par le script d'audit associé à un équipement de la Directory. Les scripts d'audits sont architecturés et il possible et aisément d'augmenter leur nombre en utilisant les Wizards de l'environnement de développement.



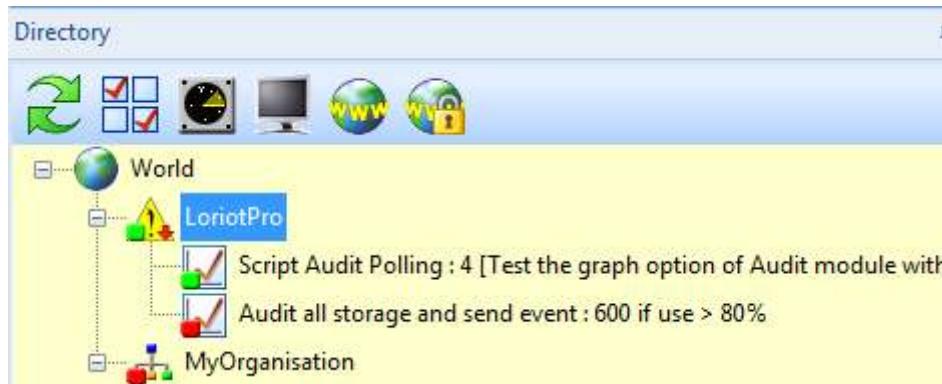
Le module de scheduling des scripts d'audits.



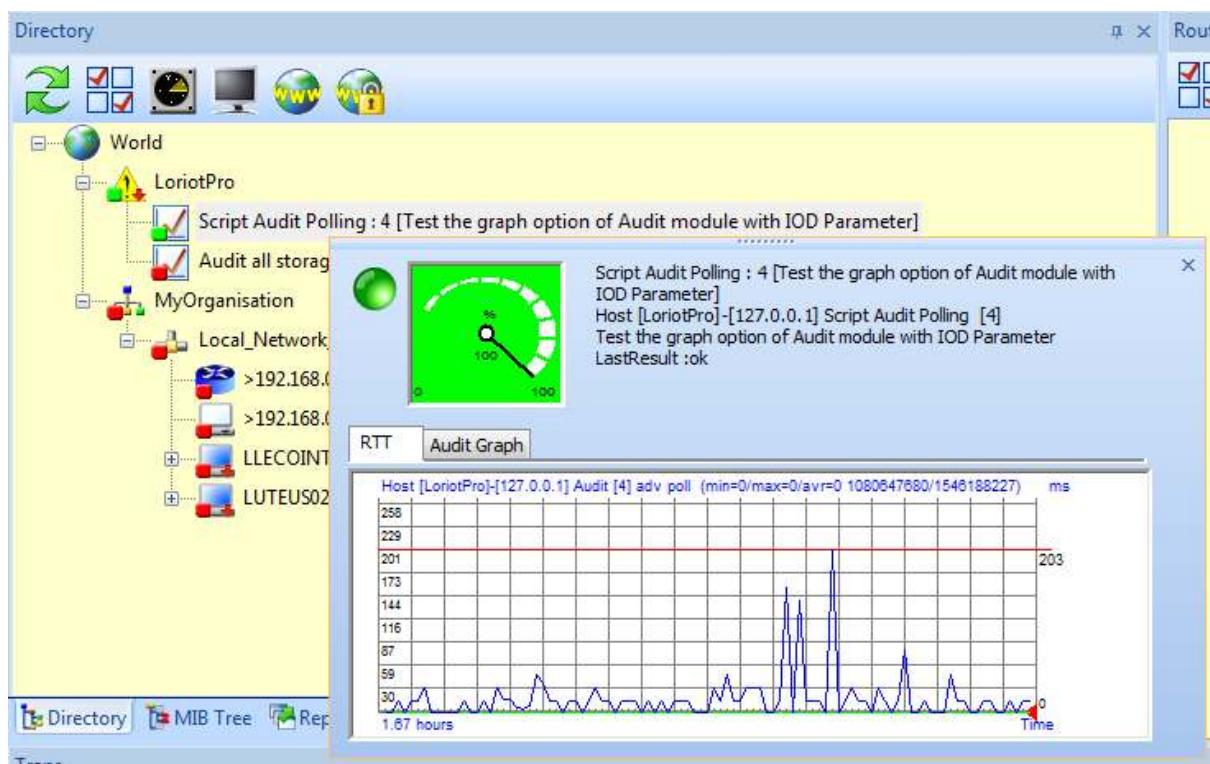
Le Wizard de création d'un script d'audit dans l'éditeur.



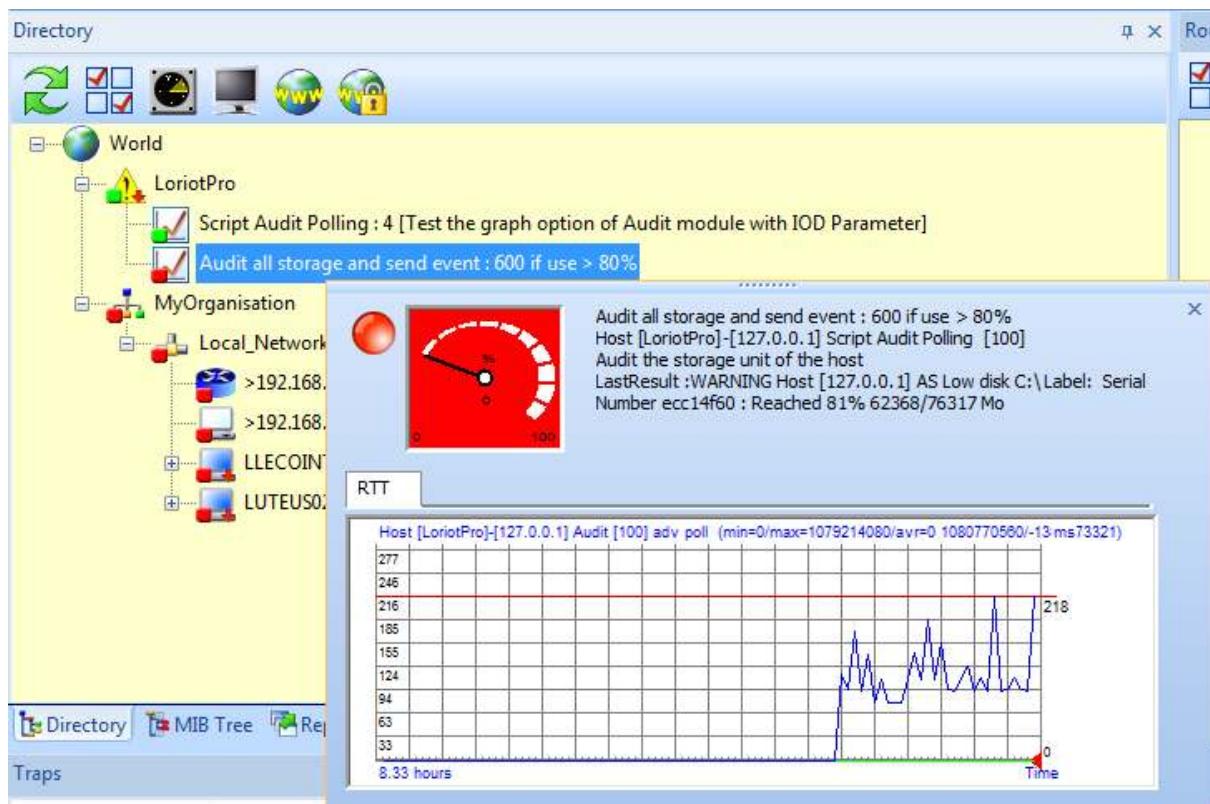
La fenêtre de sélection des scripts d'audits.



Deux scripts d'audits insérés sous un équipement dans la Directory



Une fenêtre volante associée à un script d'audit.

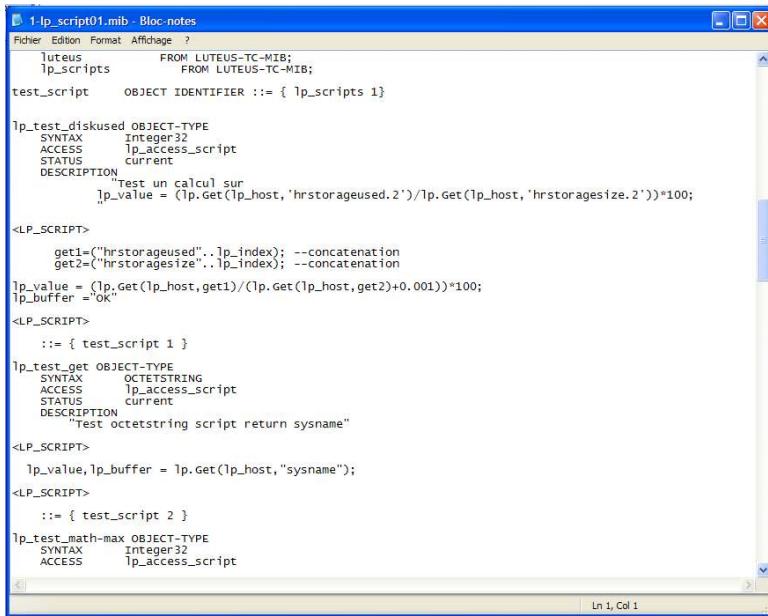


Une autre fenêtre volante associée à un script d'audit contrôlant la capacité d'un disque dur d'un équipement.

#### 4.3.5 Objets SNMP Virtuel

LoriotPro intègre la possibilité de créer des MIBs dont les objets seront des scripts. Tous les modules et fonctions de LoriotPro peuvent utiliser des objets SNMP virtuels.

Les objets virtuels SNMP (scripts) pourront être testés, graphés, affichés comme des objets SNMP classiques. Le cas du calcul de pourcentage d'occupation d'un disque dur pourra être réalisé par un objet SNMP virtuel.



```
1-lp_script01.mib - Bloc-notes
Fichier Edition Format Affichage ?
luteus          FROM LUTEUS-TC-MIB;
lp_scripts       FROM LUTEUS-TC-MIB;
test_script     OBJECT IDENTIFIER ::= { lp_scripts 1}

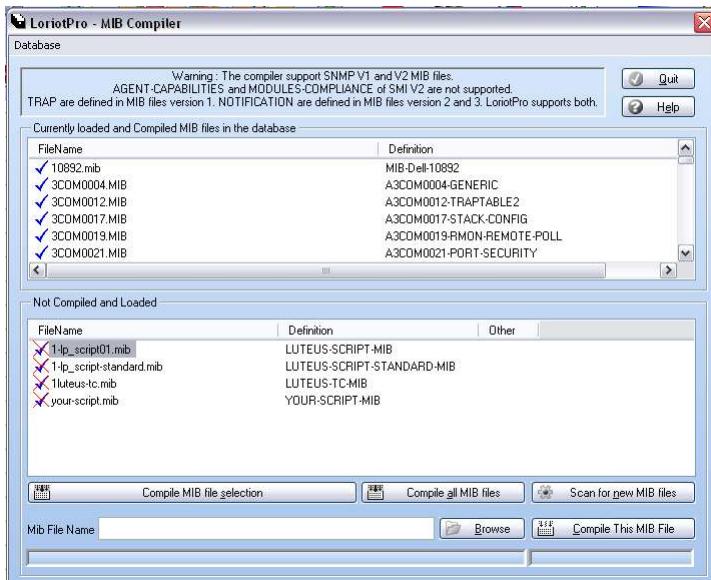
lp_test_discused OBJECT-TYPE
SYNTAX      Integer32
ACCESS     lp_access_script
STATUS      current
DESCRIPTION "test un calcul sur
lp_value = (lp.Get(lp_host, 'hrstorageused.2')/lp.Get(lp_host, 'hrstoragesize.2'))*100;

<LP_SCRIPT>
get1=("hrstorageused"..lp_index); --concatenation
get2=("hrstoragesize"..lp_index); --concatenation
lp_value = (lp.Get(lp_host,get1)/(lp.Get(lp_host,get2)+0.001))*100;
lp_buffer = "OK"
<LP_SCRIPT>
 ::= { test_script 1 }

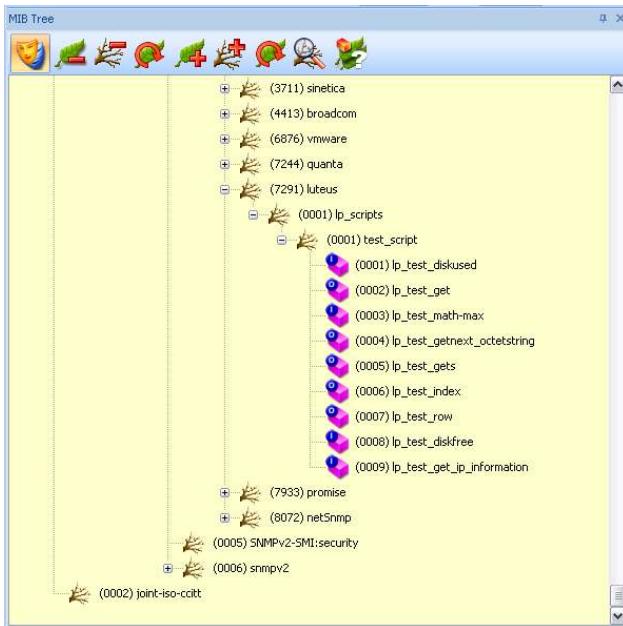
lp_test_get OBJECT-TYPE
SYNTAX      OCTETSTRING
ACCESS     lp_access_script
STATUS      current
DESCRIPTION "test octetstring script return sysname"
<LP_SCRIPT>
lp_value,lp_buffer = lp.Get(lp_host, "sysname");
<LP_SCRIPT>
 ::= { test_script 2 }

lp_test_math-max OBJECT-TYPE
SYNTAX      Integer32
ACCESS     lp_access_script
```

Le fichier de MIB de l'objet SNMP virtuel



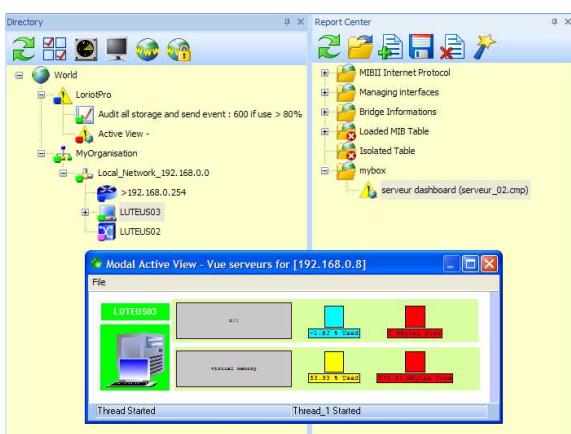
Compilation du fichier de MIB



Exploitation des objets SNMP virtuels par les différents modules du logiciel de façon transparente.

#### 4.3.6 Les ActiveView

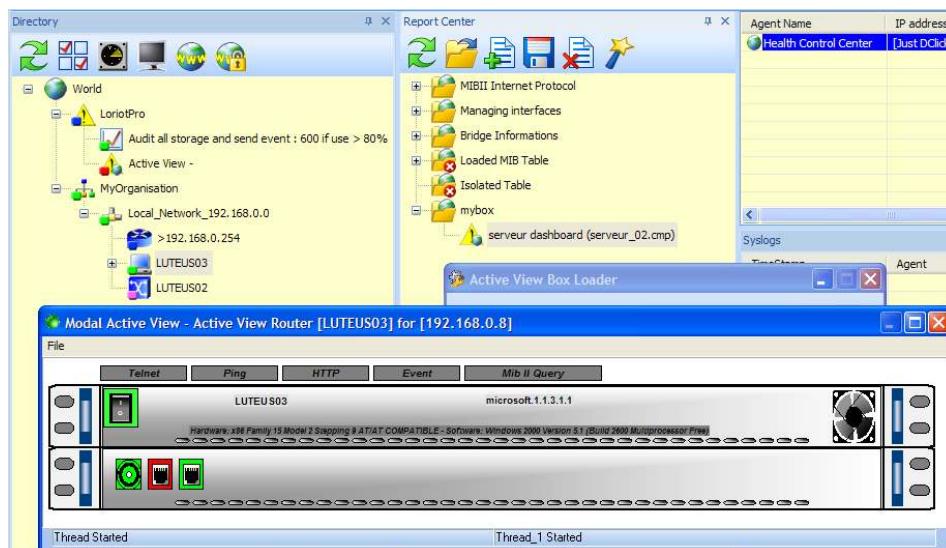
Une ActiveView est un espace graphique entièrement paramétrable permettant l'affichage d'objets vectoriels en relation avec des statuts ou informations collectés par le logiciel. Tous les objets d'une ActiveView sont contrôlables par script, chaque objet d'une ActiveView peut faire tourner un script. De même, un script peut générer une ActiveView modale et remplir sa « MAP » automatiquement. Chaque ActiveView intègre un puissant éditeur permettant la création aisée de ses représentations graphiques.



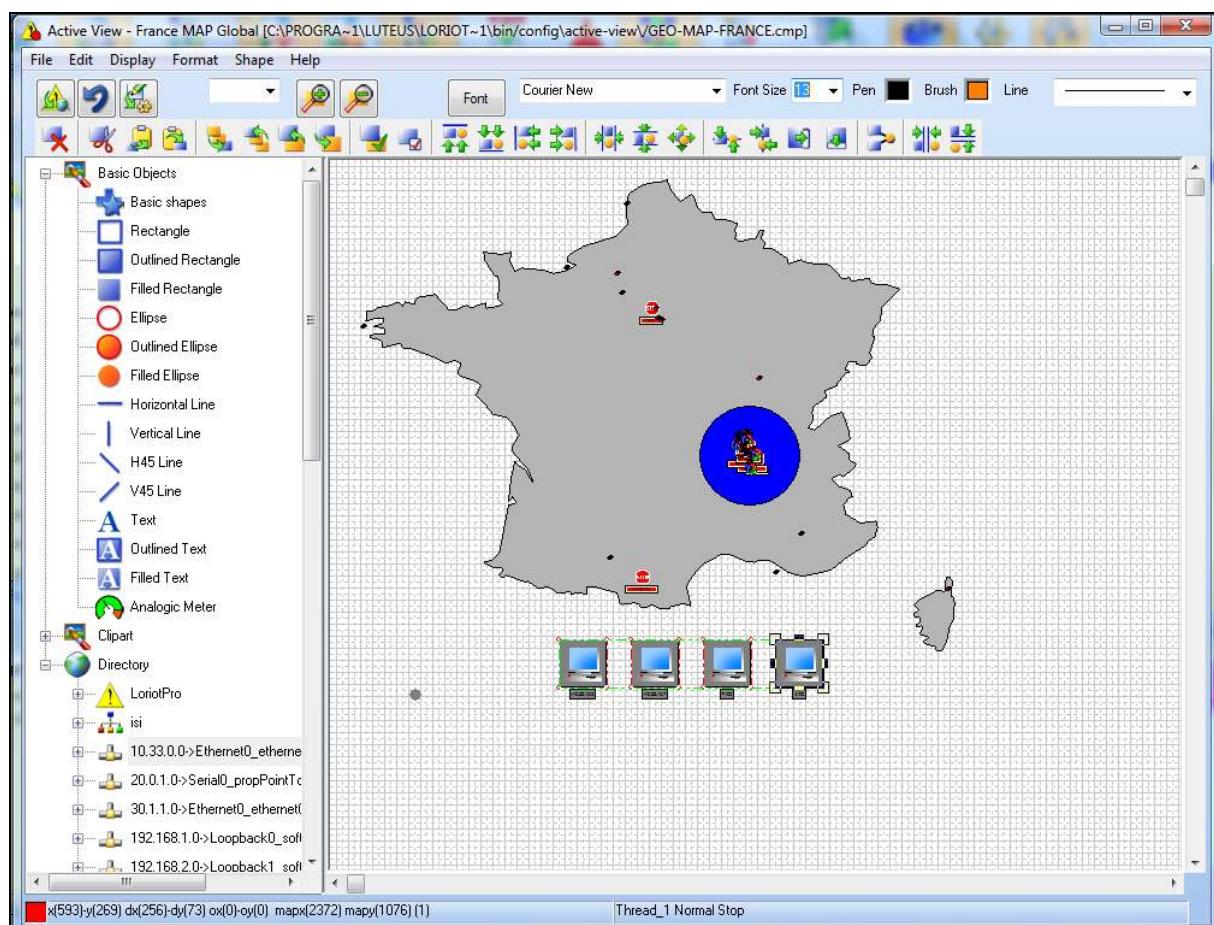
ActiveView Simple utilisée sous la forme d'une template.

# LoriotPro V5.00 Extended Edition – Supervision et gestion SNMP par scripts LUA

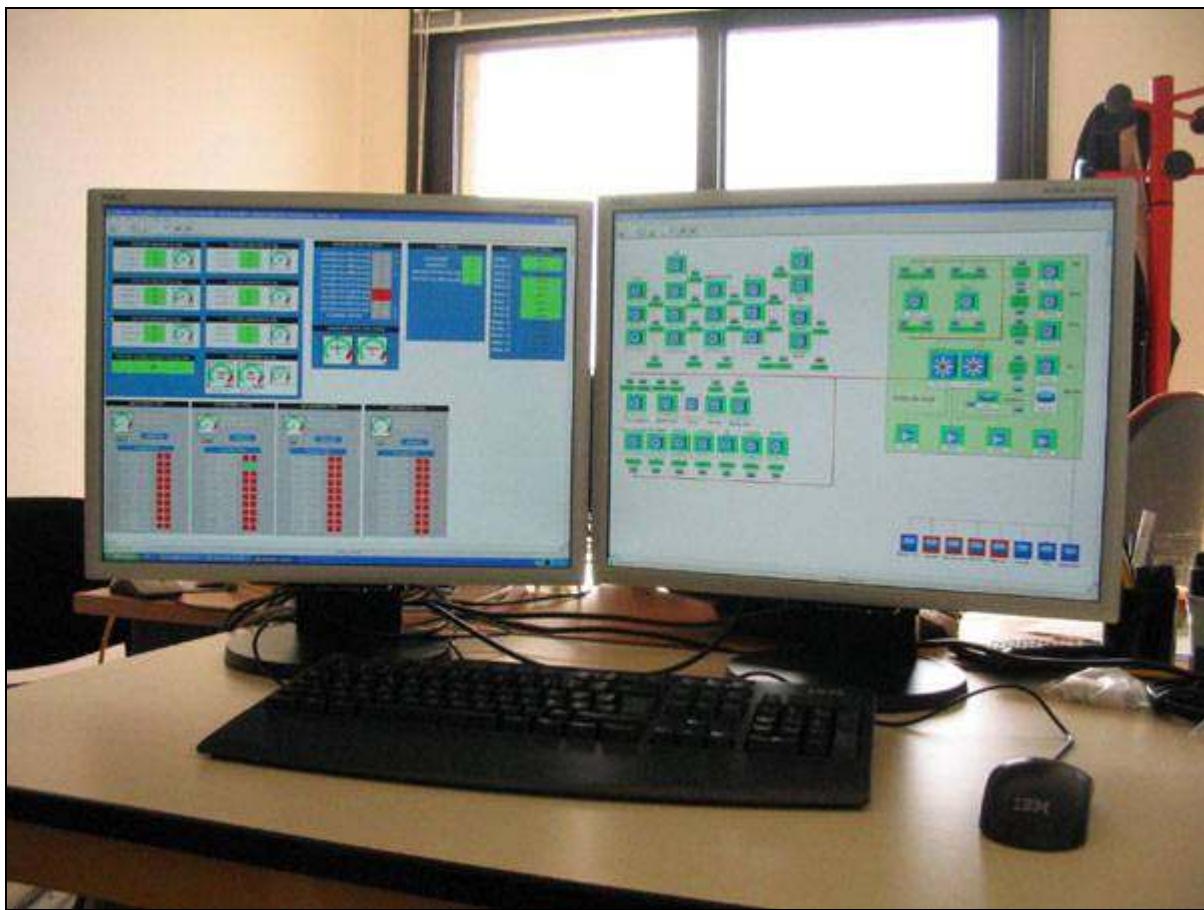
## Introduction au System Expert et aux scripts



ActiveView générée dynamiquement pour visualiser une face avant d'équipement.



L'éditeur d'ActiveView.



Des ActiveViews complexes en mode double écran ( 1 seul LoriotPro bien sur).



Un exemple de Dashboard réalisé avec une ActiveView.

**CHO-HDV-W2K-101 - Oracle 9i**

XDB	online	38948	/	39040
DRSYS	online	4928	/	20480
TOOLS	online	64	/	10240
SYSTEM	online	347584	/	358400
TS_DATA	online	400640	/	409600
TS_INDEX	online	234624	/	307200
UNDOTBS1	online	22648	/	629760

**TCP Survey**

**SERVEUR LOTUS NOTES**

Nombre d'emails que le routeur n'arrive pas à transférer : 156

Nombre de tâches serveurs Notes actuellement exécutées : 408

Statut de la tâche de réplication : Idle [10/21/2005 12:55:31 CEDT]

Statut de routeur de mail : Searching for mail to transfer [10/21/2005 12:19:33 CEDT]

Status de la tâche événement : Idle [10/21/2005 12:55:35 CEDT]

Calendar - Nombre total de rdy et ressources : 44529

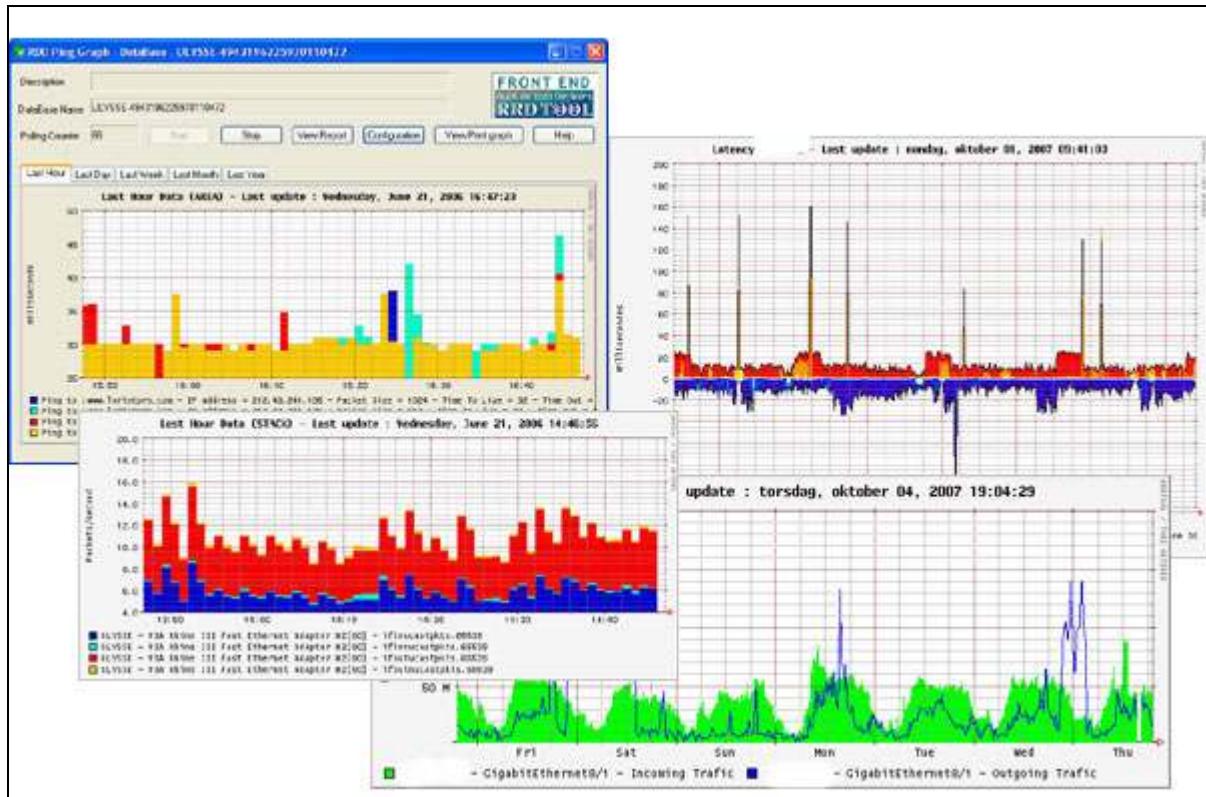
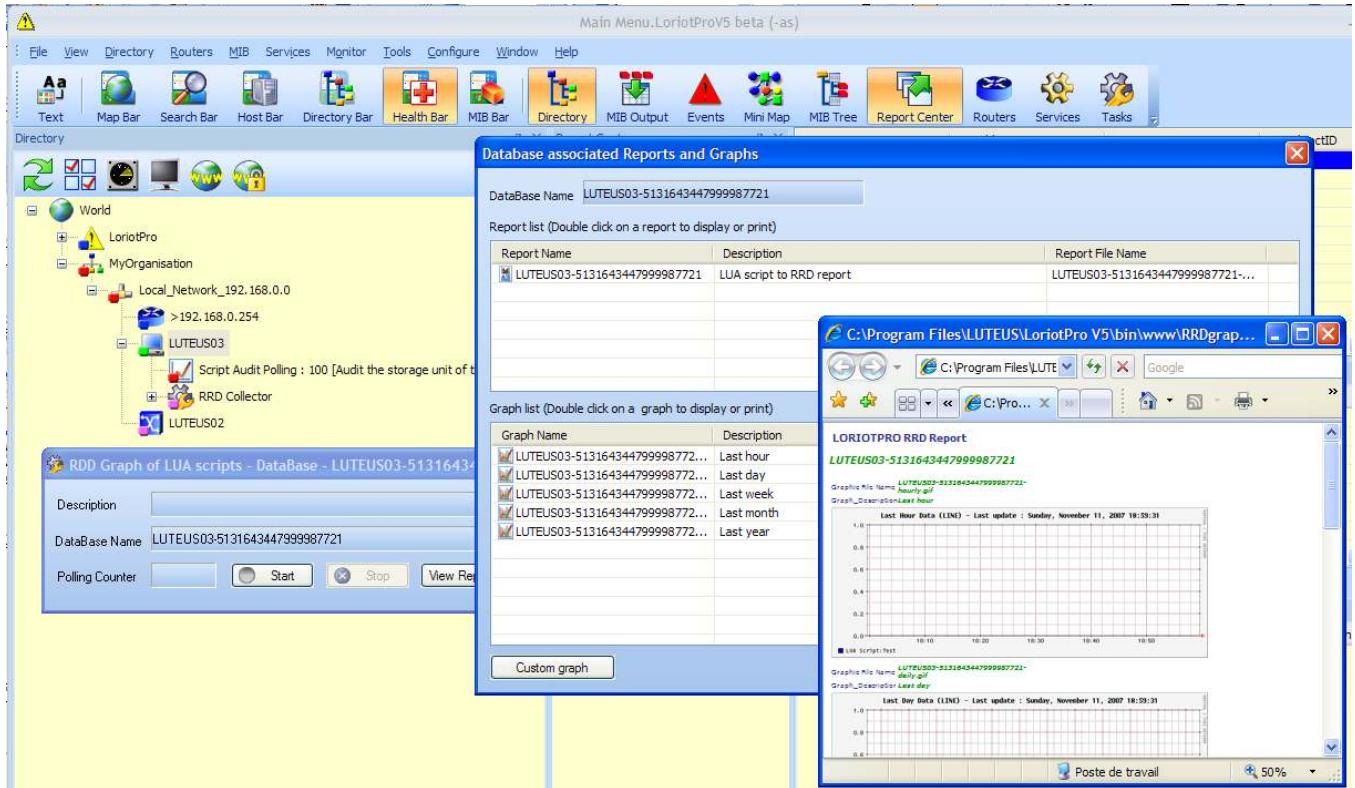
Calendar - Nombre total d'utilisateurs : 595

Des informations de niveau 7 affichées avec une ActiveView.

#### 4.3.7 Graphe et courbe

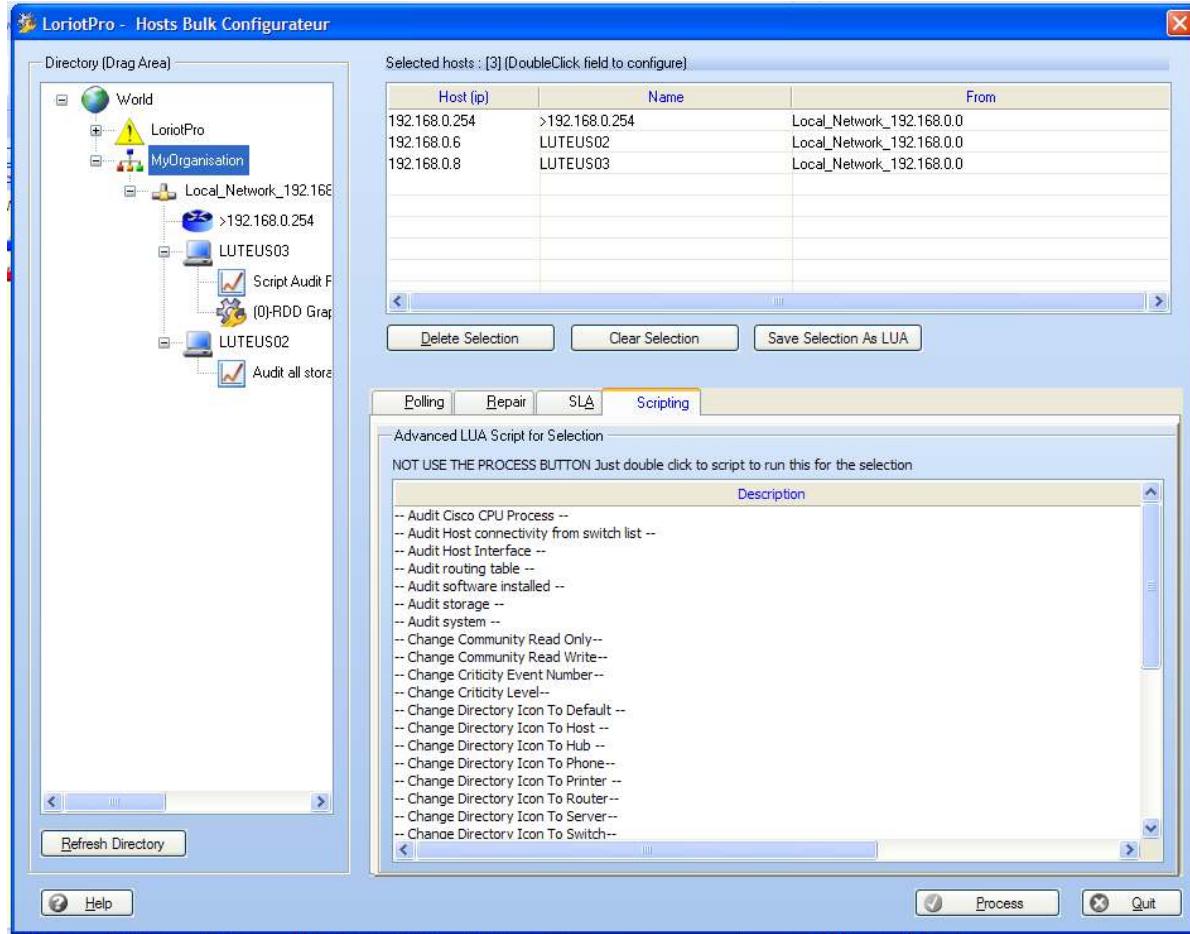
LoriotPro intègre les modules de graphes multi courbes « RRD Collector / Manager ». Le plugin « RRD Collector » peut utiliser un script pour réaliser ses graphes ; le plugin de service « RRD Manager » permet de mettre en forme plusieurs graphes collectés pour réaliser des rapports. On peut grapher tout type d'information, ping, objet SNMP, (SLA de SLA), corrélation de données.

## LoriotPro V5.00 Extended Edition – Supervision et gestion SNMP par scripts LUA Introduction au System Expert et aux scripts



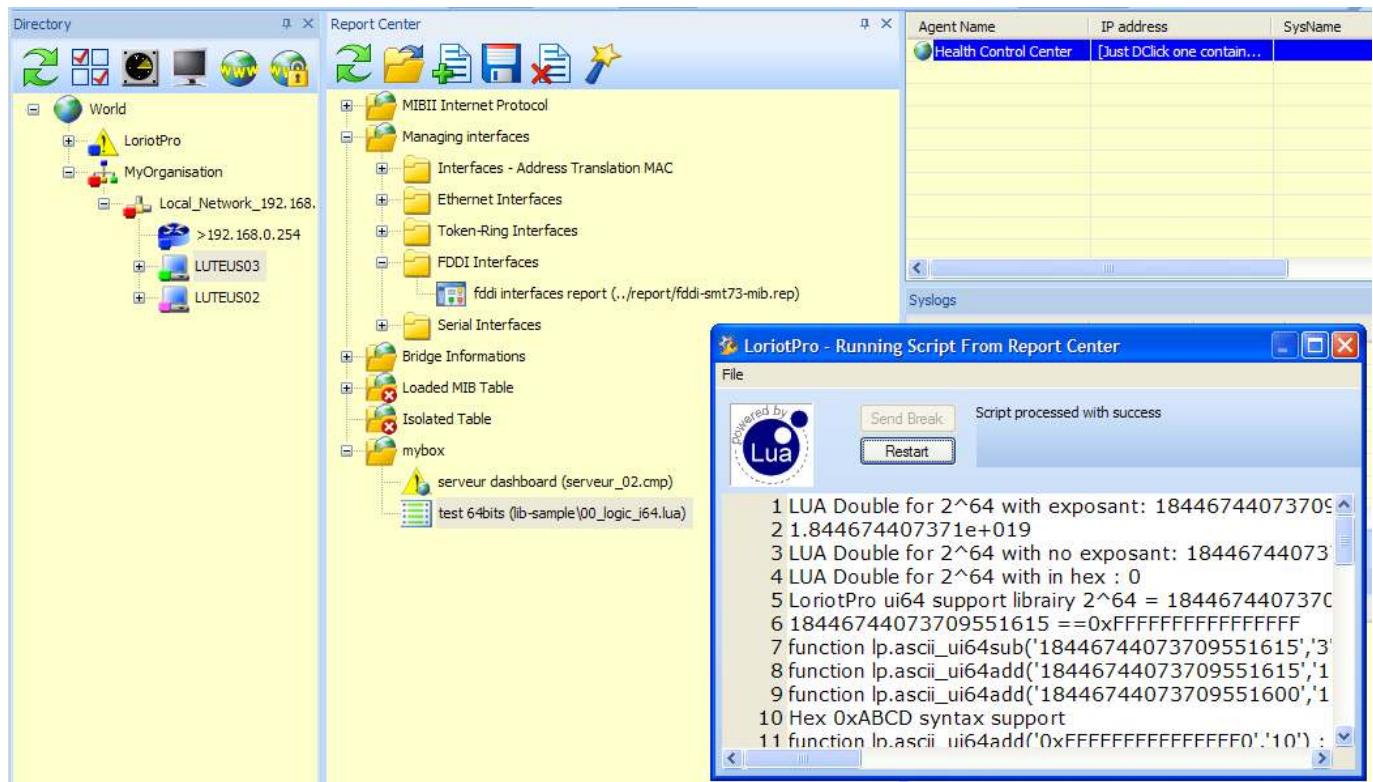
### 4.3.8 Le module de configuration multiple « bulk configuration »

Il est possible de générer des scripts de reconfigurations, d'audits, d'expertises qui seront appliqués à une sélection d'équipements.



#### 4.3.9 Le Report Center

Le report Center permet la gestion graphique et ergonomique des scripts (audits, expertises, tâches...). Les différents scripts réalisés pourront être mis en valeur avec ce module. L'ensemble des tâches et requêtes configurées pourra être structuré avec ce module.



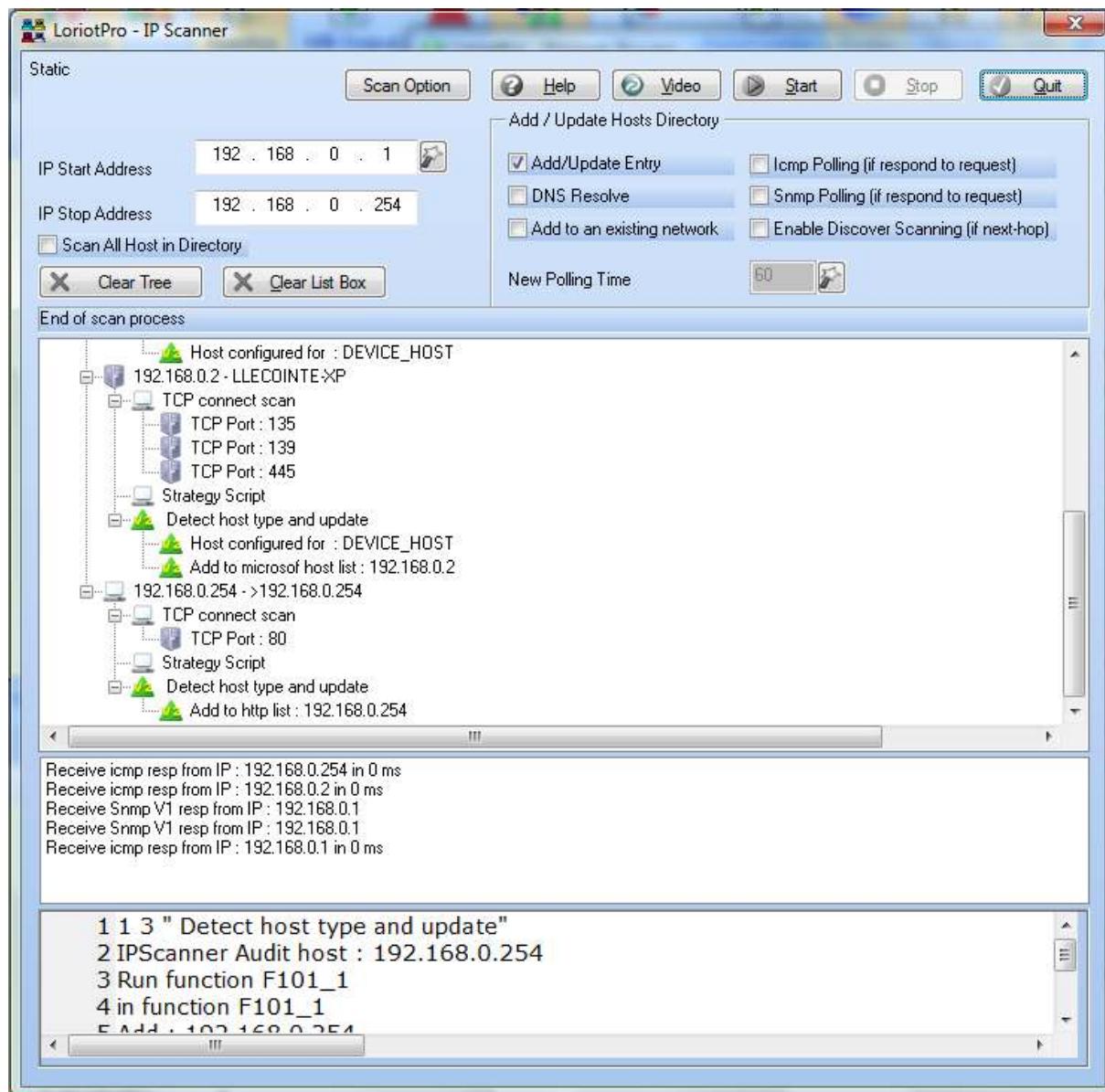
#### 4.3.10 Le Module IP Scanner

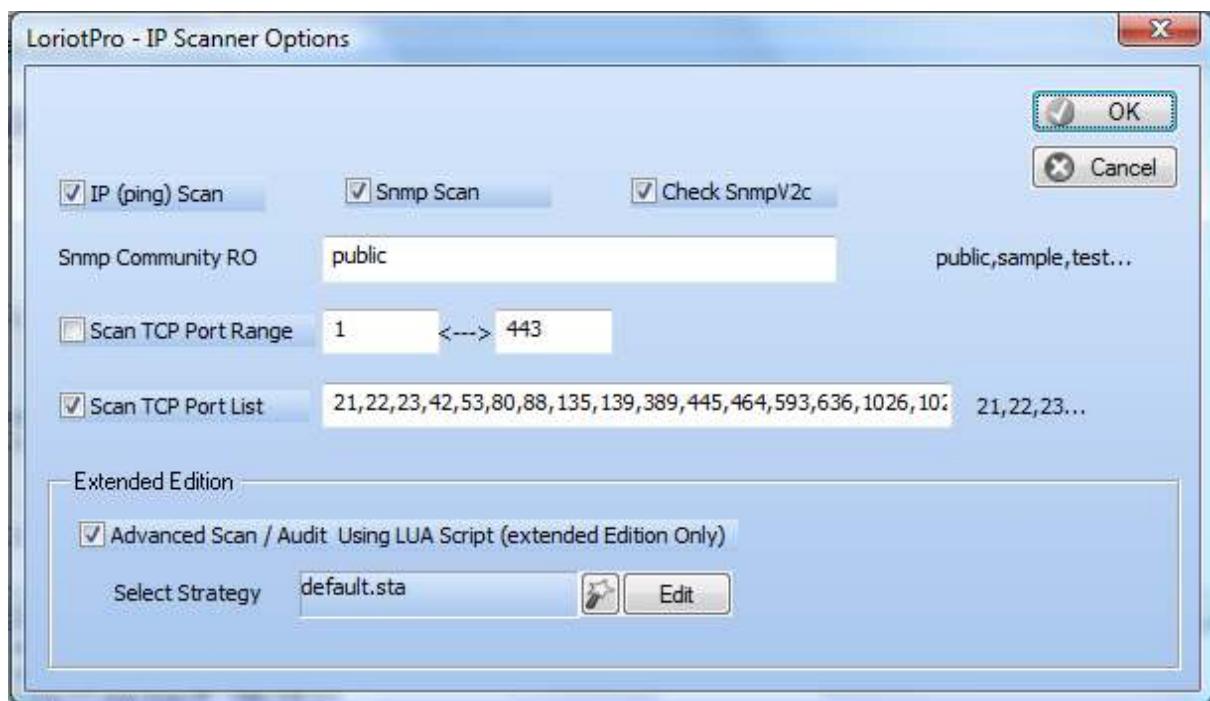
LoriotPro intègre le module IP Scanner pour la découverte et la configuration des équipements de la Directory. Ce moteur fait tourner des stratégies de découverte et d'auto configuration (script). Il peut y avoir des stratégies par network ou par équipement. Le module IP Scanner permet la réalisation d'un scan approfondi des ressources sélectionnées. Il est possible de faire jouer des scripts de scans en suivant une stratégie programmable.

Les scripts de scans du module IP Scanner sont architecturés et il possible d'augmenter leur nombre en utilisant l'environnement de développement et ses Wizards.

# LoriotPro V5.00 Extended Edition – Supervision et gestion SNMP par scripts LUA

## Introduction au System Expert et aux scripts





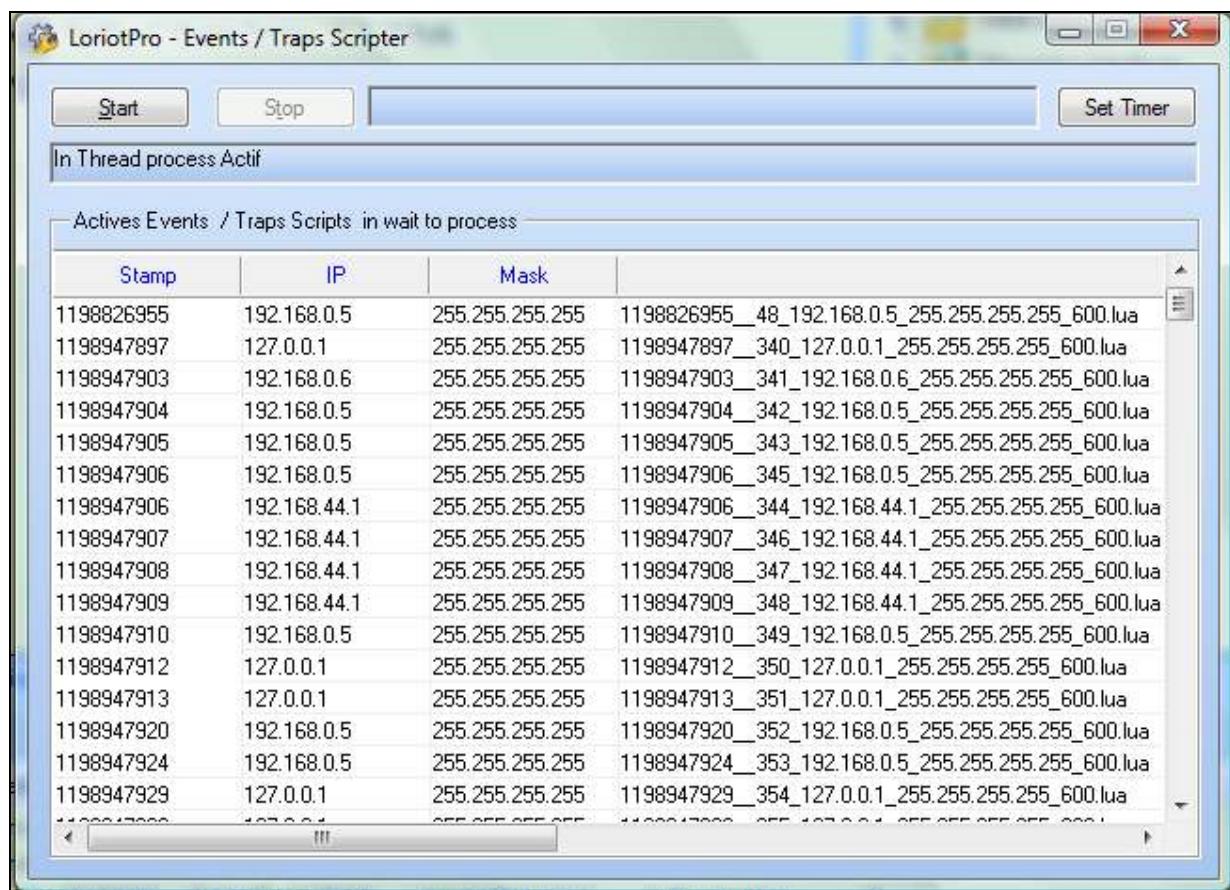
Quelques paramètres de scans possibles.



L'éditeur de stratégies pour les scans scriptés.

#### 4.3.11 Les Filtres d'Events et de Traps scriptés

Il est possible de déclencher un script d'analyse sur réception d'un Event LoriotPro ou un Trap SNMP. Le module « Event / Trap Scripter» aura accès à l'ensemble des paramètres de l'événement.

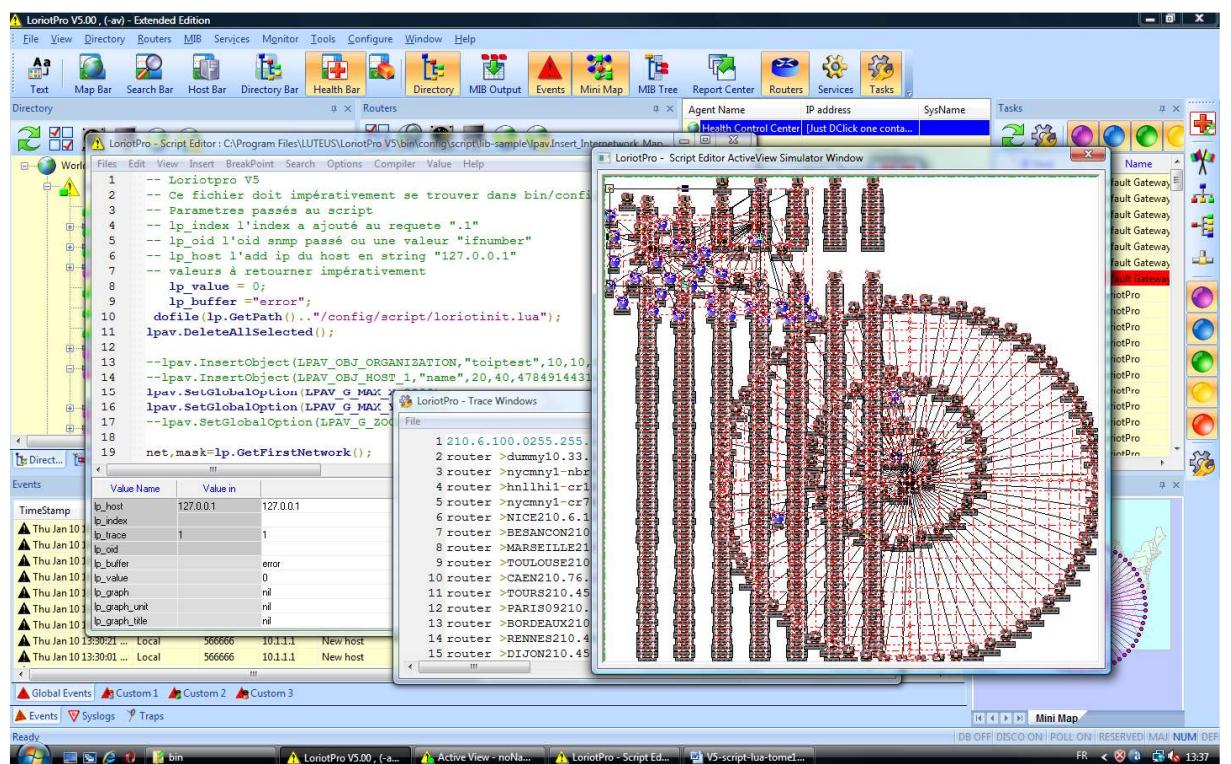


## 5 Le Langage LUA

Le langage LUA est de type interprété, c'est-à-dire qu'un programme LUA peut être écrit à partir d'un simple éditeur de texte et exécuté sans passer par l'usage d'un compilateur. Le langage LUA utilise une syntaxe spécifique décrite dans la documentation officielle du langage que l'on peut trouver sur le site <http://www.lua.org>. L'intérêt de ce langage est son mode « embedded », c'est-à-dire qu'il s'intègre à une application, LoriotPro en l'occurrence, pour étendre ses fonctionnalités. LUA permet dans son mode « embedded » la création de nouveaux mots de langage qui seront spécifiques à l'application. L'intégration de LUA à LoriotPro nous a permis de créer un ensemble de nouvelles fonctions LUA spécifiques à LoriotPro permettant de résoudre les besoins de collecte et de traitement intelligent.

### 5.1 Intégration de LUA dans LoriotPro

LoriotPro « Extended Edition » prend en charge des extensions de fonctionnalité écrites en LUA. Ces extensions se présentent sous la forme de scripts au format texte, situés dans le répertoire **\loriotpro-rep\bin\config\script**. Ces scripts sont utilisables sous la forme d'objets SNMP virtuels ou bien directement à partir d'une ActiveView, du « Report Center », de « ShortCut » d'équipements, ou des différents menus ou modules du logiciel. Un environnement de développement est fourni pour permettre de créer, tester et lancer des scripts de tous types. En fonction du mode d'utilisation des scripts, il existe plusieurs librairies de fonctions spécifiques. LoriotPro V5 « Extended Edition » utilise la version 5.1.x de LUA en mode dll (luas5.1.dll) ce qui permet l'utilisation de librairies externes. La suite du document s'applique à montrer comment utiliser ces fonctions et librairies.



L'environnement de développement intégré à LoriotPro en action.

L'ensemble des scripts LUA de LoriotPro « Extended Edition » utilise les mêmes fonctions de librairie mais ont des structures légèrement différentes en fonction de leur module d'utilisation. Ceci est lié au fait du passage de variables entre le module lançant le script et le script lui-même. Par exemple un script lancé par une ActiveView peut contenir des fonctions de gestion des objets graphiques qui n'ont pas d'existence dans un script lancé par le module « IP Expert ». Les Scripts lancés par le service « Event / Trap Scripter » contiennent les variables du Trap ou de l'Event généré par LoriotPro. Un Script d'audit contient des références de titres et de fonctions etc... Un type de script particulier est le script d'objets SNMP virtuels qui est intégré dans un fichier de MIB. La suite du document a pour objet de décrire les différents types de fichiers de scripts supportés par LoriotPro.

### 5.1.1 Variables globales

Le principe de fonctionnement des scripts dans LoriotPro est basé sur une logique de passage de paramètres entre le noyau du logiciel et l'interpréteur LUA intégré.

Quel que soit le module utilisant l'interpréteur LUA, la logique d'exécution du script est toujours la même.

#### 5.1.1.1 Variables en entrée du script

Le noyau lance un script en lui fournissant trois variables globales que vous pourrez utiliser dans votre script.

**Ip\_host** : c'est l'adresse IP sous la forme d'une chaîne de caractères de l'équipement par défaut.

**Ip\_index** : c'est une variable que vous avez configurée dans le module qui lance le script.

**Ip\_oid** : c'est une variable que vous avez configurée dans le module qui lance le script.

Le script reçoit aussi des pointeurs sur l'ensemble des ressources du noyau de LoriotPro, vous pouvez accéder à ces ressources en utilisant les librairies LoriotPro fournis avec l'interpréteur LUA et qui feront l'objet des chapitres suivants.

#### 5.1.1.2 Variables en sortie du script

En sortie, le script doit fournir deux variables au noyau du logiciel :

**Ip\_value** : c'est une valeur de type double qui servira au module qui a lancé le script pour réaliser son traitement.

**Attention** 

Le type double supporte difficilement les valeurs de type 64bits que l'on peut rencontrer en SNMP V2. La librairie **Ip** intègre un ensemble de fonctions spécifiques permettant de travailler avec ce type de variable 64bit unsigned.

**Ip\_buffer** : c'est une chaîne de caractères qui servira au module qui a lancé le script pour réaliser son traitement.

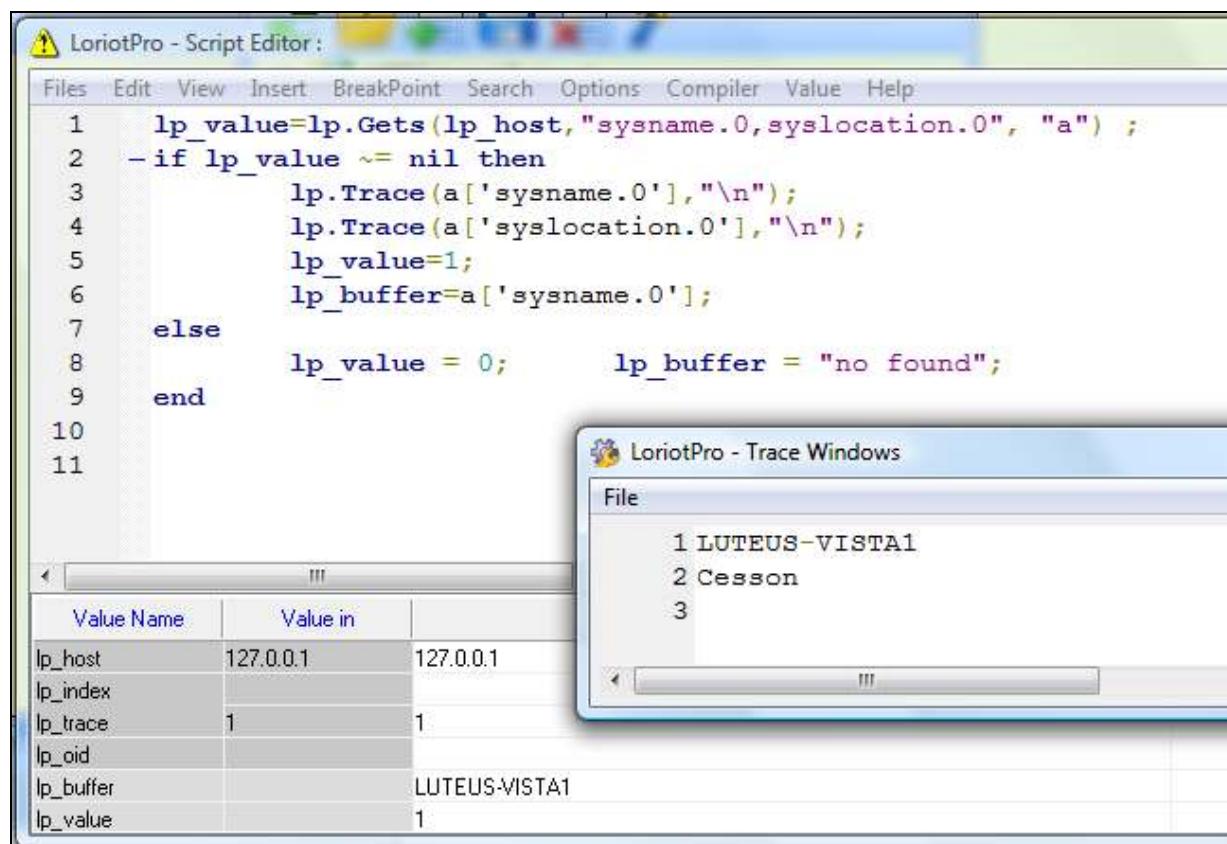
Dans le cadre des scripts d'audit il est possible de retourner d'autres variables globales au module pour assurer la création de graphe.

<b>lp_graph</b>	1 si un graphe est généré par ce script d'audit
<b>lp_graph_unit</b>	L'unité à utiliser pour le graphe
<b>lp_graph_title</b>	Le titre du graphe
<b>lp_graph_type</b>	Le graphe type (counter ou gauge) 0 ou 1
<b>lp_graph_value</b>	Le valeur retournée par le script d'audit au module d'audit.

### 5.1.2 Exemple

Dans cet exemple, le script reçoit l'adresse IP de l'équipement (du host) dans la variable **lp\_host** et retourne les résultats dans les variables **lp\_value** et **lp\_buffer**.

```
lp_value=lp.Gets(lp_host,"sysname.0,syslocation.0", "a") ;  
  
if lp_value ~= nil then  
    lp.Trace(a['sysname.0'], "\n");  
    lp.Trace(a['syslocation.0'], "\n");  
    lp_value=1;  
    lp_buffer=a['sysname.0'];  
else  
    lp_value = 0;  
    lp_buffer = "no found";  
end
```



Le script est lancé ici en utilisant l'environnement de développement fourni avec le logiciel. On peut voir en grisé foncé les variables d'entrée et en grisé clair les variables de sortie. Dans cet exemple, l'équipement (`lp_host`) 127.0.0.1 est passé en variable d'entrée . La fonction `IpGets` utilise cette variable pour réaliser une requête SNMP de collecte des variables `sysname` et `syslocation`. A ce stade, il est impératif que l'équipement (`lp_host`) soit défini dans la Directory car la fonction `IpGets` va utiliser les ressources de la Directory pour réaliser la requête SNMP en utilisant les paramètres de l'équipement 127.0.0.1. Si la requête aboutit, le script renseigne les valeurs `lp_value` et `lp_buffer` qui seront utilisées par le module exécutant le script. Ici l'environnement de développement remplit le champ de variables de sortie avec les deux valeurs. Le script affiche dans la boîte d'information les valeurs `sysname` et `syslocation` collectées sur (`lp_host`) 127.0.0.1 .

## 6 Librairies LUA de LoriotPro

### 6.1 Introduction

LoriotPro intègre en standard (intégrées au noyau du logiciel) deux librairies de fonctions propriétaires qui étendent le langage LUA en fournissant à l'utilisateur un accès aux ressources du logiciel.

La première librairie est appelée ***Ip*** comme LoriotPro, elle fournit un ensemble de fonctions utilisables sous la forme ***Ip.function( param )***. Elles peuvent être utilisées avec l'ensemble des types de scripts LUA que supportent LoriotPro.

La deuxième librairie est appelée ***Ipav*** comme « LoriotPro ActiveView », elle fournit un ensemble de fonctions graphiques utilisables sous la forme ***Ipav.function( param )***. Elles peuvent être utilisées uniquement dans des scripts LUA lancés à partir de la commande RunScript d'une ActiveView.

Une librairie externe ***Ipw*** est fournie pour vous aider dans la réalisation de scripts interactifs en mettant à votre disposition les différents modules graphiques de type Wizard du logiciel.

Vous pourrez trouver sur internet des librairies pour étendre les capacités de vos scripts, reportez-vous au chapitre traitant des librairies externes pour plus d'information.

Reportez-vous à la documentation de LUA 5.1.3 mise en annexe pour plus d'information concernant la syntaxe de base du langage. La suite de cette documentation donnera des exemples d'utilisation des librairies mais ne fournira pas d'information sur l'écriture du langage LUA.

Vous trouverez toutes les informations utiles sur la syntaxe du langage LUA sur [www.lua.org](http://www.lua.org) et les deux tomes suivants de cet ouvrage vous fourniront le détail des fonctions des librairies LoriotPro ainsi que des exemples.

### 6.2 Restriction d'usage des entiers 64bits

Comme la plupart des langages actuels, la gestion et l'utilisation d'entiers non signés de 64bits ne sont pas vraiment ni correctement supportées. La norme V2 du protocole SNMP intègre cependant des entiers 64bits non signés, les objets de type Counter64. Le langage LUA est basé sur l'utilisation de nombres à virgule flottante de type double. Les doubles sont des nombres à virgule flottante en double précision, stockés usuellement sur 64 bits. La plage de valeurs d'un double est de  $-1.7 \times 10^{-308}$  à  $1.7 \times 10^{308}$ , la plage d'un entier de 64bits elle est de 0 18446744073709551616 ou  $2^{64}$  ( $-2^{32}$  à  $2^{32}$  en mode signé). Un double peut donc en théorie couvrir la plage d'un entier de 64bits, cependant le double utilise une logique de puissance de 10 et est codé sur 64bits de la façon suivante :

52 bits pour la mantisse  
11 bits pour l'exposant  
1 bit pour le signe

Un double ne pourra pas assurer le suivi d'un compteur 64 bits avec précision car, pour de fortes valeurs, on obtiendra un arrondi et le résultat sera exploitable sous la forme d'une mantisse et d'un exposant. L'exemple de code suivant montre le problème.

The screenshot shows the LoriotPro Script Editor interface. The main window displays the following Lua script:

```
1
2     a=18446744073709551000 +10;
3     lp.Print(a, "\n");
4     a=18446744073709551000 +12;
5     lp.Print(a, "\n");
6
```

Below the editor is a "Trace Windows" window titled "LoriotPro - Trace Windows". It contains the output of the script execution:

```
File
1 1.844674407371e+019
2 1.844674407371e+019
3 |
```

Deux opérations d'addition différentes sur des nombres importants donnent le même résultat à cause de l'arrondi.

La librairie **lp** de LoriotPro intègre des fonctions pour vous aider à traiter les variables 64bits. Le principe de ces fonctions est de passer par des chaînes de caractères pour assurer les opérations de base d'addition, de soustraction, de multiplication et de division. De même si vous désirez collecter des valeurs d'objets SNMP de type Counter64, il vous faudra utiliser les fonctions de requêtes adaptées.

Fonctions	usages
lp.ascii_doubleToui64 lp.ascii_ui64ToHex	Fonctions de conversion
lp.ascii_GetNextui64 lp.ascii_Getui64 lp.ascii_SetSnmpui64	Fonctions de collecte d'objets SNMP de type Counter64
lp.ascii_GetBRCui64 lp.ascii_GetNextBRCui64 lp.ascii_SetSnmpBRCui64	Fonction de collecte d'objets SNMP de type Counter64 sur des bridges Cisco en mode multi VLAN
lp.ascii_ui64add lp.ascii_ui64div lp.ascii_ui64multi	Opérations mathématiques de base

lp.ascii_ui64rest	
lp.ascii_ui64sub	
lp.ascii_ui64or	Opérations logiques de base
lp.ascii_ui64and	
lp.ascii_ui64xor	

Le fichier **bin/config/script/lib-sample/00\_logic\_i64.lua** vous fournit des exemples d'utilisations de ces fonctions.

```
-- Loriotpro V5
-- To run correctly this file is located to bin/config/script
-- Input values
-- lp_index index for this script ".1"
-- lp_oid SNMP OID for this script "ifnumber"
-- lp_host default ip address for this script "127.0.0.1"
-- Output Values
lp_value = 0;
lp_buffer ="error";
-- dofile(lp.GetPath().."/config/script/lriotinit.lua");

val1="5085255619359801346";
val2="10";
val3 = "18446744073709551616"; --2^64

a=string.format("LUA Double for 2^64 with exposant: %f",2^64);
lp.Print(a,"\n");

lp.Print(2^64,"\n");
a=string.format("LUA Double for 2^64 with no exposant: %.f", (2^64));
lp.Print(a,"\n");
a=string.format("LUA Double for 2^64 with in hex : %X", (2^64));
lp.Print(a,"\n");
lp.Print("LoriotPro ui64 support librairy 2^64 = 18446744073709551616 \n18446744073709551615
==0xFFFFFFFFFFFFFFF\n");

lp.Print("function lp.ascii_ui64sub('18446744073709551615','3') :
",lp.ascii_ui64sub("18446744073709551615","3") ,"\n");
lp.Print("function lp.ascii_ui64add('18446744073709551615','10') :
",lp.ascii_ui64add('18446744073709551615','10') , " Warning overflow i64 \n");
lp.Print("function lp.ascii_ui64add('18446744073709551600','10') :
",lp.ascii_ui64add('18446744073709551600',"10") , " ok\n");
lp.Print("Hex 0xABCD syntax support\n");
lp.Print("function lp.ascii_ui64add('0xFFFFFFFFFFFFFFF0','10') :
",lp.ascii_ui64add('0xFFFFFFFFFFFFFFF0',"10") , " ok\n");
lp.Print("function lp.ascii_ui64add('0x0xFFFFFFFFFFFFFFF0','10') :
",lp.ascii_ui64add('0x0xFFFFFFFFFFFFFFF0',"10") , " ok\n");
lp.Print("function lp.ascii_ui64add('0x0000000000000000','10') :
",lp.ascii_ui64add('0x0000000000000000',"10") , " ok\n");

lp.Print("function lp.ascii_ui64ToHex('18446744073709551600') :
",lp.ascii_ui64ToHex('18446744073709551600') , " ok\n");

lp.Print("function double (lua number) to ascii_ui64 lp.ascii_doubleToui64(2^64) :
",lp.ascii_doubleToui64(2^64), " bad double scrap big number\n");

lp.Print("function lp.ascii_ui64add('134','3') : ",lp.ascii_ui64add("134","3") ,"\n");
lp.Print("function lp.ascii_ui64sub('134','3') : ",lp.ascii_ui64sub("134","3") ,"\n");
lp.Print("function lp.ascii_ui64multi('134','2') : ",lp.ascii_ui64multi("134","2") ,"\n");
lp.Print("function lp.ascii_ui64div('133','2') : ",lp.ascii_ui64div("133","2") ,"\n");
lp.Print("function lp.ascii_ui64rest('133','2') : ",lp.ascii_ui64rest("133","2") ,"\n");
lp.Print("function lp.ascii_ui64or('133','2') : ",lp.ascii_ui64or("133","2") ,"\n");
lp.Print("function lp.ascii_ui64or('18446744073709551615','2') :
",lp.ascii_ui64or("18446744073709551615","2") ,"\n");

lp.Print("function lp.XOR(18446744073709551615,12) : ",lp.XOR(18446744073709551615,12), " bad
double scrap big number\n";
lp.Print("function lp.ascii_ui64xor('18446744073709551615','2') :
",lp.ascii_ui64xor("18446744073709551615","2") ,"\n");
lp.Print("function lp.ascii_ui64xor('133','2') : ",lp.ascii_ui64xor("133","2") ,"\n");

lp.Print("function lp.ascii_ui64and('4','2') : ",lp.ascii_ui64and("4","2") ,"\n");
lp.Print("function lp.ascii_ui64and('18446744073709551615','2') :
",lp.ascii_ui64and("18446744073709551615","2") ,"\n");
```

## LoriotPro V5.00 Extended Edition – Supervision et gestion SNMP par scripts LUA

### Introduction au System Expert et aux scripts

```
lp.Print("function lp.ascii_doubleToui64(134) : ",lp.ascii_doubleToui64(134) ,"\n");
lp.Print("function lp.ascii_ui64add('5085974485216002051','3') :
",lp.ascii_ui64add("5085974485216002051","3") ,"\n");

a=lp.ascii_Getui64(lp_host,"sysservices");
if (a) then
lp.Print(" lp.ascii_Getui64 :",a,"\n");
end
a=lp.ascii_GetBRCui64(lp_host,"sysservices",1);
if (a) then
lp.Print(" lp.ascii_Getui64 :,a,"\n");
end
a=lp.ascii_GetNextui64(lp_host,"sysservices");
if (a) then
lp.Print("lp.ascii_GetNextui64 :,a,"\n");
end
a=lp.ascii_GetNextBRCui64(lp_host,"sysservices",1);
if (a) then
lp.Print("lp.ascii_GetNextBRCui64 :,a,"\n");
end
--[[[
a=lp.ascii_SetSnmpui64(lp_host,"location/(0)test");
if (a) then
lp.Print("lp.ascii_SetSnmpui64 :,a,"\n");
end
a=lp.ascii_SetSnmpBRCui64(lp_host,"location/(0)test",1);
if (a) then
lp.Print("lp.ascii_SetSnmpBRCui64 :,a,"\n");
end
]]]

lp.Print("Double logic support\n");

lp.Print("function lp.OR(4,1) : ",lp.OR(4,1),"\n");
lp.Print("function lp.AND(8,4) : ",lp.AND(8,4),"\n");
lp.Print("function lp.AND(8,12) : ",lp.AND(8,12),"\n");
lp.Print("function lp.XOR(8,12) : ",lp.XOR(8,12),"\n");

lp.Print("function lp.shifttoright(134,1) : ",lp.shifttoright(134,1),"\n");
lp.Print("function lp.shifttoleft(134,1) : ",lp.shifttoleft(134,1),"\n");
lp.Print("function lp.ui64div(134,2) : ",lp.ui64div(134,2) ,"\n");
lp.Print("function lp.ui64rest(133,2) : ",lp.ui64rest(133,2) ,"\n");
lp.Print("function lp.ui64div(18446744073709551615,2) : ",lp.ui64div(18446744073709551615,2)
,"\");

lp.Print("function lp.ui64rest(151,50) : ",lp.ui64rest(151,50) ,"\n");
```

# LoriotPro V5.00 Extended Edition – Supervision et gestion SNMP par scripts LUA

## Introduction au System Expert et aux scripts

The screenshot shows the LoriotPro Script Editor interface. The main window displays a LUA script titled 'lib-sample\00\_logic\_i64.lua'. The script includes comments like '-- Loriotpro V4' and '-- To run correctly this file is located to bin/config/script'. It contains several print statements and function definitions. A status bar at the bottom shows 'File' and '1 LUA Double for 2^64 with exposant: 18446744073709552000.000000'. Below the editor is a table titled 'Value Name' with columns 'Value Name' and 'Value'. The table lists variables such as 'lp\_host' (192.168.0.1), 'lp\_index' (1), 'lp\_trace' (1), 'lp\_oid' (empty), 'lp\_buffer' (empty), 'lp\_value' (empty), 'lp\_graph' (empty), 'lp\_graph\_unit' (empty), 'lp\_graph\_title' (empty), and 'lp\_type' (empty). The right side of the screen shows a taskbar with icons for IP Help, Log, and others.

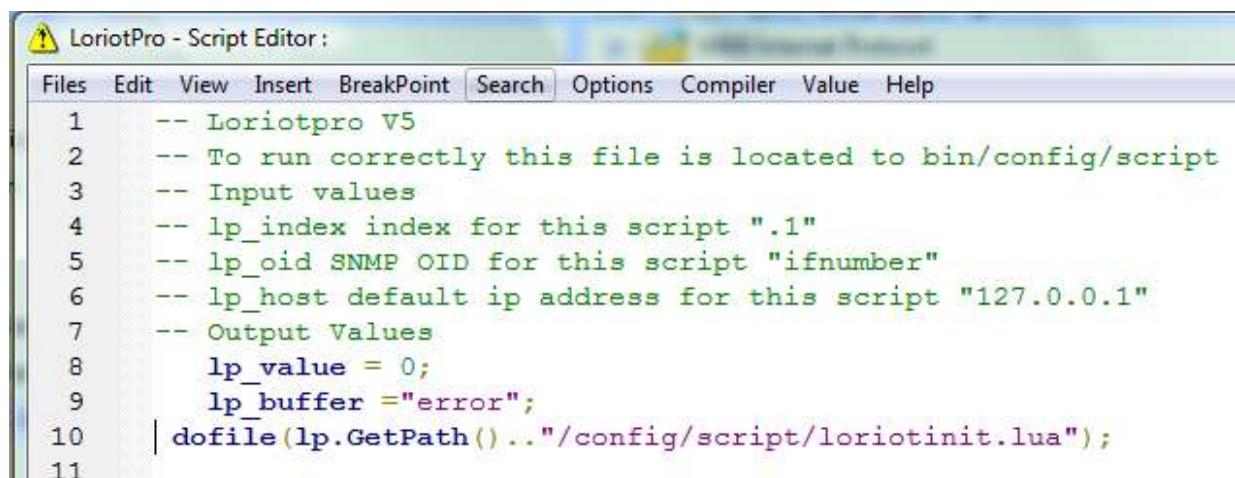
Value Name	Value
lp_host	192.168.0.1
lp_index	
lp_trace	1
lp_oid	
lp_buffer	
lp_value	
lp_graph	
lp_graph_unit	
lp_graph_title	
lp_type	

## 6.3 Fichier de définition de variables système

Le répertoire **bin/config/script** contient le fichier **loriotinit.lua** qui initialise un ensemble de variables systèmes utilisables avec les librairies. Pour activer ce fichier placez la commande suivante en début de script. Si vous n'utilisez pas ces variables, évitez de les initialiser pour accélérer l'exécution du script.

```
lp_value = 0;
lp_buffer ="error";

dofile(lp.GetPath().."/config/script/oriotinit.lua");
```



The screenshot shows the LoriotPro Script Editor interface. The menu bar includes Files, Edit, View, Insert, BreakPoint, Search, Options, Compiler, Value, and Help. The main code area contains the following LUA script:

```
1 -- Loriotpro v5
2 -- To run correctly this file is located to bin/config/script
3 -- Input values
4 -- lp_index index for this script ".1"
5 -- lp_oid SNMP OID for this script "ifnumber"
6 -- lp_host default ip address for this script "127.0.0.1"
7 -- Output Values
8     lp_value = 0;
9     lp_buffer ="error";
10    | dofile(lp.GetPath() .. "/config/script/loriotinit.lua");
11
```

Il est possible de ne pas utiliser ces variables et de les remplacer par leur valeur nominale dans les fonctions, pour cela aidez-vous du contenu du fichier **bin/config/script/loriotinit.lua**:

### loriotinit.lua

```
-- LoriotPro LUA init file V1.06
-- (c) Luteus Sarl 2006-07
```

```
--status
LP_LUA_STATUS_NONE=0
LP_LUA_STATUS_ICMP=1;
LP_LUA_STATUS_SNMP=2;
LP_LUA_STATUS_WARNING=3;
LP_LUA_STATUS_DOWN=4;

--lp setoption
LP_LUA_NAME=1;
LP_LUA_COMMUNITYRO=2;
LP_LUA_COMMUNITYRW=3;
LP_LUA_USER1=4;
LP_LUA_USER2=5;
LP_LUA_USER3=6;
LP_LUA_SYSID=7;
LP_LUA_SYSNAME=8;
LP_LUA_GROUPREF=9;
LP_LUA_POLLING_SNMP=10;
LP_LUA_POLLING_PING=11;
LP_LUA_POLLING=12;
LP_LUA_ON=1;
LP_LUA_OFF=0;
LP_LUA_ISDEVICE =13;
LP_LUA_ISDEVICE_AUTO = 0;
LP_LUA_ISDEVICE_HOST = 1;
LP_LUA_ISDEVICE_SERVER = 2;
LP_LUA_ISDEVICE_HUB = 3;
LP_LUA_ISDEVICE_SWITCH = 4;
LP_LUA_ISDEVICE_ROUTER = 5;
LP_LUA_ISDEVICE_PRINTER = 6;
LP_LUA_ISDEVICE_PHONE = 7;
LP_LUA_STATUS =14;
LP_LUA_MAC=15;
LP_LUA_IP_SWITCH=16;
LP_LUA_IP_SWITCH_INDEX=17;
LP_LUA_NUMBER_STATUS_3=18;
LP_LUA_NUMBER_STATUS_4=19;
LP_LUA_IP_DDNS_RESOLVE=20;
LP_LUA_IP_DDNS_NAME=21;
LP_LUA_IP_DDNS_STATUS=22;
LP_LUA_IP_CHANGE=23;
LP_LUA_CRITICITY=24;
LP_LUA_CRITICITY_EVENT=25;
LP_LUA_FLAG_IPSCANNER=26;
LP_LUA_IPSCANNER_INTERVAL=27;
```

## LoriotPro V5.00 Extended Edition – Supervision et gestion SNMP par scripts LUA Introduction au System Expert et aux scripts

```
LP_LUA_IPSCANNER_STATEGIE=28;
LP_LUA_DOUBLECLICK=29;
LP_LUA_IPFORWARDING=30;
LP_LUA_IPNEXTHOP=31;
LP_LUA_DISCOVERSCANNING=32;
LP_LUA_SYSSERVICES=33;
LP_LUA_NET_IPSCANNER=34;
LP_LUA_NET_DESCRIPTION=35;
LP_LUA_NET_NAME=36;
LP_LUA_NET_STRATEGY=37;
LP_LUA_NET_IPSCANNER_INTERVAL=38;
LP_LUA_SLA=39;
LP_LUA_FLAG_IPSCANNER_NOCONFIG=40;
LP_LUA_IPSCANNER_EVENT=41;
LP_LUA_IPSCANNER_EVENT_LEVEL=42;

--ParamType
LP_LUA_DT_COUNTRY=1;
LP_LUA_DT_ORGANIZATION =2;
LP_LUA_DT_ORGANIZATION_UNIT=3;
LP_LUA_DT_NETWORK=4;
LP_LUA_DT_FACTORY=5;
LP_LUA_DT_HOST= 6;
LP_LUA_DT_HOST_FUNCTION=7;
LP_LUA_DT_HOST_ALIAS=8;
LP_LUA_DT_HOST_EXECLINK=9;
LP_LUA_DT_HOST_MRTG=10;
LP_LUA_DT_HOST_PLUGIN=11;
LP_LUA_DT_HOST_ADV POLLING=12;
LP_LUA_ADV_CONDITION_DOWN= 1;
LP_LUA_ADV_CONDITION_BECOME DOWN= 2;
LP_LUA_ADV_CONDITION_UP= 3;
LP_LUA_ADV_CONDITION_BECOME UP= 4;

--lpav object type define --lpav_InsertObject

LPAV_OBJ_WORLD =0;
LPAV_OBJ_COUNTRY =1;
LPAV_OBJ_ORGANIZATION =2;
LPAV_OBJ_ORGANIZATION_UNIT = 3;
LPAV_OBJ_NETWORK =4;
LPAV_OBJ_HOST_1 =6;
LPAV_OBJ_ROUTER =7;
LPAV_OBJ_SHORTCUT_1 =8;
LPAV_OBJ_SHORTCUT_2 =9;
LPAV_OBJ_MRTG =10;
LPAV_OBJ_PLUGIN =11;
LPAV_OBJ_BADPLUGIN = 12;
LPAV_OBJ_LORIOTPRO = 13;
LPAV_OBJ_HOST_2_ROUTER = 15;
LPAV_OBJ_HOST_3_SWITCH =16;
LPAV_OBJ_HOST_4 = 17;
LPAV_OBJ_HOST_5 = 18;
LPAV_OBJ_HOST_6 = 19;
LPAV_OBJ_VIEWPLUGIN =20
LPAV_OBJ_HOST_SERVER = 27;
LPAV_OBJ_HOST_IMPRIMANTE =28;
LPAV_OBJ_HOST_PHONE =29;
LPAV_OBJ_RECT = 30;
LPAV_OBJ_CIRCLE = 31;
LPAV_OBJ_FILL_RECT = 32;
LPAV_OBJ_FILL_CIRCLE = 33;
LPAV_OBJ_H_LINE = 34;
LPAV_OBJ_V_LINE = 35;
LPAV_OBJ_DIR_EMF = 36;
LPAV_OBJ_TEXT = 37;
LPAV_OBJ_FILL_TEXT = 38;
LPAV_OBJ_FULL_RECT = 39;
LPAV_OBJ_FULL_CIRCLE = 40;
LPAV_OBJ_H45_LINE = 41;
LPAV_OBJ_V45_LINE = 42;
LPAV_OBJ_FULL_TEXT = 43;
LPAV_OBJ_BASIC_SHAPE = 44;
LPAV_OBJ_ANALOGIC_METER_01 = 45;
```

## LoriotPro V5.00 Extended Edition – Supervision et gestion SNMP par scripts LUA Introduction au System Expert et aux scripts

```
LPAV_OBJ_HOST_HUB = 46;
LPAV_OBJ_CONNECTOR =144;

--lpav.SetOption

LPAV_NAME =1;
LPAV_FONTNAME =2;
LPAV_FONTSIZE =3;
LPAV_LINEWIDTH =4;
LPAV_LINESTYLE =5;
LPAV_EXPRESSION =6;
LPAV_HIDEN_TEXT =7;
LPAV_NOBORDER =8;
LPAV_TEXTMODE =9;
LPAV_TEXTSTYLE =10;
LPAV_UNIT =11 ;
LPAV_OBJECTTYPE =12;
LPAV_POLLING =13;
LPAV_POLLING_INTERVAL =14;
LPAV_SELECTED =15;
LPAV_X =16;
LPAV_X1 =17;
LPAV_Y =18;
LPAV_Y1 =19;
LPAV_TYPE =20;

LPAV_CONNECTED_TO1 =21;
LPAV_CONNECTED_TO2 =22;
LPAV_ISRATIO =23;
LPAV_LEFT_MOUSE_MENU =24;
LPAV_RESULT_IS_TEXT =25;
LPAV_RESULT_IS_NAME =26;
LPAV_REF1 =27;
LPAV_REF2 =28;
LPAV_REF3 =29;
LPAV_REF4 =30;
LPAV_ISCOUNTER =31;
LPAV_CONNECTOR_AUTO_ALIGN =32;
LPAV_NOIPSOURCE_TEST =33;
LPAV_NOSELECTABLE =34;
LPAV_CLIPART =35;
LPAV_STATUS = 36;
LPAV_MAPTYPE =37;

--lpav.SetGlobalOption
LPAV_G_DEFAULTIP =1;
LPAV_G_MAPNAME =2;
LPAV_G_FONTNAME =3;
LPAV_G_FONTSIZE =4;
LPAV_G_FONTITALIC =5;
LPAV_G_LINEWIDTH =6;
LPAV_G_LINESTYLE =7;
LPAV_G_MAX_X =8;
LPAV_G_MAX_Y =9;
LPAV_G_ZOOMAUTO =10;
LPAV_G_ZOOMINDEX =11;
LPAV_G_SCALEX=12 ;
LPAV_G_SCALEY=13 ;

--lpav.InsertAction lpav.InsertMenu
LPAV_ACTION_NONE =0;
LPAV_ACTION_LOADMAP =1;
LPAV_ACTION_WINEXEC =2;
LPAV_ACTION_SHELLEXEC =3;
LPAV_ACTION_PLUGIN =4;
LPAV_ACTION_TABLE =5;
LPAV_ACTION_SCRIPTREP =6;
LPAV_ACTION_SCI =7;
LPAV_ACTION_FOUND_INDIR =8;
LPAV_ACTION_MODAL_MAP =9;
LPAV_ACTION_SET_OID =10;
LPAV_ACTION_CLEAR_EVENTREF =11;
LPAV_ACTION_CLEAR_EVENTFILTER =12;
LPAV_ACTION_CLEAR_TRAPFILTER =13;
LPAV_ACTION_CLEAR_EVENTGENERATED =14;
LPAV_ACTION_CLEAR_TRAPGENERATED =15;
```

```

LPAV_ACTION_CLEAR_TRAPREF =16;
LPAV_ACTION_RUN_SCRIPT_LUA = 17;

--lpav.InsertFilter

LPAV_FILTER_CONDITION_EGAL =0;
LPAV_FILTER_CONDITION_SUP =1;
LPAV_FILTER_CONDITION_INF =2;
LPAV_FILTER_CONDITION_DIF =3;
LPAV_FILTER_CONDITION_SUP_EGAL =4;
LPAV_FILTER_CONDITION_INF_EGAL =5;
LPAV_FILTER_CONDITION_FIND =6;
LPAV_FILTER_CONDITION_BAD_EXPRESSION =7;
LPAV_FILTER_CONDITION_ANY =8;

```

## 6.4 LoriotPro Kernel Librairie (Ip)

Cette librairie est attachée à la dll **dsmnp\_kernel42.dll** ; elle fournit l'ensemble des fonctions de base permettant d'exploiter les ressources du noyau de LoriotPro. Le fichier **bin/config/lua\_editor.ini** contient l'ensemble des fonctions disponibles ainsi qu'un rappel simple de leur syntaxe. Ce fichier est utilisé par l'environnement de développement pour gérer les fonctions de colorisation du texte et de l'affichage des tooltips.

Les fonctions de la librairie ont une syntaxe du type :

`lp.GetNext('value',...);`

Le tableau suivant est là uniquement pour vous fournir la liste des fonctions disponibles ainsi que leur syntaxe, consultez le manuel de référence des librairies pour une explication détaillée de chaque fonction et le manuel d'exploitation pour des exemples d'utilisation de ces fonctions. Les noms de variables entre guillemets simples correspondent à des variables de type chaîne de caractères. Le nombre de fonctions disponibles avec la version 5 de LoriotPro Extended Edition a doublé depuis la version 4.

### Tableau de définition de la librairie

Fonctions	Syntaxe simplifié
lp.Get	<code>lp.Get('ip','oid');</code>
lp.Set	<code>lp.Set('ip','oid');</code>
lp.GetNext	<code>lp.GetNext('ip','oid');</code>
lp.Gets	<code>lp.Gets('ip','oid','array');</code>
lp.GetRows	<code>lp.GetRows('ip','oid,oid,oid','array');</code>
lp.GetPath	<code>lp.GetPath();</code>
lp.Trace	<code>lp.Trace('texte');</code>
lp.Break	<code>lp_Break();</code>
lp.Stop	<code>lp.Stop();</code>
lp.GetFirstIP	<code>lp.GetFirstIP();</code>
lp.GetNextIP	<code>lp.GetNextIP('ip');</code>
lp.GetIPInformation	<code>lp.GetIPInformation('ip','array');</code>
lp.SetIPOption	<code>lp.SetIPOption('ip',param_type,'value');</code>
lp.GetUIDInformation	<code>lp.GetUIDInformation('uid','array');</code>
lp.GetChildUID	<code>lp.GetChildUID('uid','array');</code>
lp.atoip	<code>lp.uptoa(ip);&amp;</code>
lp.GetIPFromMAC	<code>lp.GetIPFromMAC('mac');</code>
lp.FindIPFromDynamicDNS	<code>lp.FindIPFromDynamicDNS('www.google.com');</code>

# LoriotPro V5.00 Extended Edition – Supervision et gestion SNMP par scripts LUA

## Introduction au System Expert et aux scripts

---

lp.SendEvent	lp.SendEvent( eventnumber,level,'ipref','ipmask','buffer');
lp.SendExternEvent	lp.SendExternEvent( 'ipdest', port, eventnumber, level, 'ipref','ipmask','buffer');
lp.IsLoadedMIBRef	lp.IsLoadedMIBRef('REF-MIB');
lp.FindSNMPObject	lp.FindSNMPObject('sysname');
lp.GetFirstNetwork	lp.GetFirstNetwork();
lp.GetNextNetwork	lp.GetNextNetwork('net','mask');
lp.GetNetworkFromIP	lp.GetNetworkFromIP('ip');
lp.GetFirstRouter	lp.GetFirstRouter();
lp.GetNextRouter	lp.GetNextRouter('ip_groupref');&
lp.GetNetworkInformation	lp.GetNetworkInformation('net','mask','array');
lp.GetRouterInformation	lp.GetRouterInformation('ip_groupref','array');
lp.LoadLibrary	lp.LoadLibrary('library_path-name');
lp.HexToOID	lp.HexToOID('xx:xx:xx:xx:');
lp.GetRowsBRC	lp.GetRowsBRC('ip','oid,oid,oid','array',VlanID);
lp.GetNextBRC	lp.GetNextBRC('ip','oid',VlanID);
lp.GetBRC	lp.GetBRC('ip','oid',vlanid);
lp.SetBRC	lp.SetBRC('ip','oid',vlanid);
lp.GetsBRC	lp.GetsBRC('ip','oid','array',VlanID);
lp.Print	lp.Print('texte',texte,...);
lp.GetVersion	lp.GetVersion();
lp.IsDebugMode	lp.IsDebugMode();
lp.InsertNewHost	lp.InsertNewHost('ip','name','father uid',mode);
lp.InsertNewContainer	lp.InsertNewContainer('name','father uid',type);
lp.InsertNewNetwork	lp.InsertNewNetwork( 'name', 'desc', 'net', 'mask', type, speed, 'father uid');
lp.FindName	lp.FindName('name');
lp.SaveDirectoryAs	lp.SaveDirectoryAs('filename');
lp.GetTableEntryList	lp.GetTableEntryList('entryname','array');
lp.atof	lp.atof('text number');
lp.GetAllHostFromContainer	lp.GetAllHostFromContainer('uid','array');
lp.GetSNMPObjectDescription	lp.GetSNMPObjectDescription('sysname');
lp.GetTrapFilterList	lp.GetTrapFilterList('array');
lp.GetTrapFilterInformation	lp.GetTrapFilterInformation('uid','array');
lp.GetTrapFilterActionList	lp.GetTrapFilterActionList('uid','array');
lp.GetTrapFilterActionInformation	lp.GetTrapFilterActionInformation('uid- action','array');
lp.GetEvenList	lp.GetEvenList('array');
lp.GetEventInformation	lp.GetEventInformation(number,'array');
lp.GetEventActionList	lp.GetEventActionList(number,'array');
lp.GetEventFilterActionInformation	.GetEventFilterActionInformation( 'uidaction' 'array');
lp.AckEvent	lp.AckEvent(number);
lp.AckEventFilterAction	lp.AckEventFilterAction('uid-action');
lp.AckTrapFilter	lp.AckTrapFilter('uid');
lp.AckTrapFilterAction	lp.AckTrapFilterAction('uid-action');
lp.ClearIPEventCounter	lp.ClearIPEventCounter('ip');
lp.ClearIPTrapCounter	lp.ClearIPTrapCounter('ip');
lp.GetLoriotProID	lp.GetLoriotProID();
lp.SetEventFilterActionDisable	lp.SetEventFilterActionDisable( 'uid', 0 or 1);
lp.SetTrapFilterActionDisable	lp.SetTrapFilterActionDisable('uid',0 or 1);
lp.SetEventFilterDisable	lp.SetEventFilterDisable(number,0 or 1);
lp.SetTrapFilterDisable	lp.SetTrapFilterDisable('uid',0 or 1);
lp.GetTrapFilterGlobalStatus	lp.GetTrapFilterGlobalStatus();
lp.GetEventFilterGlobalStatus	lp.GetEventFilterGlobalStatus();
lp.SetTrapFilterGlobalStatus	lp.SetTrapFilterGlobalStatus( 0 or 1);
lp.SetEventFilterGlobalStatus	lp.SetEventFilterGlobalStatus( 0 or 1);
lp.Parse	lp.Parse('texte','array','token');
lp.Fgets	lp.Fgets(file);
lp.Fclose	lp.Fclose(file);
lp.Fopen	lp.Fopen('filename','wt')
lp.Fprintf	lp.Fprintf(file,'texte',number,...);
lp.CopyFile	lp.CopyFile('filename1','filename2');
lp.FindFile	lp.FindFile('path','ext','array');

# LoriotPro V5.00 Extended Edition – Supervision et gestion SNMP par scripts LUA

## Introduction au System Expert et aux scripts

---

lp.RemoveFile	lp.RemoveFile('filename');
lp.GetFileAttrib	lp.GetFileAttrib('path/filename','array');
lp.MakeDir	lp.MakeDir('path');
lp.RemoveDir	lp.RemoveDir('path');
lp.ShowPlugin	lp.ShowPlugin('uid');
lp.GetIPAddress	lp.GetIPAddress('www.name.com')
lp.HidePlugin	lp.HidePlugin('uid');
lp.CliPlugin	lp.CliPlugin('uid', 'Command');
lp.SendSyslog	lp.SendSyslog('ip server','buffer');
lp.InsertADVAuditPolling	lp.InsertADVAuditPolling( 'ip', audit_ref, 'param', poll_interval, enable, gen_audio, gen_report, event, level, condition, sla, ['title']);
lp.UpdateADVAuditPolling	lp.UpdateADVAuditPolling('ip', audit_ref, 'param', poll_interval, enable, gen_audio, gen_report, event, level, condition, sla);
lp.FoundADVAuditUID	lp.FoundADVAuditUID('ip',audit_ref,'param');
lp.EnableDirectoryObject	lp.EnableDirectoryObject('uid');
lp.LocateDirectoryUID	lp.LocateDirectoryUID('uid');
lp.UpdateDirectory	lp.UpdateDirectory();
lp.SetNetworkOption	lp.SetNetworkOption('net', 'mask', variable, 'value');
lp.OR	lp.OR(value1 , value2);
lp.XOR	lp.XOR(value1 , value2);
lp.AND	lp.AND(value1 , value2);
lp.shifttoright	lp.shifttoright(value1,shift );
lp.shifttoleft	lp.shifttoleft(value1,shift );
lp.ui64div	lp.ui64div(value1 , value2);
lp.ui64rest	lp.ui64rest(value1 , value2);
lp.ascii ui64add	lp.ascii ui64add('value1' , 'value2');
lp.ascii ui64sub	lp.ascii ui64sub('value1' , 'value2');
lp.ascii ui64div	lp.ascii ui64div('value1' , 'value2');
lp.ascii ui64rest	lp.ascii ui64rest('value1' , 'value2');
lp.ascii ui64multi	lp.ascii ui64multi('value1' , 'value2');
lp.ascii ui64or	lp.ascii ui64or('value1' , 'value2');
lp.ascii ui64xor	lp.ascii ui64xor('value1' , 'value2');
lp.ascii ui64and	lp.ascii ui64and('value1' , 'value2');
lp.ascii Getui64	lp.ascii Getui64('ip','oid');
lp.ascii GetBRCui64	lp.ascii GetBRCui64('ip','oid',vlanid);
lp.ascii GetNextui64	lp.ascii GetNextui64('ip','oid');
lp.ascii GetNextBRCui64	lp.ascii GetNextBRCui64('ip','oid',vlanid);
lp.ascii SetSnmpui64	lp.ascii SetSnmpui64('ip','oid');
lp.ascii SetSnmpBRCui64	lp.ascii SetSnmpui64('ip','oid',vlanid);
lp.ascii ui64ToHex	lp.ascii ui64ToHex('1234');
lp.ascii doubleToui64	lp.ascii doubleToui64(value);
lp.GetSnmpObjectInformations	lp.GetSnmpObjectInformations('oid','array');
lp.FoundADVTCPUID	lp.FoundADVTCPUID('ip',tcp_port);
lp.InsertADVTCPPolling	lp.InsertADVTCPPolling('ip', tcp_port, poll_interval, enable, gen_audio, gen_report, event, level, condition, sla, ['title']);
lp.UpdateADVTCPPolling	lp.UpdateADVTCPPolling('ip', tcp_port, poll_interval, enable, gen_audio, gen_report, event, level, condition, sla);
lp.UpdateADVTCPPollingUID	lp.UpdateADVTCPPollingUID('uid', poll_interval, enable, gen_audio, gen_report, event, level, condition, sla);
lp.SendTrapV1	lp.SendTrapV1('ipdest', 'ipagent', 'community', 'trapname', type, specific, 'iod,oid');
lp.SendTrapV2	lp.SendTrapV2('ipdest', 'community', 'trapname', 'iod1,oid2');

## 6.5 LoriotPro ActiveView Librairie (Ipav)

Cette librairie est attachée à la dll **dsnmp\_exctrl42.dll** et n'est utilisable que dans le cadre d'un script d'ActiveView.

Les fonctions de la librairie ont une syntaxe du type :

**Ipav.SetPosition (« value »,...) ;**

Cette librairie a pour objet de fournir des fonctions de base pour travailler sur les objets graphiques (vectoriels) d'une ActiveView. L'ensemble des fonctions utilise la notion d'objet par défaut. Quelques fonctions permettent de définir l'objet par défaut (actif). Lorsque qu'un nouvel objet est créé (inséré) il devient automatiquement l'objet par défaut. Il est possible de sélectionner un ensemble d'objets qui devient « l'objet par défaut » ; dans ce cas, certaines fonctions appliqueront les modifications demandées sur l'ensemble des objets. Chaque objet de l'ActiveView possède un UID permettant de le sélectionner individuellement. Chaque objet possède aussi X variables permettant des regroupements et des sélections multiples.

On peut associer un script à chaque objet d'une ActiveView, cependant l'éditeur ne permet de travailler que sur deux scripts à la fois. Chaque objet d'une ActiveView pouvant faire tourner un script de façon indépendante ou coordonnée, il est possible de réaliser des programmes très complexes. Le moteur de lancement des scripts d'une ActiveView est séquentiel, les scripts se dérouleront donc les uns après les autres. L'ordre de réalisation des scripts dépend de la liste chaînée des objets en mémoire.

### Tableau de définition de la librairie

Fonctions	Syntaxe simplifié
lpav.SetPosition	lpav.SetPosition(x,y,x1,y1);
lpav.GetSize	lpav.GetSize(x,y);
lpav.GetPosition	lpavGetPosition();
lpav.RefreshMap	lpav.RefreshMap();
lpav.SetClipart	lpav.SetClipart('path/filename.wmf');
lpav.SetToBackGround	lpav.SetToBackGround();
lpav.SetToFrontGround	lpav.SetToFrontGround();
lpav.SetDown	lpav.SetDown();
lpav.SetUp	lpav.SetUp();
lpavSetFontName	lpavSetFontName('fontname');
lpavSetName	lpavSetName('name');
lpavSetBrush	lpavSetBrush(R,G,B);
lpavSetLineColor	lpavSetLineColor(R,G,B);
lpavSetLineWidth	lpavSetLineWidth(Width);
lpavGetInformation	lpavGetInformation('array')
lpavSetCoord	lpavSetCoord(x,y);
lpavInsert	lpavInsert(x,y,cx,cy,type);
lpavSelectUID	lpavSelectUID('uid');
lpavSetRef	lpavSetRef(number (1-4), string);
lpavFindRef	lpavFindRef(number (1-4), mode, 'string');
lpavGetRef	lpavGetRef(number (1-4));
lpavDeleteAll	lpavDeleteAll();
lpavSetSelected	lpavSetSelected();
lpavClearSelected	lpavClearSelected();
lpavClearAllSelected	lpavClearAllSelected();

## LoriotPro V5.00 Extended Edition – Supervision et gestion SNMP par scripts LUA Introduction au System Expert et aux scripts

lpav.ClearMultiSelect	lpav.ClearMultiSelect();
lpav.IsMultiSelectedMode	lpav.IsMultiSelectedMode();
lpav.DeleteAllMenu	lpav.DeleteAllMenu();
lpav.SelectAllRef	lpav.SelectAllRef(number (1-4), mode, 'string')
lpav.Delete	lpav.Delete();
lpav.SetOption	lpav.SetOption(optionref, value);
lpav.Save	lpav.Save('filename');
lpav.Append	lpav.Append('filename');
lpav.Load	lpav.Load('filename');
lpav.SelectFirst	lpav.SelectFirst();
lpav.SelectNext	lpav.SelectNext();
lpav.SetGlobalOption	lpav.SetGlobalOption(optionref, 'value');
lpav.GetGlobalOption	lpav.GetGlobalOption('array');
lpav.RequestString	lpav.RequestString('box_caption');
lpav.InsertMenu	lpav.InsertMenu(type, 'caption','parameters');
lpav.InsertAction	lpav.InsertAction(type,'parameters');
lpav.SetExpression	lpav.SetExpression('expression');
lpav.InsertObject	lpav.InsertObject(type, 'name',x,y,'UID');
lpav.FindFromAssociatedUID	lpav.FindFromAssociatedUID('uid-in');
lpav.ShiftPosition	lpav.ShiftPosition(x,y,x1,y1);
lpav.ShiftPositionAllSelected	lpav.ShiftPositionAllSelected(x,y,x1,y1);
lpav.GetRectAllSelected	lpav.GetRectAllSelected();
lpav.SetOptionAllSelected	lpav.SetOptionAllSelected(optionref, 'value');
lpav.SelectAll	lpav.SelectAll();
lpav.CopyAllSelected	lpav.CopyAllSelected();
lpav.Paste	lpav.Paste(mode);
lpav.Copy	lpav.Copy(mode);
lpav.SetUpCornerAllSelected	lpav.SetUpCornerAllSelected(x,y);
lpav.SizeInRectAllSelected	lpav.SizeInRectAllSelected(x,y,x1,y1);
lpav.UnSelectAllRef	lpav.UnSelectAllRef(number (1-4), mode, string);
lpav.InsertFilter	lpav.InsertFilter(condition, 'value', color, event, threshold, 'ip', 'mask', level,'msg');
lpav.SaveGraphToJpeg	lpav.SaveGraphToJpeg('filename',quality);

### 6.6 LoriotPro Wizard Librairie (lpw)

Cette librairie donne accès à certains modules Wizard de LoriotPro. C'est une librairie dynamique « dll » chargeable avec la fonction lp.LoadLibrary. Cette librairie étant gourmande en mémoire, elle n'est chargée qu'à la demande.

#### Attention

Il ne faut pas utiliser les ressources de cette librairie dans des objets SNMP virtuels, ou dans des scripts d'objets graphiques (schedulés) mais uniquement dans des scripts lancés unitairement.

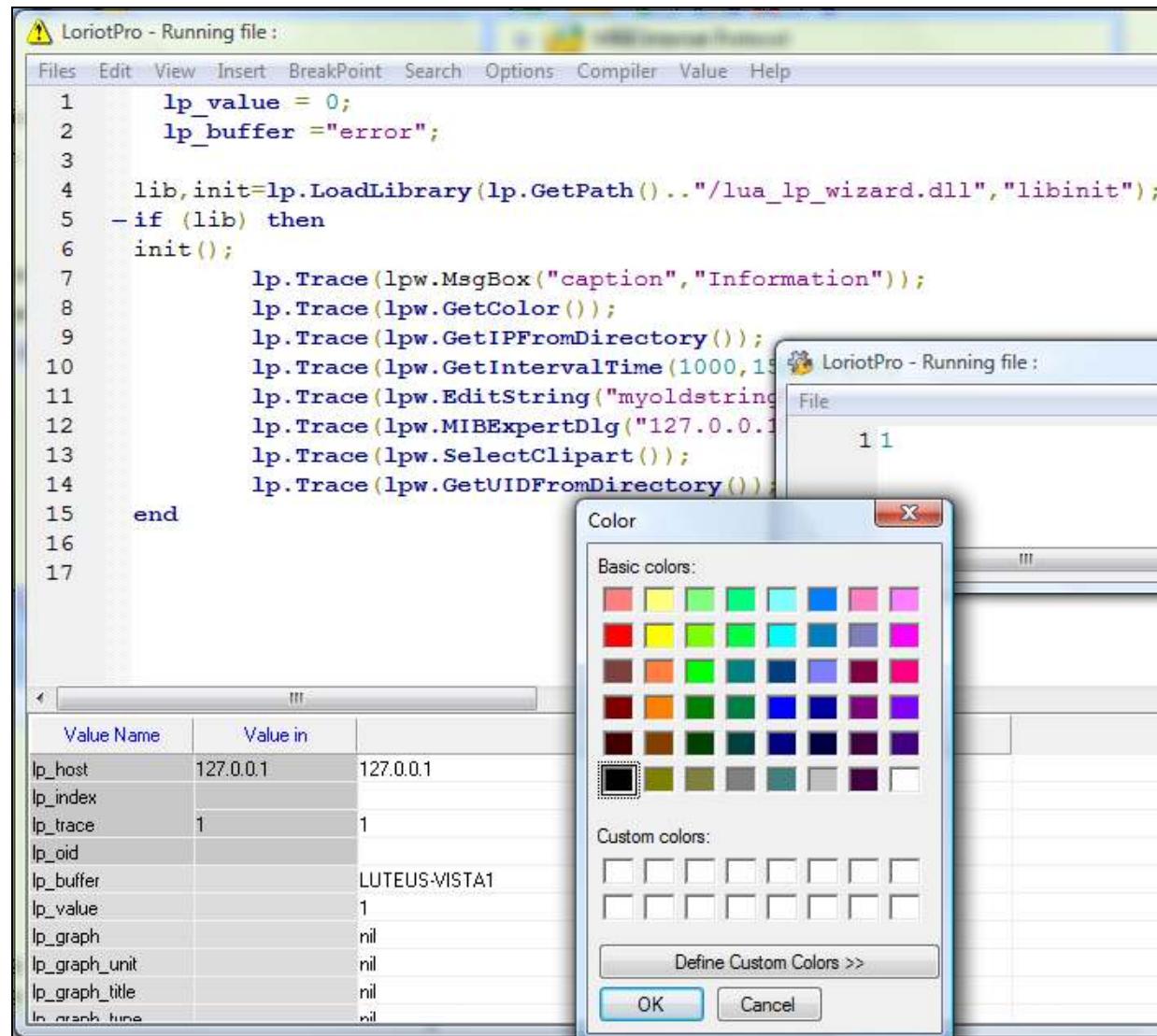
Code minimum pour initialiser la librairie :

```
lib.init=lp.LoadLibrary(lp.GetPath().."/lua_lp_wizard.dll","libinit");
if (lib) then
init();
end
```

## Exemple

```
lp_value = 0;
lp_buffer ="error";

lib.init=lp.LoadLibrary(lp.GetPath().."/lua_lp_wizard.dll","libinit");
if (lib) then
init();
    lp.Trace(lpw.MsgBox("caption","Information"));
    lp.Trace(lpw.GetColor());
    lp.Trace(lpw.GetIPFromDirectory());
    lp.Trace(lpw.GetIntervalTime(1000,15000,1));
    lp.Trace(lpw>EditString("myoldstring","please change this string"));
    lp.Trace(lpw.MIBExpertDlg("127.0.0.1"));
    lp.Trace(lpw.SelectClipart());
    lp.Trace(lpw.GetUIDFromDirectory());
end
```



## Tableau de définition de la librairie

Fonctions	Syntaxe simplifié
lpw.GetColor	lpw.GetColor();
lpw.GetIPFromDirectory	lpw.GetIPFromDirectory();
lpw.GetIntervalTime	lpw.GetIntervalTime(value,min,true);
lpw.GetEventNumber	lpw.GetEventNumber();
lpw>EditString	lpw.EditString('value','caption');
lpw.MIBExpertDlg	lpw.MIBExpertDlg('ip');
lpw>SelectClipart	lpw.SelectClipart();
lpw.GetUIDFromDirectory	lpw.GetUIDFromDirectory();
lpw.RequestString	lpw.RequestString('caption');
lpw.TimedMessageBox	lpw.TimedMessageBox(timeout,'caption');
lpw.FileDialog	lpw.FileDialog('path','ext',TRUE/FALSE);
lpw.MIBTablePicker	lpw.MIBTablePicker();
lpw.SetSnmpDlg	lpw.SetSnmpDlg('ip','oid');
lpw.MibTabDlg	lpw.MibTabDlg('ip','table');
lpw.ReportGeneratorDlg	lpw.ReportGeneratorDlg('ip','filein.rep','fileout');
lpw.ShellExecute	lpw.ShellExecute('url')
lpw.LoadPlugin	lpw.LoadPlugin('ip','plugin name','param'/nil);

## 6.7 LoriotPro SLA (Ipsla) Librairie

Cette librairie donne accès à des fonctions permettant d'exploiter les données SLA associées aux équipements de la Directory. Il faut, pour exploiter des informations de collectes SLA, que l'équipement soit configuré pour permettre la sauvegarde de ces informations.

La librairie externe **bin/lua\_lp\_sla.dll** permet de réaliser le calcul du SLA d'un host pour une période donnée.

Initialisation de la librairie dans un script

```
if (Ip.IsDebugMode()==1) then
    lib,init=Ip.LoadLibrary(Ip.GetPath().."/lua_lp_slad.dll","libinit");
else
    lib,init=Ip.LoadLibrary(Ip.GetPath().."/lua_lp_sla.dll","libinit");
end
```

Exemple d'utilisation

```
-- Loriotpro V5
-- To run correctly this file is located to bin/config/script
-- Input values
-- lp_index index for this script ".1"
-- lp_oid SNMP OID for this script "ifnumber"
-- lp_host default ip address for this script "127.0.0.1"
-- Output Values
lp_value = 0;
lp_buffer ="error";
-- dofile(lp.GetPath() .. "/config/script/loriotinit.lua");

if (lp.IsDebugMode()==1) then
lib,init=lp.LoadLibrary(Ip.GetPath().."/lua_lp_slad.dll","libinit");
```

## LoriotPro V5.00 Extended Edition – Supervision et gestion SNMP par scripts LUA

### Introduction au System Expert et aux scripts

```
else
lib,init=lp.LoadLibrary(lp.GetPath().."/lua_lp_sla.dll","libinit");
end

if (lib) then

init();

id="1002";
k=lpsla.GetLoriotProIDList("a");

for l=0,k-1 do
    lp.Print(a[l]," LoriotPro ID \n");

i=lpsla.GetSLAList(a[l],"aa");

    if i then

        for j=0,i-1 do
            lp.Print("\t",aa[j]," SLA \n");

-- Compute('id', 'sla_rep', Syear, Smonth, Sday, Eyear, Emonth, Eday, STime, ETime,
RTT_Threshold, Availability, Performance,'array')

        if lpsla.Compute(a[l], aa[j], 2005, 5, 1, 2006, 6, 30,
os.time{year=2005,month=5,day=1,hour=0}, os.time{year=2006,month=6,day=30,hour=0}, 50
, 90, 90, 'array') then
            lp.Print("\t\tip : ",array.ip,"\n");
            lp.Print("\t\tname : ",array.name,"\n");
            lp.Print("\t\tpolling_type : ",array.polling_type,"%\n");
            lp.Print("\t\tperiode : ",array.periode,"%\n");
            lp.Print("\t\tavailability : ",array.availability,"%\n");
            lp.Print("\t\tperformance : ",array.performance,"%\n");
            lp.Print("\t\tgood_polling : ",array.good_polling,"\n");
            lp.Print("\t\ttotal_collected : ",array.total_collected,"\n");
            lp.Print("\t\ttotal_waited : ",array.total_waited,"\n");
            end
        end
    end
end
end
```

# LoriotPro V5.00 Extended Edition – Supervision et gestion SNMP par scripts LUA

## Introduction au System Expert et aux scripts

The screenshot shows the LoriotPro Script Editor and Trace Windows interface.

**Script Editor:**

```

37 --Compute('id','sla_rep',Syear,Smonth,Sday,Eyear,Emonth,Eday,STime,ETime,RTT_T
38 - if lpsla.Compute(a[1],aa[j],2005,5,1,2006,6,30,os.time{year=2005,m
39 lp.Print("\t\tip : ",array.ip,"\n");
40 lp.Print("\t\tname : ",array.name,"\n");
41 lp.Print("\t\tpolling_type : ",array.polling_type,"%\n");
42 lp.Print("\t\tperiode : ",array.periode,"%\n");
43 lp.Print("\t\tavaibility : ",array.avaibility,"%\n");
44 lp.Print("\t\tperformance : ",array.performance,"%\n");
45 lp.Print("\t\tgood_polling : ",array.good_polling,"%\n");
46 lp.Print("\t\ttotal_collected : ",array.total_collected,"%\n");
47 lp.Print("\t\ttotal_waited : ",array.total_waited,"%\n");
48 end
49 end
50 end
51 end
52 end
53
54
55

```

**Trace Windows:**

```

1 1 LoriotPro ID
2 1000 LoriotPro ID
3 127.0.0.1 SLA
4 ip : 127.0.0.1
5 name : LoriotPro
6 polling_type : 2%
7 periode : 0.55936819172113%
8 avaibility : 99.990262901655%
9 performance : 91.129503407984%
10 good_polling : 10269
11 total_collected : 10270
12 total_waited : 1836000
13 2.2.2.6 SLA
14 ip : 2.2.2.6
15 name : 2.2.2.6
16 polling_type : 0%

```

Value Name	Value in
lp_host	169.254.45.35
lp_index	
lp_trace	1
lp_oid	
lp_buffer	error
lp_value	0
lp_graph	nil
lp_graph_unit	nil
lp_graph_title	nil

## Tableau de définition de la librairie

Fonctions	Syntaxe simplifié
lpsla.Compute	<code>lpsla.Compute('loriotpro_id', 'sla_rep', Syear, Smonth, Sday, Eyear, Emonth, Eday, STime, ETime, RTT_Threshold, Avaibility, Performance, 'array');</code>
lpsla.GetLoriotProIDList	<code>lpsla.GetLoriotProIDList('array');</code>
lpsla.GetSLAList	<code>lpsla.GetSLAList('ID','array');</code>

## 7 Exemple de code sous forme de fonction LUA

Vous trouverez dans ce chapitre des exemples de fonctions qui vous permettront de vous familiariser avec le langage LUA intégré à LoriotPro. Un ouvrage entier est consacré à des exemples d'utilisation de LUA dans l'environnement LoriotPro, consultez cet ouvrage pour plus d'informations.

### 7.1.1 Gestion de périodes

Cet ensemble de fonctions permet de déterminer des périodes de temps

L'ensemble de ces fonctions retourne 1 si la période est actuelle ou 0 si hors périmètre.

#### 7.1.1.1 function LP\_GetAbsolutePeriodeMin

```
-----  
function LP_GetAbsolutePeriodeMin (Syear,Smonth,Sday,Shour,Smin,Eyear,Emonth,Eday,Ehour,Emin)  
temp=os.date("%*t",os.time());  
if (temp.year >= Syear and temp.year <= Eyear) then  
--lp.Print(temp.year," ",Syear," ",Eyear,"\n");  
    if (temp.month >= Smonth and temp.month <= Emonth) then  
--lp.Print(temp.month," ",Smonth," ",Emonth,"\n");  
        if (temp.day >= Sday and temp.day <= Eday ) then  
--lp.Print(temp.day," ",Sday," ",Eday,"\n");  
            if (temp.hour >= Shour and temp.hour<= Ehour) then  
--lp.Print(temp.hour," ",Shour," ",Ehour," \n");  
                if (temp.min >= Smin and temp.min<= Emin) then  
--lp.Print(temp.min," ",Smin," ",Emin," \n");  
                    return 1;  
                end  
            end  
        end  
    end  
end  
return 0  
end
```

#### 7.1.1.2 function LP\_GetAbsolutePeriodeHour

```
-----  
function LP_GetAbsolutePeriodeHour(Syear,Smonth,Sday,Shour,Eyear,Emonth,Eday,Ehour)  
temp=os.date("%*t",os.time());  
if (temp.year >= Syear and temp.year <= Eyear) then  
    if (temp.month >= Smonth and temp.month <= Emonth) then  
        if (temp.day >= Sday and temp.day <= Eday ) then  
            if (temp.hour >= Shour and temp.hour<= Ehour) then  
                return 1;  
            end  
        end  
    end  
end  
return 0  
end
```

#### 7.1.1.3 function LP\_GetAbsolutePeriodeDay

```
-----  
function LP_GetAbsolutePeriodeDay (Syear,Smonth,Sday,Eyear,Emonth,Eday)  
temp=os.date("%t",os.time());  
if (temp.year >= Syear and temp.year <= Eyear) then  
    if (temp.month >= Smonth and temp.month <= Emonth) then  
        if (temp.day >= Sday and temp.day <= Eday ) then  
            return 1;  
        end  
    end  
end  
return 0  
end
```

#### 7.1.1.4 function LP\_GetPeriodeMinute

```
-----  
function LP_GetPeriodeMinute (Smin,Emin)  
temp=os.date("%t",os.time());  
if (temp.min >= Smin and temp.min <= Emin) then  
--lp.Print(temp.min," ",Smin," ",Emin,"\n");  
    return 1;  
end  
return 0  
end
```

#### 7.1.1.5 function LP\_GetPeriodeHour

```
-----  
function LP_GetPeriodeHour(Shour,Smin,Ehour,Emin)  
temp=os.date("%t",os.time());  
if (temp.hour >= Shour and temp.hour <= Ehour) then  
--lp.Print(temp.hour," ",Shour," ",Ehour,"\n");  
    if (temp.min >= Smin and temp.min <= Emin) then  
--lp.Print(temp.min," ",Smin," ",Emin,"\n");  
        return 1;  
    end  
end  
return 0  
end
```

#### 7.1.1.6 function LP\_ComputeUpTime

Cette fonction convertit la valeur fournie par des objets SNMP tel que sysuptime en chaîne de caractères.

```
-----  
function LP_ComputeUpTime(value)  
    if value==nil then return "Error" end  
--utilise librairie de gestion d'entiers 64bist  
    value=lp.ui64div(value,100); -- passe en secondes  
  
--86400 secondes par jour  
    days=lp.ui64div(value,86400);  
    reste=lp.ui64rest(value ,86400);  
    -- lp.Print(" value ",value," days ",days," reste ",reste,"\n");  
    if (reste <= 0 ) then
```

```
        return (string.format("%.0f days 0 hours 0 minutes 0 secondes",days))
    end
--1 hours = 3600 seconde
hours=lp.ui64div(reste,3600);
reste=lp.ui64rest(reste,3600);
if (reste < 0 ) then
    return (string.format("%.0f days %.0f hours 0 minutes 0 secondes",days,hours))
end
--1 minutes = 60 secondes
minutes=lp.ui64div(reste,60);
secondes=lp.ui64rest(reste,60);
if (secondes < 0 ) then
    return (string.format("%.0f days %.0f hours %.0f minutes 0 secondes",days,hours,minutes))
end

return (string.format("%.0f days %.0f hours %.0f minutes %.3f secondes",days,hours,minutes,secondes));end
```

## 7.1.2 Analyse de ressource SNMP

Ces fonctions permettent d'analyser des ressources SNMP d'un équipement ; les paramètres en entrée sont l'adresse IP de l'équipement, un numéro d'événement et un niveau de gravité (level) qui sera utilisé pour envoyer un événement à LoriotPro si un certain seuil est atteint.

### 7.1.2.1 function LP\_AuditStorage

Cette fonction permet d'auditer la taille disque d'un équipement et d'envoyer un événement en cas de dépassement d'un seuil.

```
function LP_AuditStorage(ip,event,level,file)

if ip==nil then return 0 end
lp.Print("Audit Storage for : ",ip,"\n")
uptime,buffer=lp.Get(ip,"sysuptime");
if uptime==nil then
lp.Print("Host no repond\n");
return 0;
end
--HOST-MIB
i,obj,objname=lp.GetNext(ip,"hrstorageused.0");
-- lp.Print("out 1 ",i," ",obj," ",objname,"\n");
if obj~="Nul" then
-- test disk usage

i = lp.GetRows(ip,"hrstorageused,hrstoragesize,hrstorageallocationunits,hrstoragedescr","a");

if i then lp.Print("List disk usage\n");

for j=0,(i-1) do

n1=a["hrstorageused-..j"];
n2=a["hrstoragesize-..j"];
n3=a["hrstorageallocationunits-..j"];

-- lp.Print(n1," ",n2," ",n3," ", "\n");
if (n2~="0" and n1 and n2) then
pourcent=(n1/n2)*100;
used=n3 * n1;
max=n3 * n2 ;
lp.Print("\tindex[",a["l-..j]," ",a["hrstoragedescr-..j"],"]\t",string.format("%.2f%% %.0f/%.0f
Mo\n",pourcent,(used/1024)/1024,(max/1024)/1024));
```

```
        if (pourcent>70) then
            alarm=string.format("WARNING Host [%s] AS Low disk %s : Reached %.2i%% %.0f/.0f
Mo",ip,a["hrstoragedescr-.j"],pourcent,(used/1024)/1024,(max/1024)/1024);
            lp.Print(alarm,"\\n");
            if event >0 then lp.SendEvent( event,level,ip,"255.255.255.255",alarm) end
        end
    end
return 1;
end
end
```

### 7.1.2.2 function LP\_AuditSystem

```
function LP_AuditSystem(ip,event,level,tab,file)
number=lp.Gets(ip,"sysdescr,sysobjectid,sysuptime,syscontact,sysname,syslocation,sysservices,ipForwarding","aa
");
uptime,buffer=lp.Get(ip,"sysuptime");
lp.Print("Audit system Information for : ",ip," ",number,"\\n")

if number~=8 then
lp.Print("Host no respond or error\\n");
return 0;
end
--last=(aa["sysuptime.0"])*1;

lp.Print(string.format("Host Informations \\nName [%s]\\nLocation [%s]\\nContact [%s]\\nUptime [%s]\\nDescription
[%s]\\nSysObjectID [%s]\\nSysServices [%s]\\nIpForwarding [%s]\\n"
,aa["sysname.0"]
,aa["syslocation.0"]
,aa["syscontact.0"]
,ComputeUpTime(uptime)
,aa["sysdescr.0"]
,aa["sysobjectid.0"]
,aa["sysservices.0"]
,aa["ipforwarding.0"]
));

tab["name"]=aa["sysname.0"];
tab["objectid"]=aa["sysobjectid.0"];
tab["services"]=aa["sysservices.0"];
tab["uptime"]=uptime;
tab["forwarding"]=aa["ipforwarding.0"];
return 1;
end
```

### 7.1.2.3 function LP\_AuditSoftwareInstalled

```
function LP_AuditSoftwareInstalled(ip,event,level,file)
if ip==nil then return 0 end
```

```
lp.Print("Audit Software Installed for : ",ip,"\\n")
uptime,buffer=lp.Get(ip,"sysuptime");
if uptime==nil then
    lp.Print("Host no repond\\n");
    return 0;
end
i = lp.GetRows(ip,"hrswinstalledname,hrswinstalledtype,hrswinstalleddate","a");
if i then lp.Print("List installed Softwares\\n")
    for j=0,(i-1) do

        lp.Print("\\tindex[",a["l-..j]," ]\\t\\t",string.format("%s [%s] [%s]\\n",a["hrswinstalledname-"
"..j],a["hrswinstalledtype-..j],a["hrswinstalleddate-..j]));
    end
end
end
```

#### 7.1.2.4 function LP\_AuditDriverInstalled

```
-----  
function LP_AuditDriverInstalled(ip,event,level,file)

if ip==nil then return 0 end
lp.Print("Audit Driver Installed for : ",ip,"\\n")
uptime,buffer=lp.Get(ip,"sysuptime");
if uptime==nil then
    lp.Print("Host no repond\\n");
    return 0;
end
i = lp.GetRows(ip,"hrdevicetype,hrdevicedescr,hrdevicestatus","a");
if i then lp.Print("List Driver\\n") end ;
    for j=0,(i-1) do

        lp.Print("\\tindex[",a["l-..j]," ]\\t\\t",string.format("%s [%s] [%s]\\n",a["hrdevicetype-..j],a["hrdevicedescr-"
"..j],a["hrdevicestatus-..j]));
    end
end
```

#### 7.1.2.5 function LP\_AuditRouteTable

```
-----  
function LP_AuditRouteTable(ip,event,level,tab,file)

if ip==nil then return 0 end
lp.Print("Audit Routing Table for : ",ip,"\\n")
uptime,buffer=lp.Get(ip,"sysuptime");
if uptime==nil then
    lp.Print("Host no repond\\n");
    return 0;
end
i =
lp.GetRows(ip,"iproutedest,iprouteifindex,iproutemetric1,iproutemetric2,iproutemetric3,iproutemetric4,iproutenexthop,iproutetype,iprouteproto,iprouteage,iproutemask","a");
if i~=nil then lp.Print("List Routes(.i,.)\\n") end ;
    for j=0,(i-1) do
if a["iproutedest-..j"]=="0.0.0.0" then tab["gw"]=a["iproutenexthop-..j"] end

        lp.Print("\\t",string.format("Dest[%s]\\t\\tMask[%s]\\t\\tPolicy[%s]\\tNextHop[%s]\\tIndex[%s]\\tType[%s]\\tProto[%s]\\tAge[%s]\\tMetric1[%s]\\tMetric2[%s]\\tMetric3[%s] \\n"
,a["iproutedest-..j],a["iproutemask-..j],a["iproutetype-..j]
,a["iproutenexthop-..j],a["iprouteifindex-..j],a["iproutetype-..j]
,a["iprouteproto-..j],a["iprouteage-..j],a["iproutemetric1-..j]
```

```
,a["iproutemetric2-..j],a["iproutemetric3-..j));  
        end  
    end
```

### 7.1.2.6 function LP\_AuditDot3Stat

```
-----  
function LP_AuditDot3Stat(ip,index,level,file)  
req=string.format("dot3statsalignmenterrors.%s,dot3statsfcerrors.%s,dot3statssinglecollisionframes.%s,dot3stats  
multiplecollisionframes.%s,dot3statssqtesterrors.%s,dot3statsdeferredtransmissions.%s,dot3statslatecollisions.%s,  
,dot3statsexcessivecollisions.%s,dot3statsinternalmactransmiterrors.%s,dot3statscarriersenseerrors.%s,dot3statsfr  
ametoolongs.%s,dot3statsinternalmacreceiveerrors.%s"  
,index,index,index,index,index,index,index  
,index,index,index,index);  
--Ip.Print(ip," ",req,"\n");  
rcv=lp.Gets(ip,req,"aa");  
  
if (rcv) then  
    --Ip.Print("\t\t\t Dot 3 errors for this port : \n");  
    if (aa["dot3statsalignmenterrors.."..index]~="0" and aa["dot3statsalignmenterrors.."..index]~=nil) then  
        lp.Print("\t\t\t","Alignment Errors(",string.format("%s",aa["dot3statsalignmenterrors.."..index]),")\n") end  
        if aa["dot3statsfcerrors.."..index]~="0" and aa["dot3statsfcerrors.."..index]~=nil then lp.Print("\t\t\t","FSC  
Errors(",string.format("%s",aa["dot3statsfcerrors.."..index]),")\n") end  
        if aa["dot3statssinglecollisionframes.."..index]~="0" and aa["dot3statssinglecollisionframes.."..index]~=nil then  
lp.Print("\t\t\t","Single Collision Frames(",string.format("%s",aa["dot3statssinglecollisionframes.."..index]),")\n") end  
        if aa["dot3statsmultiplecollisionframes.."..index]~="0" and aa["dot3statsmultiplecollisionframes.."..index]~=nil  
then lp.Print("\t\t\t","Multiple Collision  
Frames(",string.format("%s",aa["dot3statsmultiplecollisionframes.."..index]),")\n") end  
        if aa["dot3statssqtesterrors.."..index]~="0" and aa["dot3statssqtesterrors.."..index]~=nil then  
lp.Print("\t\t\t","SQE Test Errors(",string.format("%s",aa["dot3statssqtesterrors.."..index]),")\n") end  
        if aa["dot3statsdeferredtransmissions.."..index]~="0" and aa["dot3statsdeferredtransmissions.."..index]~=nil then  
lp.Print("\t\t\t","Deferred Transmissions(",string.format("%s",aa["dot3statsdeferredtransmissions.."..index]),")\n") end  
        if aa["dot3statslatecollisions.."..index]~="0" and aa["dot3statslatecollisions.."..index]~=nil then lp.Print("\t\t\t","Late  
Collisions(",string.format("%s",aa["dot3statslatecollisions.."..index]),")\n") end  
        if aa["dot3statsexcessivecollisions.."..index]~="0" and aa["dot3statsexcessivecollisions.."..index]~=nil then  
lp.Print("\t\t\t","Excessive Collisions(",string.format("%s",aa["dot3statsexcessivecollisions.."..index]),")\n") end  
        if aa["dot3statsinternalmactransmiterrors.."..index]~="0" and aa["dot3statsinternalmactransmiterrors.."..index]~=nil then lp.Print("\t\t\t","Internal MAC  
Transmiterrors(",string.format("%s",aa["dot3statsinternalmactransmiterrors.."..index]),")\n") end  
        if aa["dot3statscarriersenseerrors.."..index]~="0" and aa["dot3statscarriersenseerrors.."..index]~=nil then lp.Print("\t\t\t","Carrier Sense Errors(",string.format("%s",aa["dot3statscarriersenseerrors.."..index]),")\n") end  
        if aa["dot3statsframetoolongs.."..index]~="0" and aa["dot3statsframetoolongs.."..index]~=nil then lp.Print("\t\t\t","Frame Too Longs(",string.format("%s",aa["dot3statsframetoolongs.."..index]),")\n") end  
        if aa["dot3statsinternalmacreceiveerrors.."..index]~="0" and aa["dot3statsinternalmacreceiveerrors.."..index]~=nil  
then lp.Print("\t\t\t","Internal MAC Receive  
Errors(",string.format("%s",aa["dot3statsinternalmacreceiveerrors.."..index]),")\n") end  
  
    end  
end
```

### 7.1.2.7 function LP\_AuditCiscoCPU

```
-----  
function LP_AuditCiscoCPU(ip,event,level,threshold,file)  
  
if ip==nil then return 0 end  
lp.Print("Audit Cisco CPU Load for : ",ip,"\\n")  
uptime,buffer=lp.Get(ip,"sysuptime");  
if uptime==nil then  
    lp.Print("Host no respond\\n");  
    return 0;  
end
```

## LoriotPro V5.00 Extended Edition – Supervision et gestion SNMP par scripts LUA

### Introduction au System Expert et aux scripts

```
i = Ip.GetRows(ip,"cpmcputotal5sec,cpmcputotal1min,cpmcputotal5min","a");
if i then Ip.Print("List CPU Usage in %\n")
    for j=0,(i-1) do
        result=string.format("cpmcputotal5sec[%s%%] cpmcputotal1min[%s%%]
cpmcputotal5min[%s%%]",a["cpmcputotal5sec-..j"],a["cpmcputotal1min-..j"],a["cpmcputotal5min-..j"]);
        Ip.Print("\tindex[",a["l-..j"], " ]\t\t",result,"\n");

        if (Ip.atof(a["cpmcputotal5sec-..j])>threshold) or (Ip.atof(a["cpmcputotal1min-..j])>threshold) or
(Ip.atof(a["cpmcputotal5min-..j])>threshold) then
            rep=Ip.GetPIInformation(ip,"aaa");
            if rep then
                alarm=string.format("CPU Load for [%s/%s ] reach %.2f%% (%s)",ip,aaa.name,threshold, result);
            else
                alarm=string.format("CPU Load for [%s/%s ] reach %.2f%% (%s)",ip,aaa.name,threshold, result);
            end
            Ip.SendEvent( event,level,ip,"255.255.255.255",alarm);
        end
    end
end
end
```

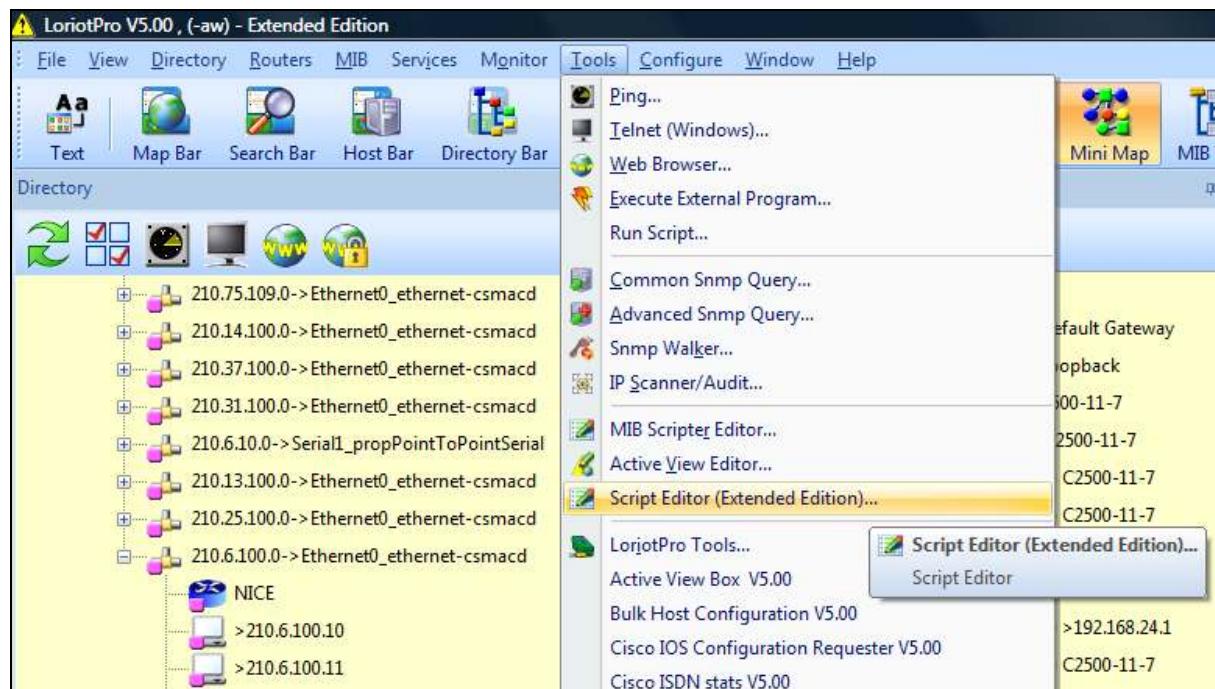
## 8 Environnement de développement intégré

### 8.1 Introduction

LoriotPro V5 « Extended Edition » contient un environnement de développement intégré permettant d'éditer et de tester vos scripts. Le module principal est un éditeur que vous pouvez appeler à partir du menu « Tools » du logiciel. Vos scripts peuvent bien entendu être édités avec votre éditeur de texte préféré, mais ce plugin intègre des fonctions de tests et d'aides en ligne qui vous seront bien utiles pour « déboguer » votre travail. Vous pouvez lancer plusieurs éditeurs dans le cas d'un travail sur plusieurs scripts. Il est possible de développer des scripts sur une plateforme LoriotPro « Extended Edition » active (en production), LoriotPro est multitâche et supporte ce mode de travail coopératif. L'expérience montre que dans le cas de développement important, il est préférable de disposer de deux plateformes LoriotPro, une pour la production et une autre pour la mise au point des scripts. L'intégration du scripting dans LoriotPro est réalisé en mode protégé, ce qui permet une utilisation multitâche des scripts ou des autres parties du logiciel, mais personne n'est à l'abri d'un script mal écrit qui oblige à relancer le logiciel et donc à perturber la production.

### 8.2 Lancement de l'éditeur

L'éditeur se lance à partir de l'option Tools du menu principal du logiciel.



Vous pouvez lancer plusieurs éditeurs pour vous permettre de travailler sur plusieurs fichiers en même temps. Il est possible de faire des copier / coller entre les fenêtres. La fenêtre de l'éditeur n'est pas « Modal » et vous pouvez naviguer dans les autres

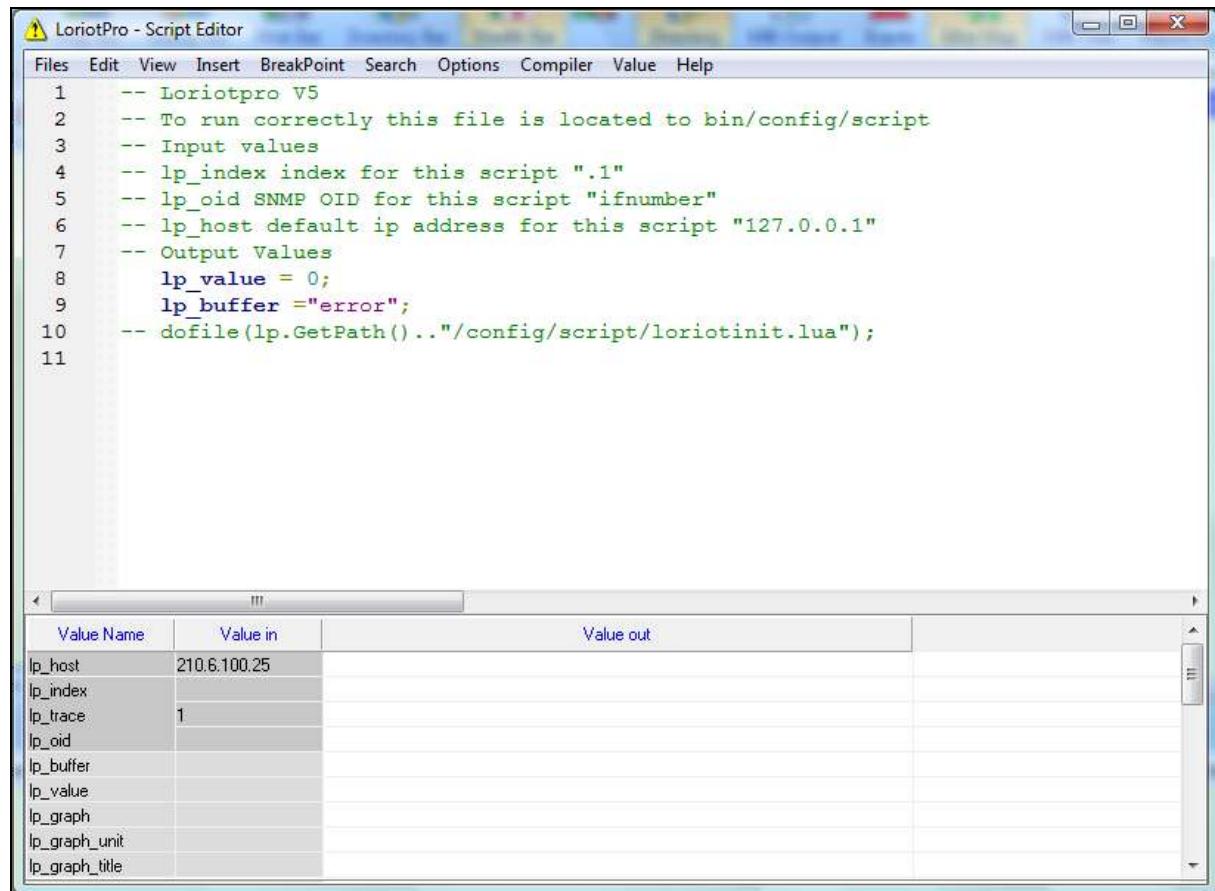
fenêtres ou modules du logiciel en même temps. Ceci vous permet de réaliser des tests ou contrôles sur les éléments de la Directory durant votre développement ou bien de lancer des tâches en parallèle.

**Attention** 

Vous pouvez créer des scripts et faire des tests dans votre environnement de travail, mais, comme dans tout processus de développement, vous pouvez faire des erreurs et « crasher » LoriotPro. Il est grandement préférable de réaliser vos scripts sur une plateforme de développement plutôt que sur votre plateforme de production. Une fois vos scripts débogués, vous pourrez les utiliser en production.

L'éditeur est composé de deux zones :

1. La zone d'édition du script
2. La zone d'édition des variables



### 8.3 Zone d'édition du script

Cette zone contient le menu de l'éditeur et la zone d'édition du script.

The screenshot shows the LoriotPro Script Editor window. The title bar says "LoriotPro - Script Editor". The menu bar includes Files, Edit, View, Insert, BreakPoint, Search, Options, Compiler, Value, and Help. The code area contains the following Lua script:

```
1 -- Loriotpro V5
2 -- To run correctly this file is located to bin/config/script
3 -- Input values
4 -- lp_index index for this script ".1"
5 -- lp_oid SNMP OID for this script "ifnumber"
6 -- lp_host default ip address for this script "127.0.0.1"
7 -- Output Values
8     lp_value = 0;
9     lp_buffer ="error";
10    -- dofile(lp.GetPath().."config/script/loriotinit.lua");
11
```

### 8.3.1 Coloration

La zone d'édition du script utilise la librairie Scintilla <http://www.scintilla.org> pour colorer le texte, l'ensemble des mots des librairies LoriotPro y apparaît en « bleu gras ».

The screenshot shows the LoriotPro Script Editor window with the title "LoriotPro - Script Editor : C:\Program Files\LUTEUS\LoriotPro V5\bin\config\lua\_editor.ini". The menu bar is identical to the previous screenshot. The code area contains the following configuration file:

```
1 ######
2 # LoriotPro V5 lua editor init file V1.8
3 # (c) 2002-2008 Luteus SARL
4 #####
5 [HELP_TIPS]
6 lp_host "The input ip address of the default host"
7 lp_oid "The oid input value"
8 lp_index "The index imput value"
9 lp_value "one return value (double)&for snmp kernel"
10 lp_buffer "one return value (string)&for snmp kernel"
11 lp_graph "1 if audit script return graph value"
12 lp_graph_type "0 if counter 1 if gauge"
13 lp_graph_value "the value returned par audit script"
```

Le fichier `lua_editor.ini` situé dans le répertoire `bin/config` du logiciel contient l'ensemble des informations de coloration de l'éditeur. Il est possible de modifier ce fichier pour intégrer vos propres mots ou noms de fonction dans vos scripts. Ce fichier est chargé dynamiquement à chaque lancement d'un éditeur, il n'est donc pas nécessaire de relancer le logiciel pour profiter de cette option.

*License for Scintilla and SciTE*

*Copyright 1998-2003 by Neil Hodgson <neilh@scintilla.org>*

*All Rights Reserved*

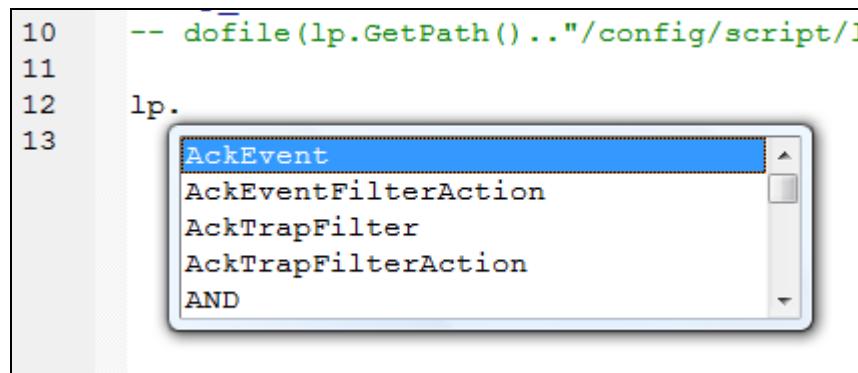
*Permission to use, copy, modify, and distribute this software and its*

*documentation for any purpose and without fee is hereby granted,  
provided that the above copyright notice appear in all copies and that  
both that copyright notice and this permission notice appear in  
supporting documentation.*

*NEIL HODGSON DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS  
SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY  
AND FITNESS, IN NO EVENT SHALL NEIL HODGSON BE LIABLE FOR ANY  
SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES  
WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS,  
WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER  
TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE  
OR PERFORMANCE OF THIS SOFTWARE.*

### 8.3.2 Complétion

La zone de texte intègre des fonctions de complétion sur les mots réservés des librairies. Lorsque vous tapez les premières lettres d'une fonction de type classe, une listbox vous propose les options de langage disponibles. Le fichier *lua\_editor.ini* situé dans le répertoire *bin/config* du logiciel contient l'ensemble des informations de complétion. Il est possible de modifier ce fichier pour intégrer vos propres mots ou noms de classe dans vos scripts. Ce fichier est chargé dynamiquement à chaque lancement d'un éditeur, il n'est donc pas nécessaire de relancer le logiciel pour profiter de cette option.



### 8.3.3 Tooltips

La zone de texte intègre des fonctions de « tooltips » sur les mots réservés des librairies. Lorsque vous tapez la parenthèse d'une fonction de type classe, un « tooltips » vous fournit une aide concernant les variables associées à la fonction. Le fichier *lua\_editor.ini* situé dans le répertoire *bin/config* du logiciel contient l'ensemble des informations de « tooltips ». Il est possible de modifier ce fichier pour intégrer vos propres « tooltips » dans vos scripts. Ce fichier est chargé dynamiquement à chaque lancement d'un éditeur, il n'est donc pas nécessaire de relancer le logiciel pour profiter de cette option.

Vous pouvez modifier ce fichier pour intégrer vos références de librairie, mais il risque d'être écrasé lors d'une mise à jour du logiciel. Il est préférable d'utiliser le fichier :

***bin/config/lua\_editor\_custom.ini***

qui lui ne sera pas touché par une mise à jour.

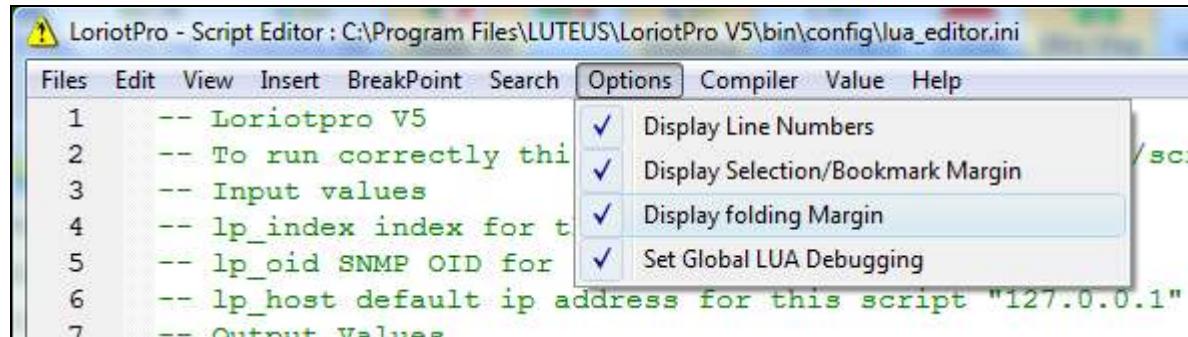
```
- lp.AckEvent(  
    lp.AckEvent(number)  
    nil if fail
```

```
lp.ascii_ui64sub(  
    lp.ascii_ui64sub('value1' , 'value2');  
    'result'  
    value  
    nil if fail
```

Si vous laissez la souris sur une fonction connue de l'éditeur, un « tooltips » apparaît.

### 8.3.4 Numerotation des lignes

Vous pouvez afficher ou non les numéros de ligne sur la gauche de la zone d'édition.



### 8.3.5 Gestion des encapsulations de code

La zone de gauche supporte la gestion de l'encapsulation de la structure de votre code. Le signe ‘-’ à gauche du code indique que vous pouvez simplifier la visualisation en cachant du texte, le signe ‘+’ que vous pouvez ouvrir le texte caché.

```
19 net,mask=lp.GetFirstNetwork();  
20 i=1;  
21 - if i==1 then  
22 - while net~=nil do  
23         lp.Trace(net..mask);  
24 -     if (lp.GetNetworkInformation(net,mask,"a")) t  
25             --lp.Trace(a.name..net..mask);  
26             --if (a.nb_router>0) then lpav.Insert  
27             if (a.nb_host>0) then lpav.InsertObj  
28             lpav.InsertObject(LPAV_OBJ_NETWORK,ne  
29         end  
30     net,mask=lp.GetNextNetwork(net,mask);  
31 end  
32 end  
--
```

Avant

```
19 net,mask=lp.GetFirstNetwork();  
20 i=1;  
21 - if i==1 then  
22 + while net~=nil do  
32     end  
33  
34 lpav.RefreshMap();  
35
```

Après

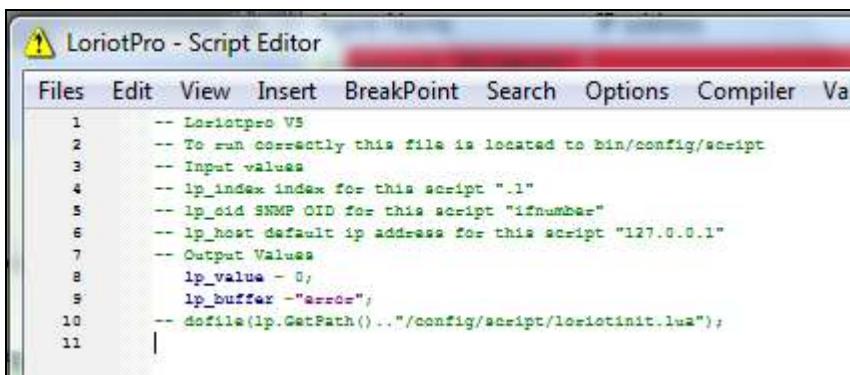
### 8.3.6 Sélection de texte

La zone de gauche permet de sélectionner des lignes de texte, en vue d'un copier / coller .

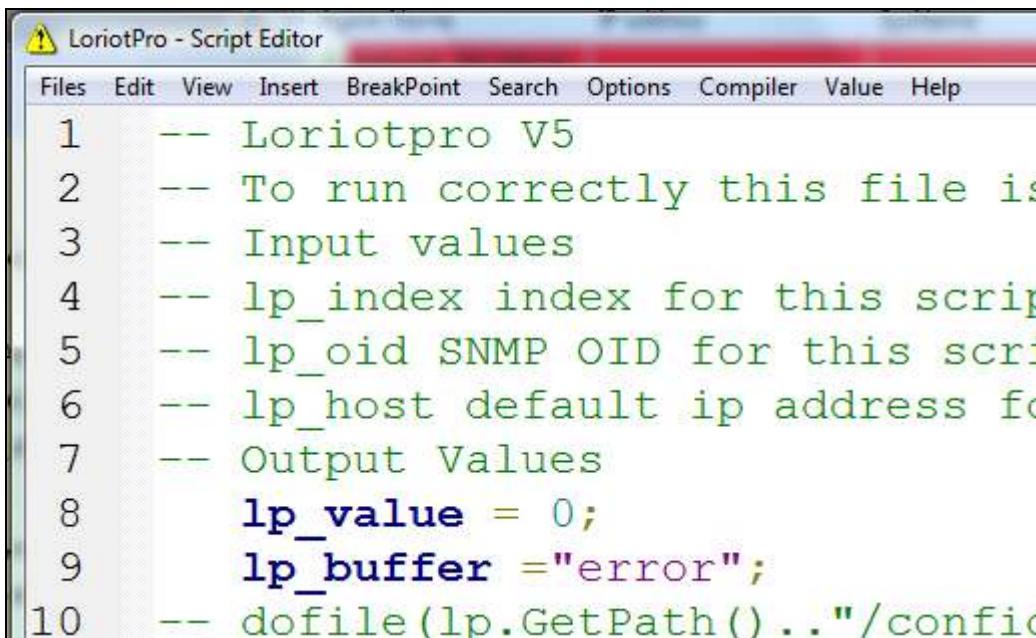
```
34 lpav.RefreshMap();  
35  
36 router=lp.GetFirstRouter();  
37 - while router~=nil do [REDACTED]  
38     --lp.Trace("router >..net");  
39 -     if lp.GetRouterInformation(router,"a") ==1 then  
40         lp.Trace("router >..a.name..router.." ..a.x.." ..a.y,"\\n");  
41         lpav.InsertObject(LPAV_OBJ_ROUTER,a.name, a.x,a.y,a.UID);  
42     [REDACTED]
```

### 8.3.7 Taille des caractères

Vous pouvez utiliser la combinaison touche [CTRL] + molette de souris pour modifier dynamiquement la taille des caractères de la fenêtre de texte.



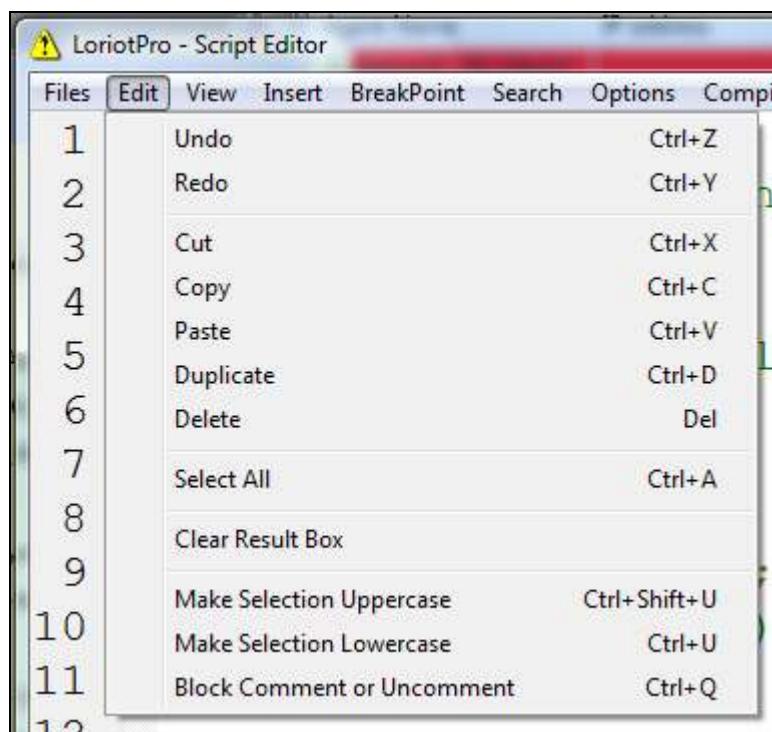
```
1 -- Loriotpro V5
2 -- To run correctly this file is located to bin/config/script
3 -- Input values
4 -- lp_index index for this script ".1"
5 -- lp_oid SNMP OID for this script "ifnumber"
6 -- lp_host default ip address for this script "127.0.0.1"
7 -- Output Values
8     lp_value = 0;
9     lp_buffer ="error";
10    -- dofile(lp.GetPath() .. "/config/script/loriotinit.lua");
11
```



```
1 -- Loriotpro V5
2 -- To run correctly this file is
3 -- Input values
4 -- lp_index index for this scrip
5 -- lp_oid SNMP OID for this scri
6 -- lp_host default ip address fo
7 -- Output Values
8     lp_value = 0;
9     lp_buffer ="error";
10    -- dofile(lp.GetPath() .. "/config
```

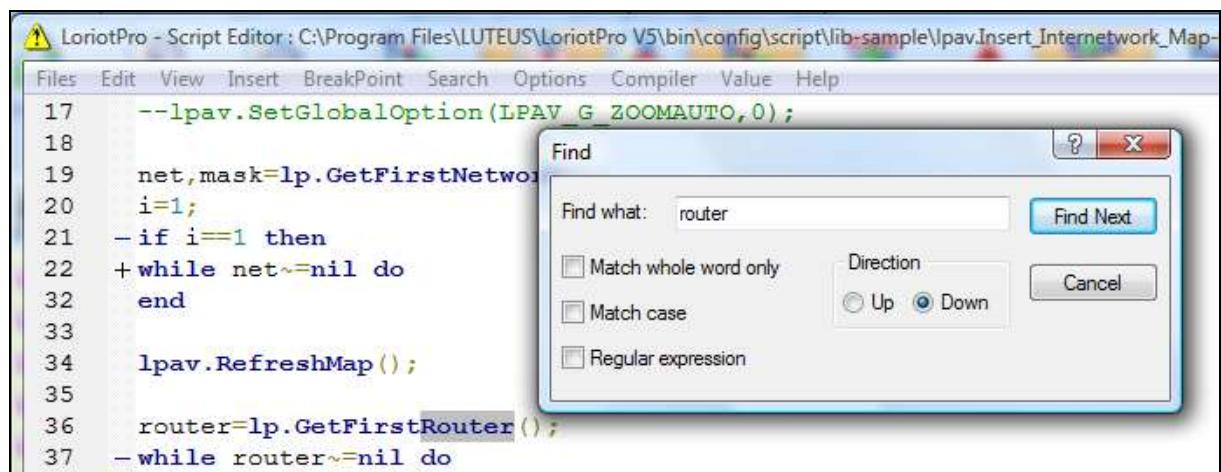
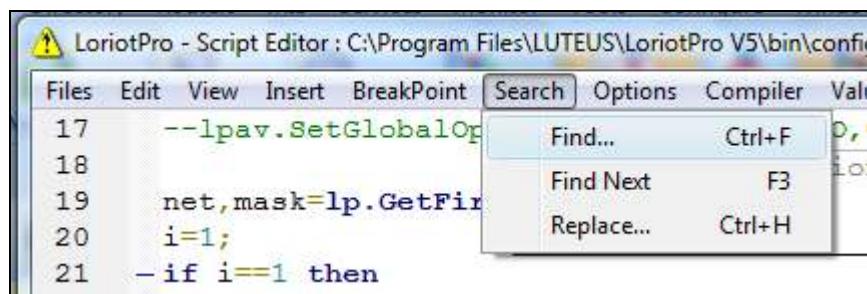
#### 8.3.7.1 Options - Copier / coller ‘undo redo’

L'éditeur supporte toutes les fonctions de copier / coller et de ‘undo’ classique en respectant les raccourcis clavier standard.

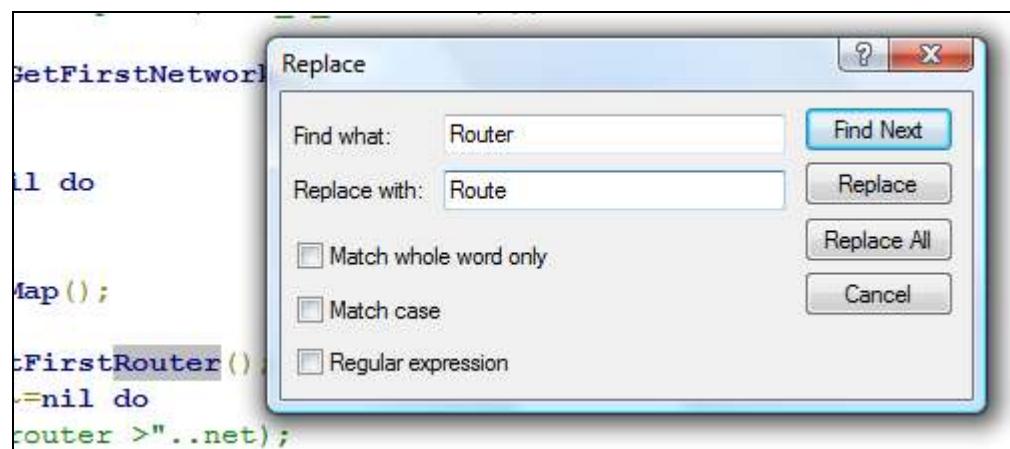


### 8.3.8 Search Replace

Des fonctions de recherche de portions de texte et de remplacements classiques sont disponibles.



## Replace

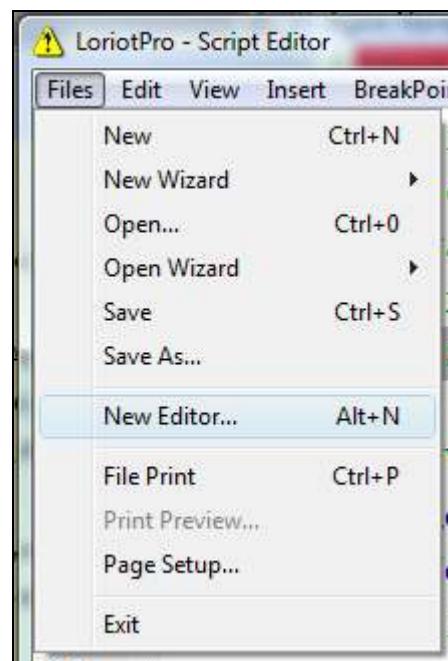


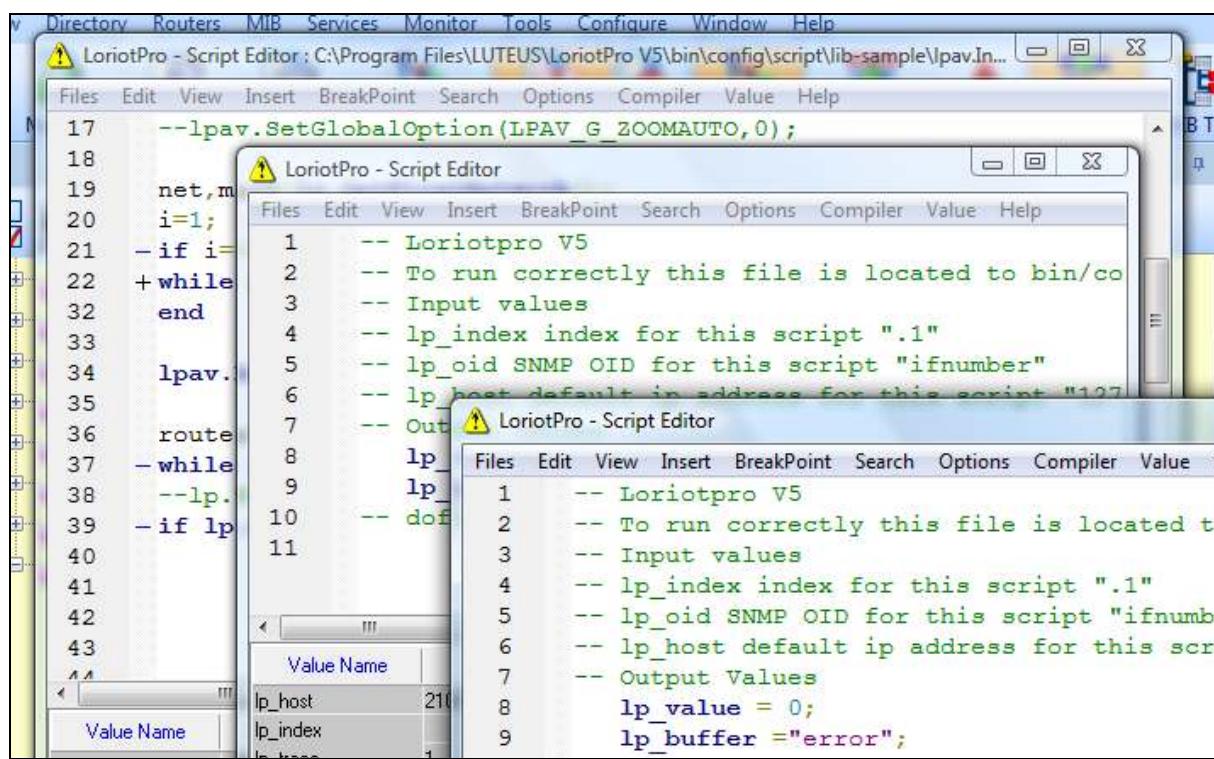
### 8.3.9 Mode multi-pages

Il est possible de lancer plusieurs éditeurs et de réaliser des opérations de copier / coller entre les fenêtres.

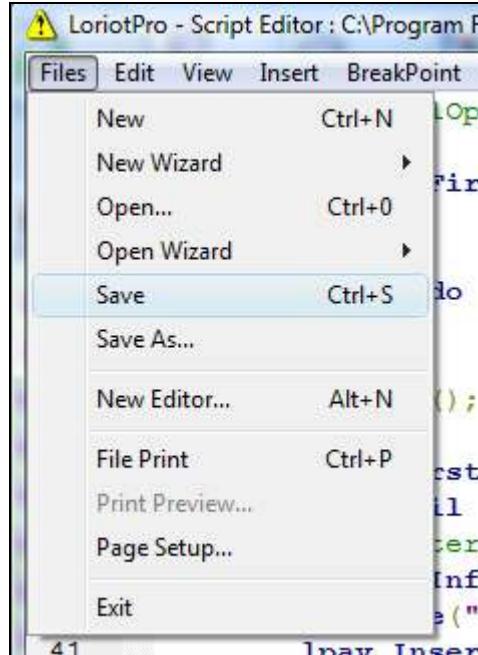
#### Attention 💣

Chaque nouvel éditeur possède son interpréteur et ses fenêtres de mise au point.





### 8.3.10 Menu files



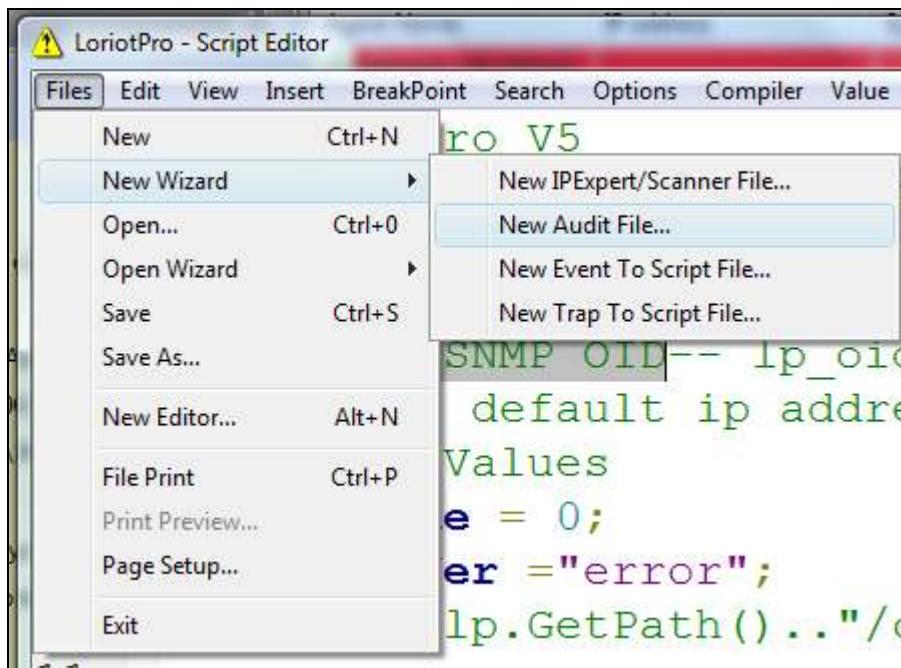
#### 8.3.10.1 Options - New

Elles vous proposent la création d'un nouveau script.

Vous pouvez importer / exporter vos scripts avec le menu Files, les fichiers de script ont l'option « .lua » et sont situés dans l'arborescence bin/config/script du logiciel.

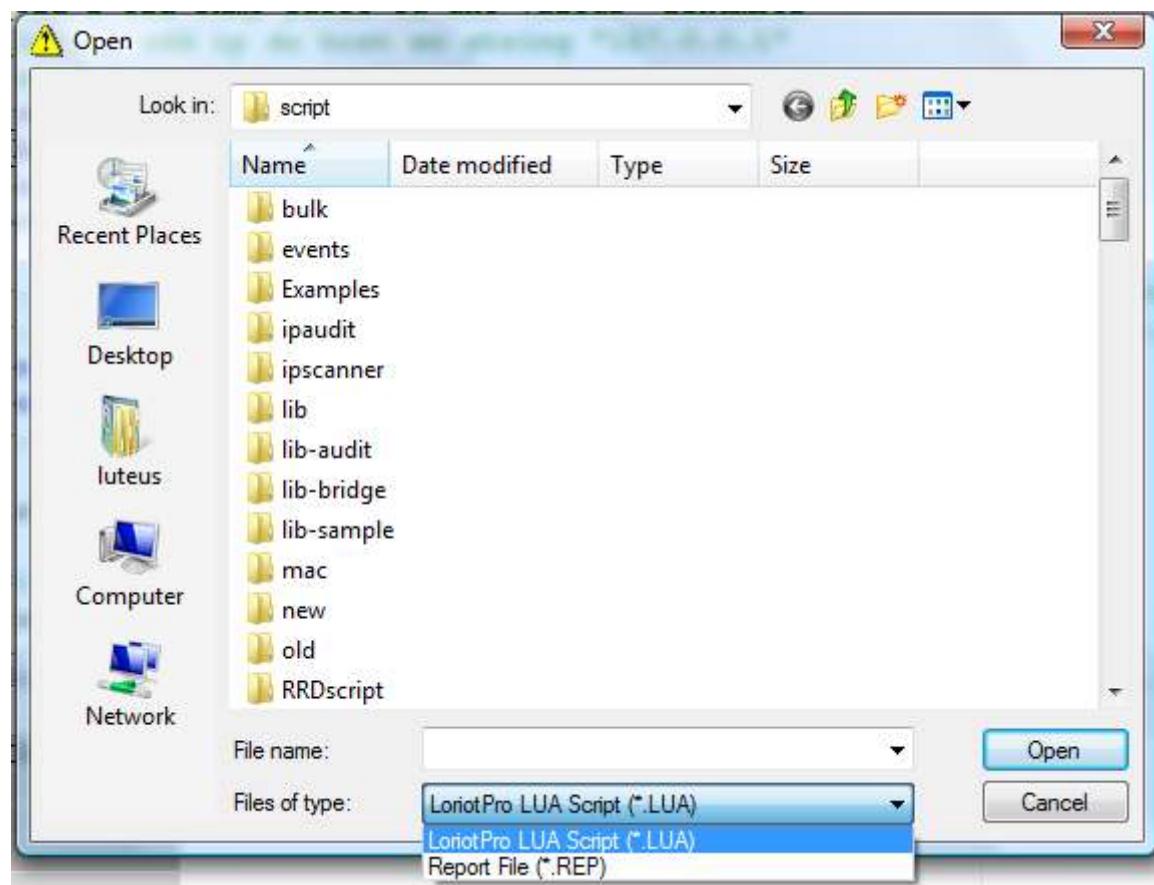
### 8.3.10.2 Options - New Wizard

Devant le nombre important de types de scripts supportés par LoriotPro V5 « Extended Edition », différents Wizards de création ou de chargement de fichiers sont disponibles. Les différents types de fichiers de scripts associés à des modules dédiés seront documentés dans les paragraphes traitant des modules concernés.



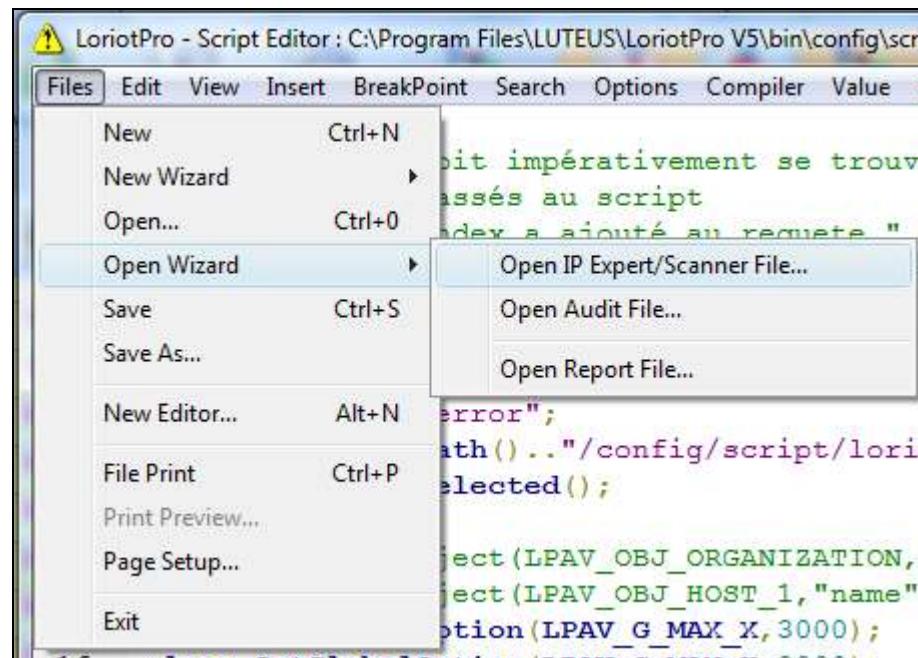
### 8.3.10.3 Options - Open

Elles ouvrent un fichier LUA de script ou un fichier Rep de génération de rapport.



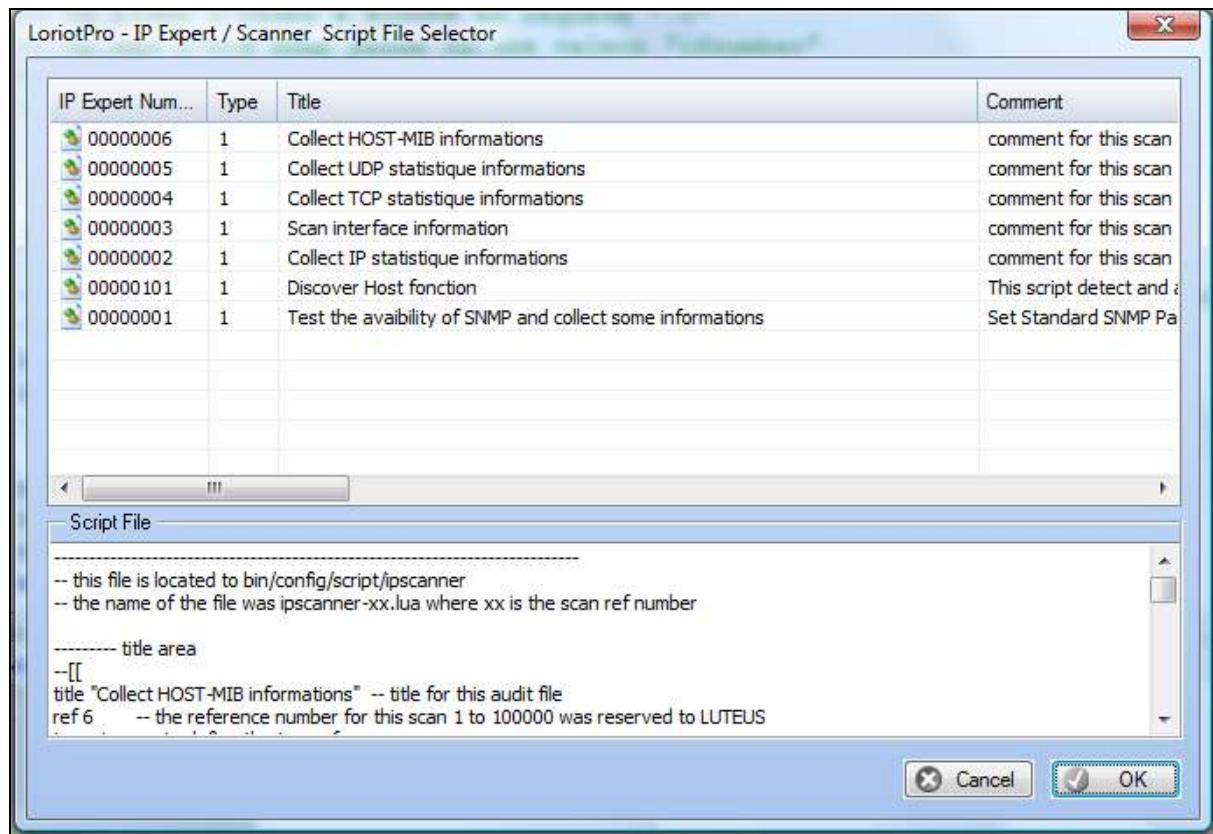
#### 8.3.10.4 Options - Open Wizard

Certains fichiers de scripts sont de type managés et disposent d'un titre et de paramètres, les options « Open Wizard » vous aide dans le choix du bon fichier.



### 8.3.10.5 Options - Open IP Expert/Scanner File

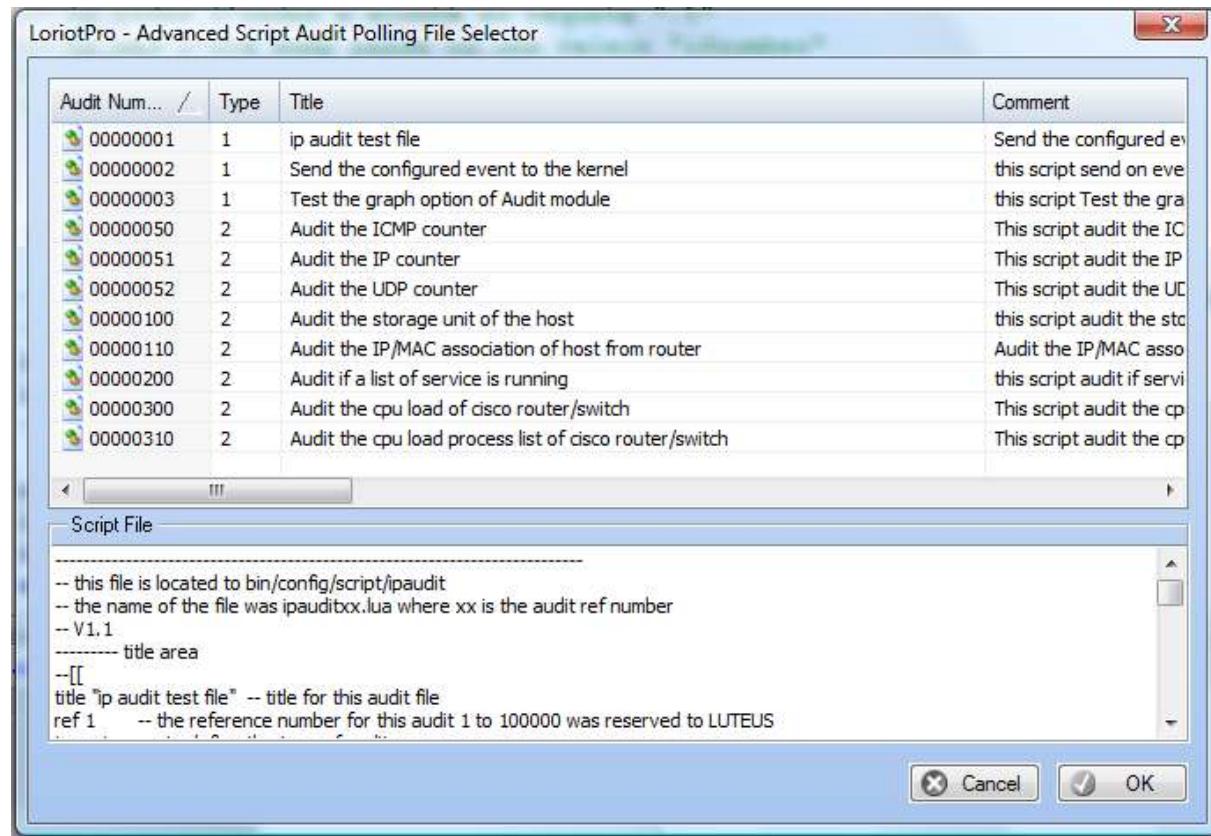
Cette option ouvre la fenêtre de sélection des scripts du module IP Expert.



Ces fichiers de scripts ont l'option « .lua » et sont situés dans l'arborescence *bin/config/script/IPscanner* du logiciel.

### 8.3.10.6 Options - Open Audit File

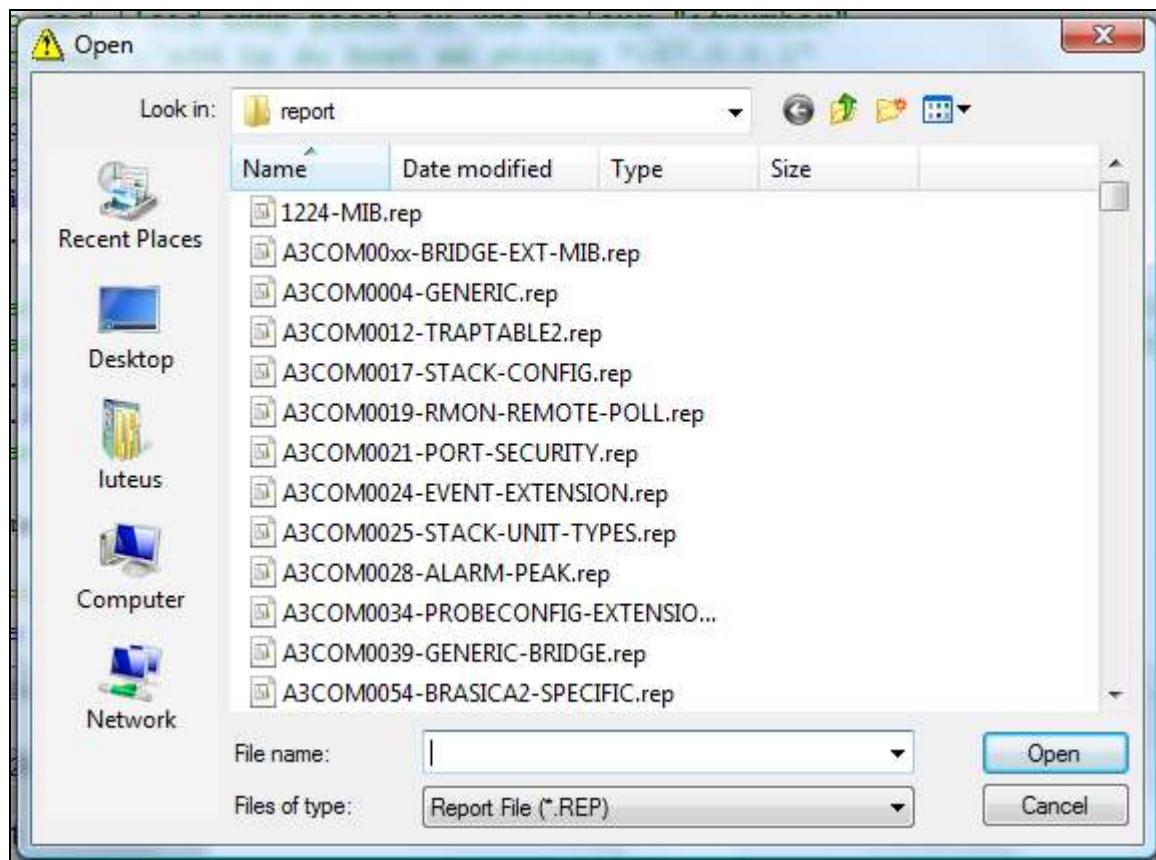
Cette option ouvre la fenêtre de sélection des scripts du module d'Audit.



Ces fichiers de scripts ont l'option « .lua » et sont situés dans l'arborescence *bin/config/script/IPscanner* du logiciel.

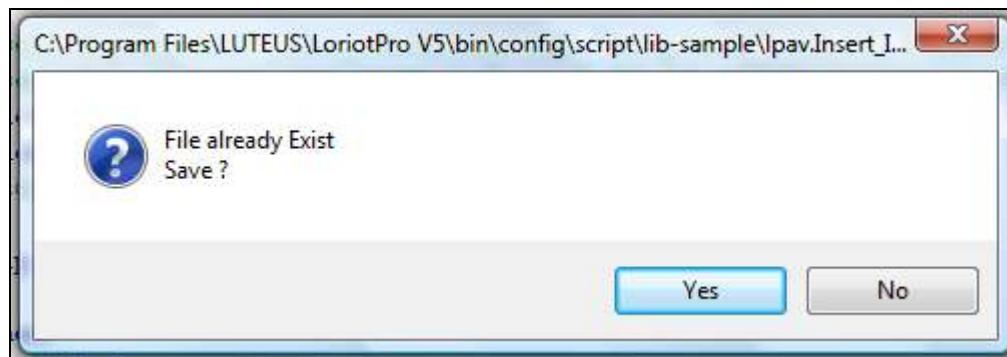
### 8.3.10.7 Options - Open Report File

L'éditeur peut être utilisé pour éditer et tester des fichiers de génération de rapports. La plupart des fichiers de génération de rapports sont situés dans le répertoire *bin/report* du logiciel mais ils peuvent être utilisés par différents modules et placés dans des répertoires dédiés pour effectuer différentes tâches.



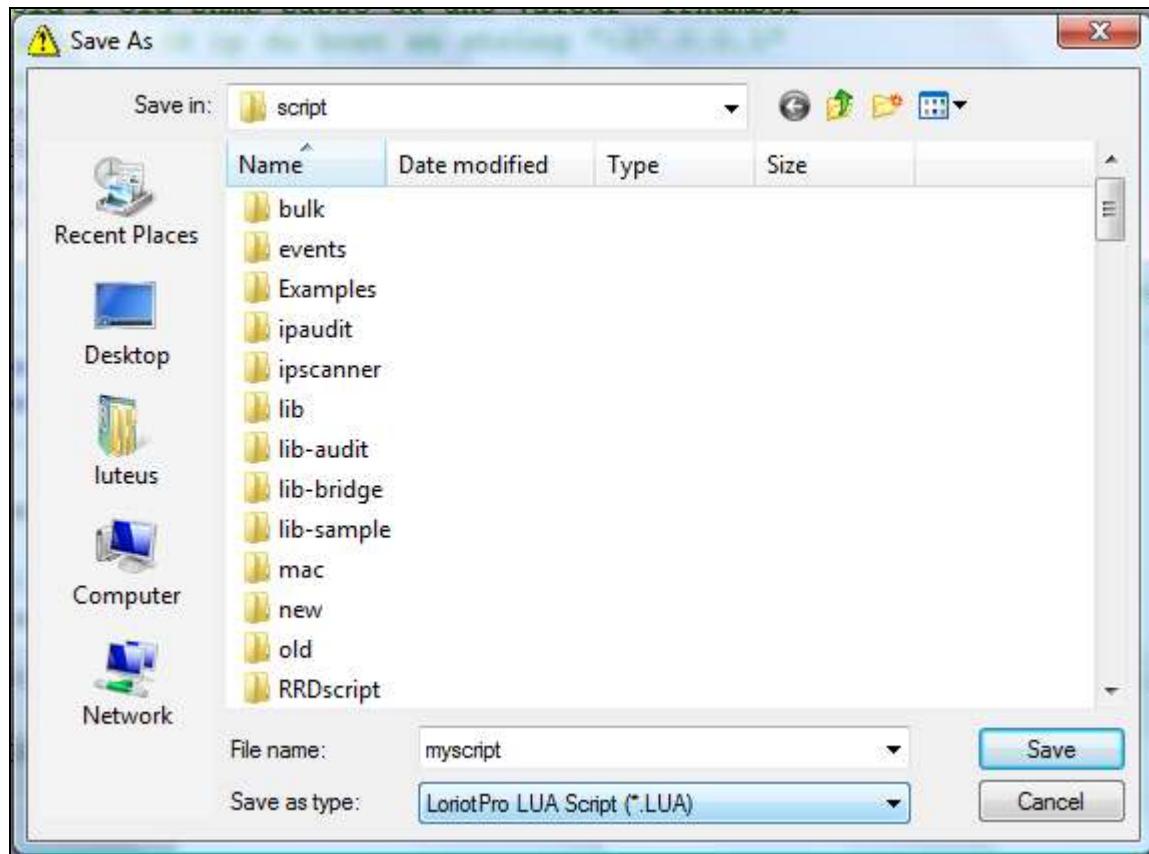
### 8.3.10.8 Options - Save

Elle permet la sauvegarde du script en cours ; il est fort probable qu'une fenêtre vous informe que le fichier existe déjà.



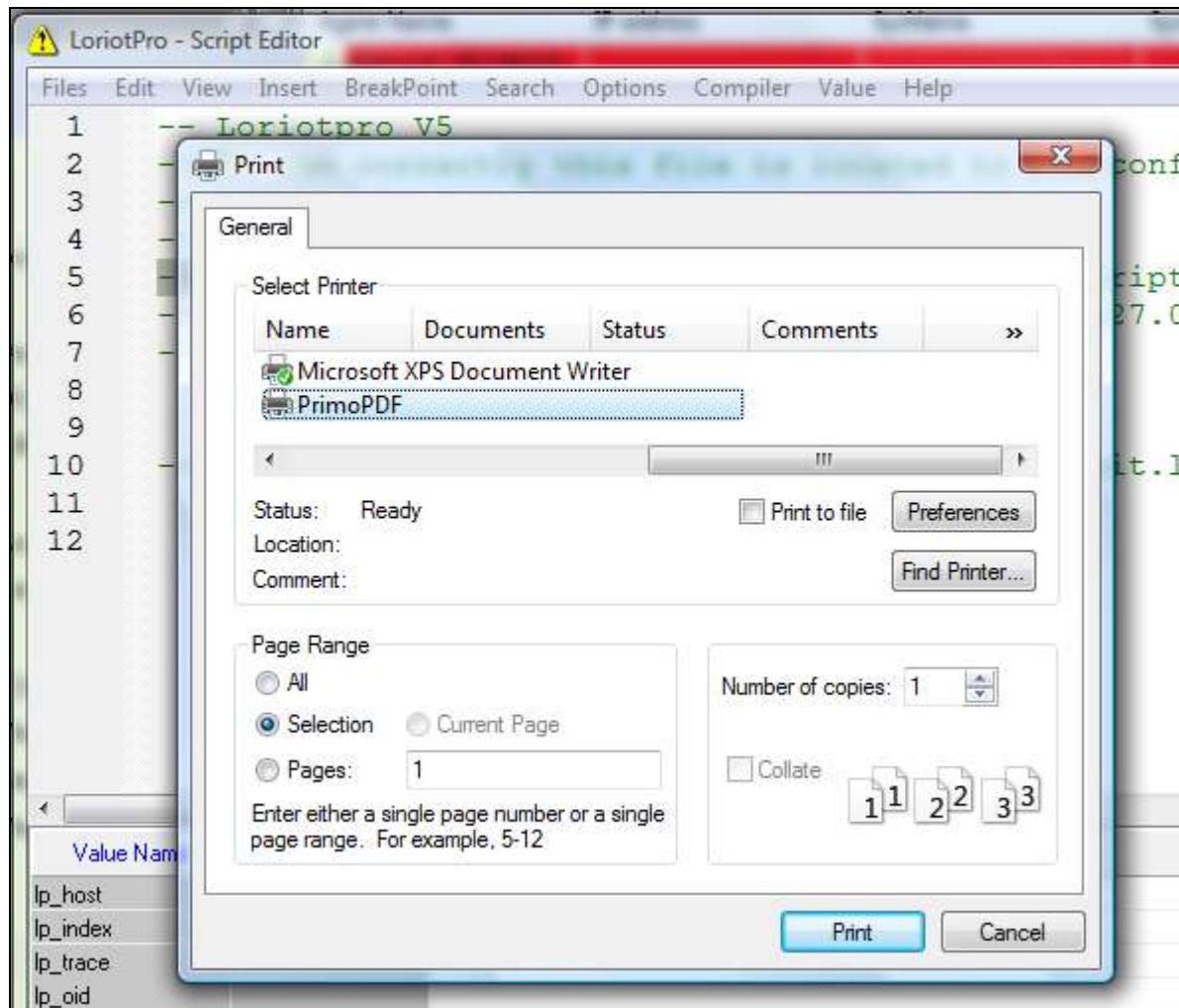
### 8.3.10.9 Options - Save As

Elle vous propose de sauvegarder votre fichier sous un autre nom.



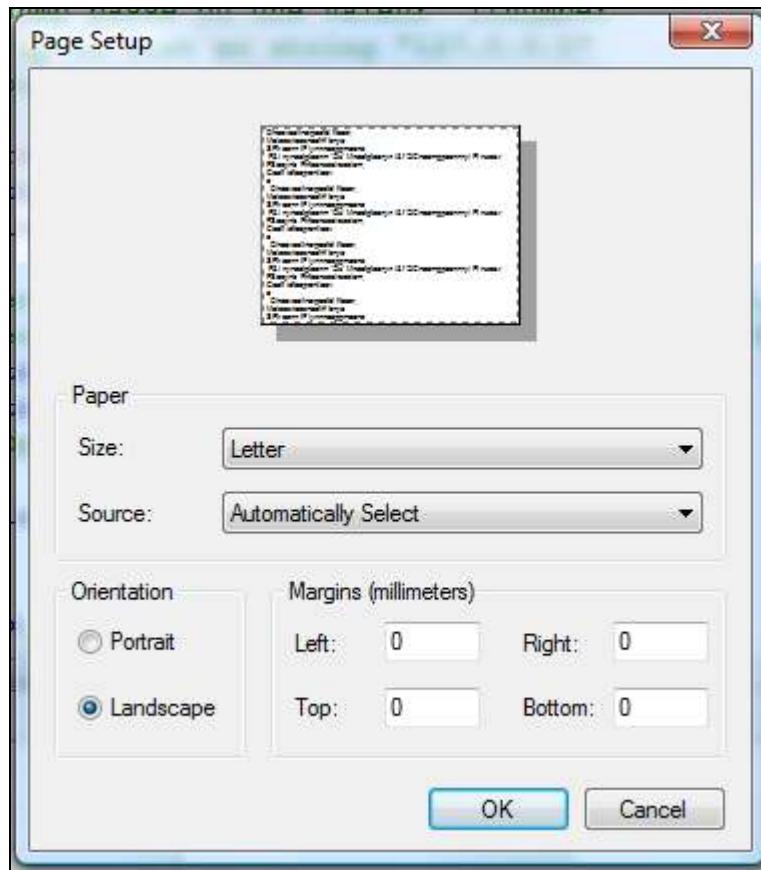
### 8.3.10.10 Options - Print

Elle permet l'impression du code.

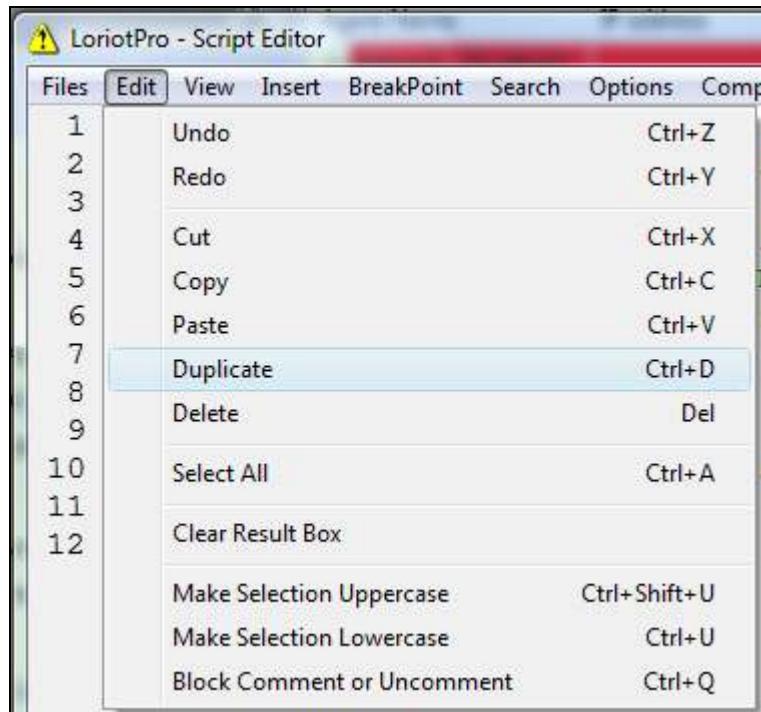


### 8.3.10.11 Options - Page Setup

Elle gère les paramètres d'impression.



### 8.3.11 Menu Edit



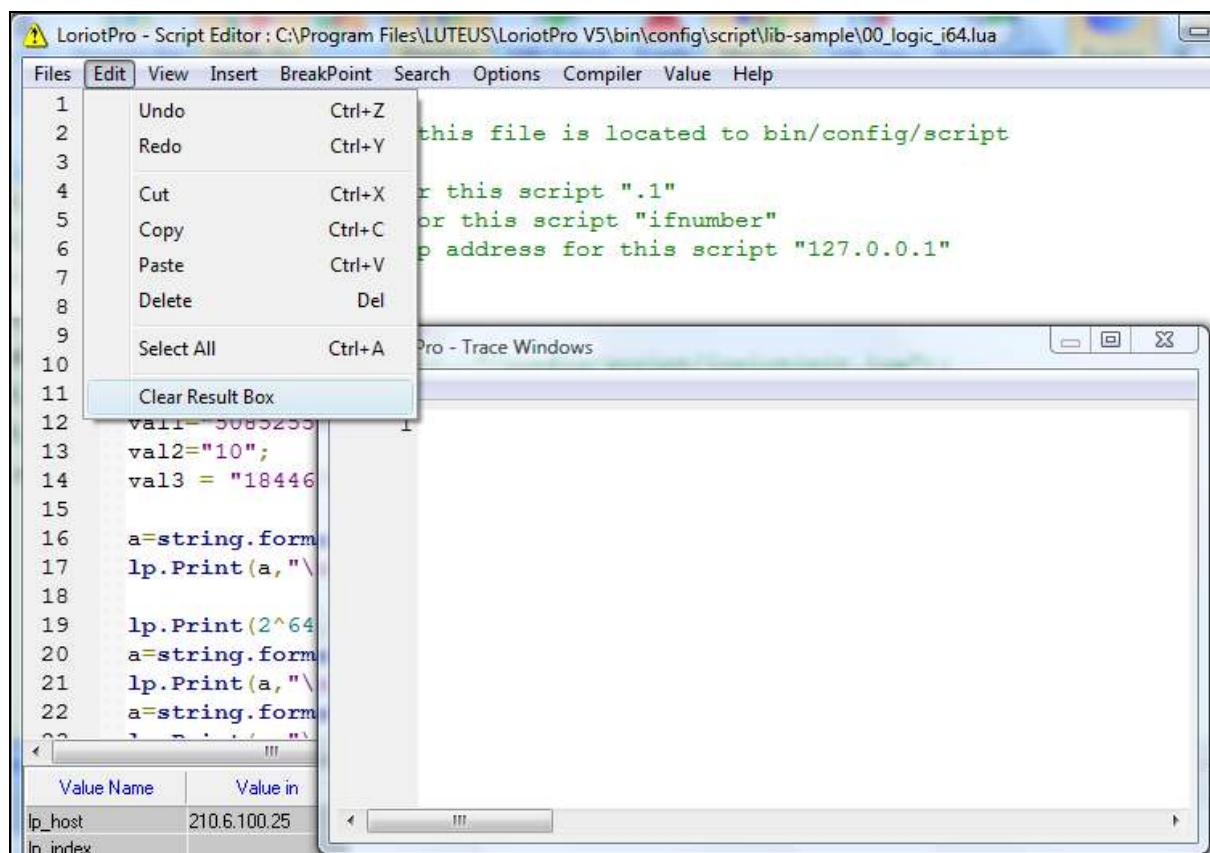
Ce menu permet les opérations de copier / coller classiques.

### 8.3.11.1 Options - Clear Result Box

Une fenêtre est lancée lors de l'interprétation d'un script pour afficher des informations générées par l'exécution du script. Cette option permet d'effacer le contenu de cette fenêtre.

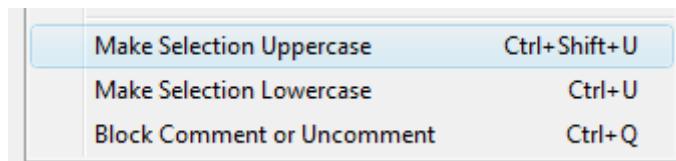
The screenshot shows the LoriotPro interface. On the left, a script editor window displays a LUA script named 'lib-sample\00\_logic\_i64.lua'. The script includes comments about the environment (Loriotpro V4), input values (lp\_index, lp\_oid, lp\_host), and output values (lp\_value, lp\_buffer). It also includes a 'dofile(lp.)' command. On the right, a results window titled 'File' shows the output of the script execution. The output includes several LUA Double for 2^64 with exposant values, a hex value (18446744073709551615), and various functions related to ui64 support. Below the script editor, there is a table with two rows:

Value Name	Value in
lp_host	210.6.100.25
lp_index	



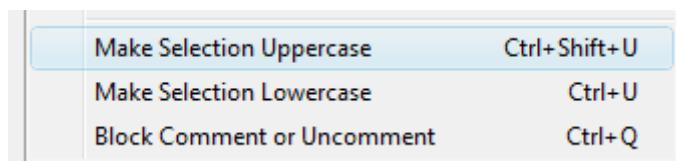
### 8.3.11.2 Options – Make Selection Uppercase

Cette option convertit les caractères sélectionnés en caractères majuscules.



### 8.3.11.3 Options – Make Selection Lowercase

Cette option convertit les caractères sélectionnés en caractères minuscules.



### 8.3.11.4 Options – Block Comment or Uncomment

Cette option place la partie sélectionnée en mode commentaire.

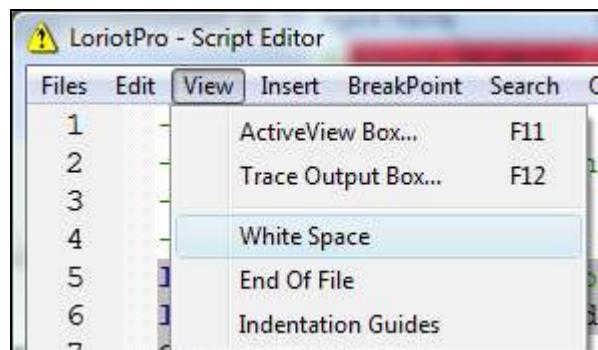
```
4 -- lp_index index for this script ".1"
5 -- lp_oid SNMP OID-- lp_oid SNMP OID for this script "ifnumber"
6 -- lp_host default ip address for this script "127.0.0.1"
7 -- Output Values
8   lp_value = 0;
9   lp_buffer ="error";
10  dofile(lp.GetPath().."config/script/loriotinit.lua");
11
```

Avant

```
4 -- lp_index index for this script ".1"
5 lp_oid SNMP OID-- lp_oid SNMP OID for this script "ifnumber"
6 lp_host default ip address for this script "127.0.0.1"
7 Output Values
8   lp_value = 0;
9   lp_buffer ="error";
10  dofile(lp.GetPath().."config/script/loriotinit.lua");
11
12
```

Après

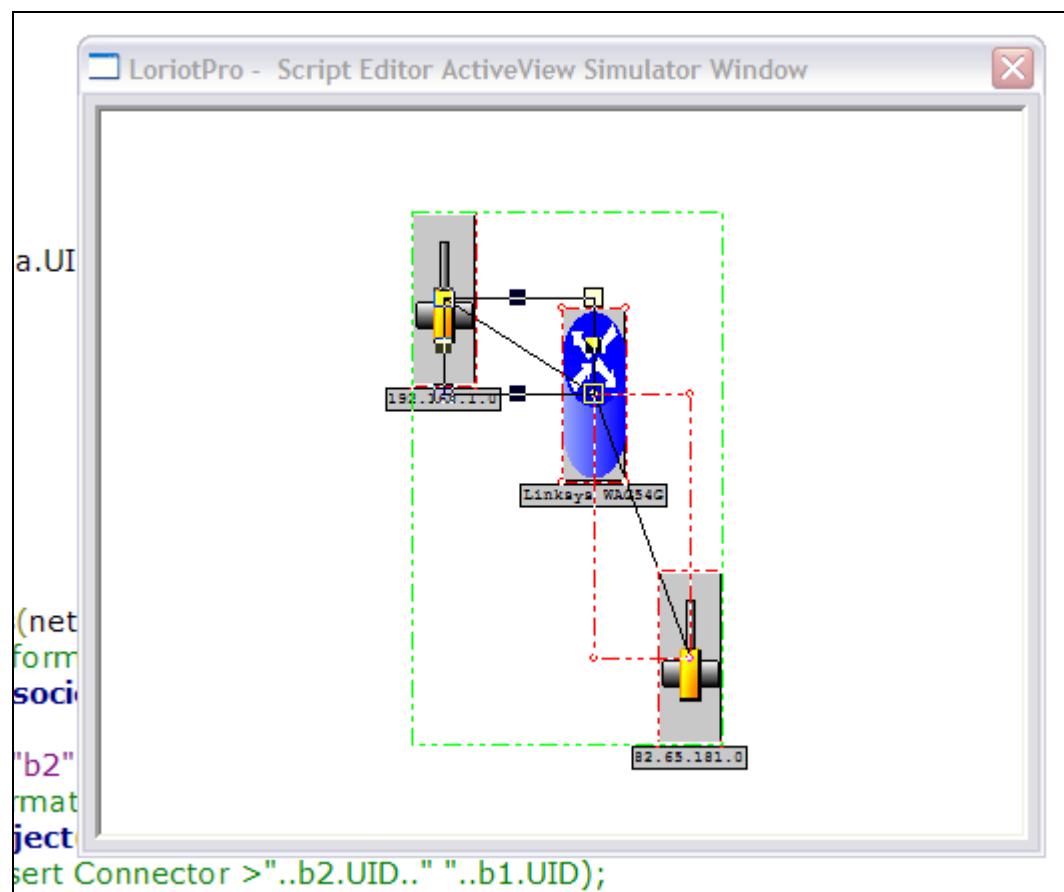
### 8.3.12      Menu View



L'éditeur dispose de deux fenêtres de contrôle permettant de vérifier la bonne exécution des scripts.

#### 8.3.12.1      Options - ActiveView Box

Cette fenêtre permet de simuler une ActiveView.



### 8.3.12.2 Options - Trace Output Box

Cette fenêtre affiche les résultats des fonctions Ip.Print(texte) et Ip.Trace(texte).

The screenshot shows the 'LoriotPro - Running file : C:\Program Files\LUTEUS\LoriotPro V5\bin\config\sc...' window. It displays the results of Lua print statements:  
1 LUA Double for 2^64 with exposant: 1844674407370955.  
2 1.844674407371e+019  
3 LUA Double for 2^64 with no exposant: 1844674407370  
4 LUA Double for 2^64 with in hex : 0  
5 LoriotPro ui64 support librairy 2^64 = 184467440737  
6 18446744073709551615 ==0xFFFFFFFFFFFFFF  
7 function lp.ascii\_ui64sub('18446744073709551615','3  
8 function lp.ascii\_ui64add('18446744073709551615','1  
9 function lp.ascii\_ui64add('18446744073709551600','1  
10 Hex 0xABCD syntax support  
11 function lp.ascii\_ui64add('0xFFFFFFFFFFFFFFF0','10'  
12 function lp.ascii\_ui64add('0xFFFFFFFFFFFFFFF0','10'  
13 function lp.ascii\_ui64ToHex('18446744073709551600')  
14 function double (lua number) to ascii ui64 lp.ascii

### 8.3.12.3 Options – White Space

Cette option permet de visualiser les caractères blancs dans le texte du script.

```
5  --- lp_oid·SNMP·OID---·lp_oid·SNMP·OID·for·this·script·"ifnumber"
6  --- lp_host·default·ip·address·for·this·script·"127.0.0.1"
7  --- Output·Values
8  ... lp_value·=·0;
9  ... lp_buffer·="error";
10 --- dofile(lp.GetPath()..."/config/script/loriotinit.lua");
11
```

Les caractères blancs sont représentés par un point.

### 8.3.12.4 Options – End Of File

Cette option permet de visualiser les caractères de fin de ligne.

```
5  --- lp_oid·SNMP·OID---·lp_oid·SNMP·OID·for·this·script·"ifnumber" C
6  --- lp_host·default·ip·address·for·this·script·"127.0.0.1" CRLF
7  --- Output·Values CRLF
8  ... lp_value·=·0; CRLF
9  ... lp_buffer·="error"; CRLF
10 --- dofile(lp.GetPath()..."/config/script/loriotinit.lua"); CRLF
11 CRLF
12
```

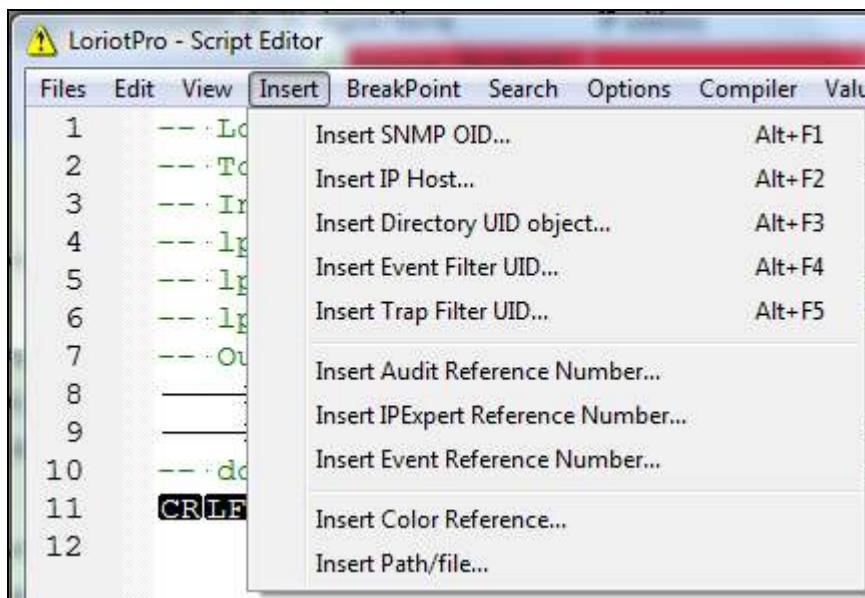
### 8.3.12.5 Options – Indentation Guides

Cette option permet de visualiser les caractères de tabulation contenus dans les lignes de code.

```
5  --- lp_oid·SNMP·OID---·lp_oid·SNMP·OID·for·this·sc
6  --- lp_host·default·ip·address·for·this·script·"1
7  --- Output·Values CRLF
8  →|→|→|→lp_value·=·0; CRLF
9  →|→|→|→lp_buffer·="error"; CRLF
10 --- dofile(lp.GetPath()..."/config/script/loriotin
11 CRLF
```

## 8.3.13 Menu Insert

Des modes Wizard permettent d'insérer dans le texte, à l'emplacement du curseur, des objets SNMP ou LoriotPro.



Ces objets seront insérés à l'emplacement du curseur.

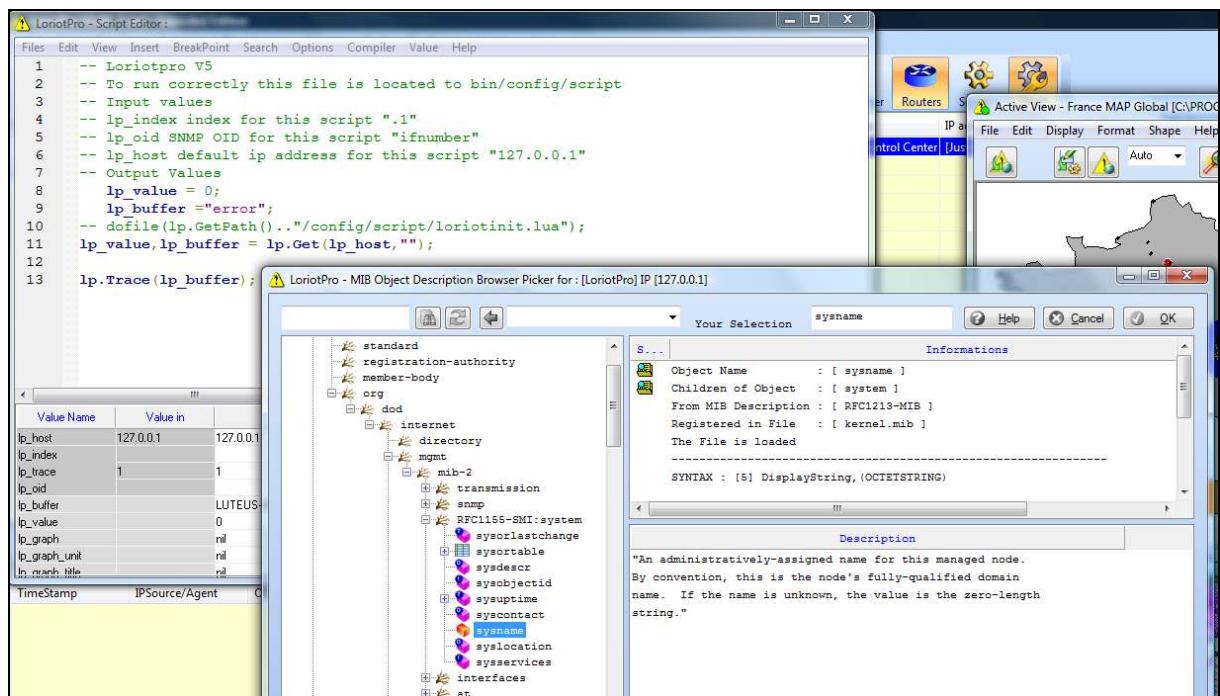
#### 8.3.13.1 Options - Insert SNMP OID

Ce Wizard insert un OID SNMP dans le code à l'emplacement du curseur. L'intérêt de ce module est qu'il peut réaliser une requête sur l'équipement défini par la variable `lp_host` située dans la zone des variables.

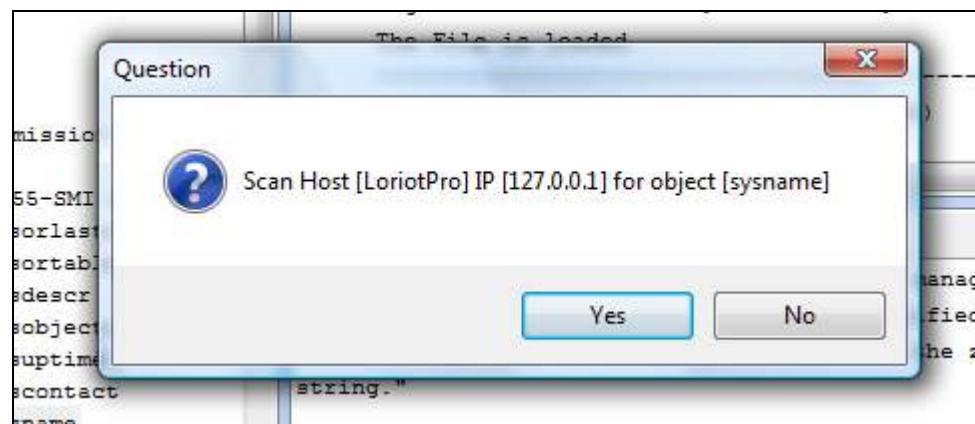
Value Name	Value in	
lp_host	127.0.01	127.0.01
lp_index		
lp_trace	1	1

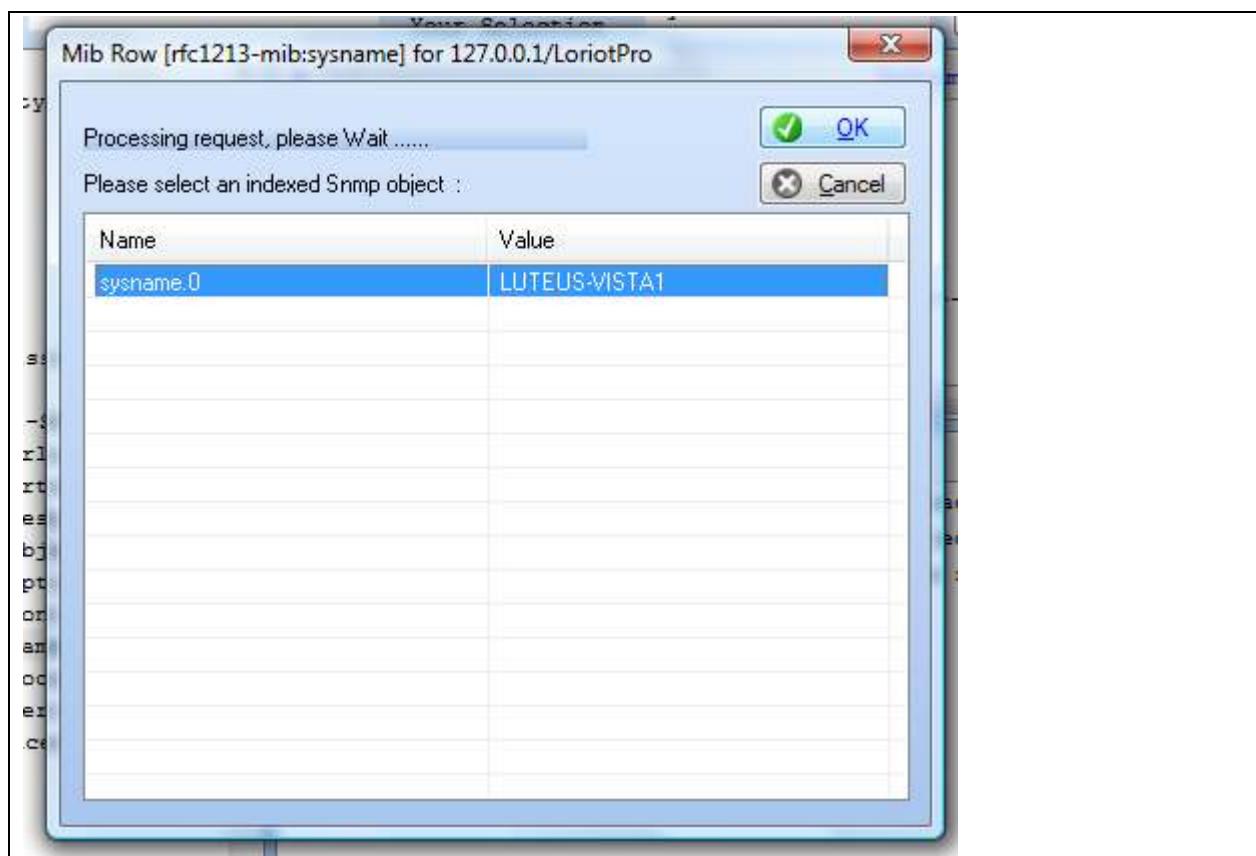
# LoriotPro V5.00 Extended Edition – Supervision et gestion SNMP par scripts LUA

## Introduction au System Expert et aux scripts

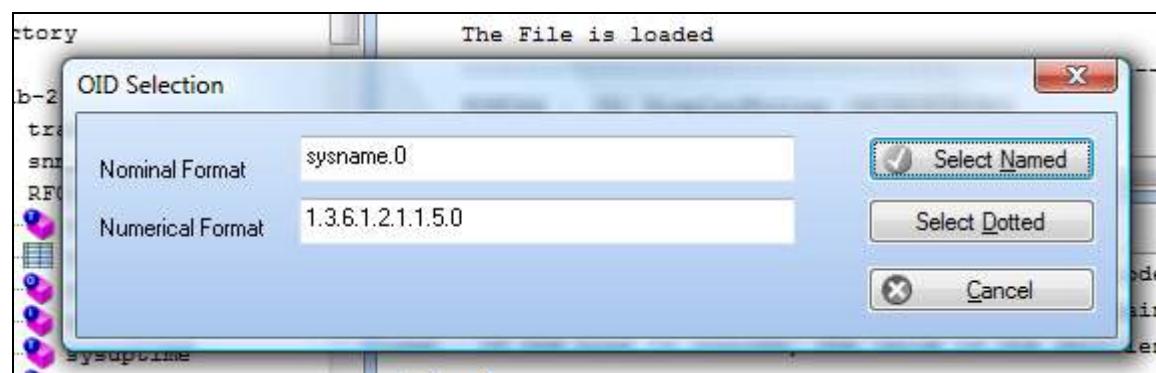


Un double-click sur l'objet OID de l'arbre provoque une requête vers l'équipement.





Si vous validez le Wizard, il vous propose une insertion de l'OID en mode Numerical ou Nominal.



```
lp_value = 0;
lp_buffer ="error";
-- dofile(lp.GetPath().."/config/script/loriotinit.lua");
lp_value,lp_buffer = lp.Get(lp_host,"sysname.0");
```

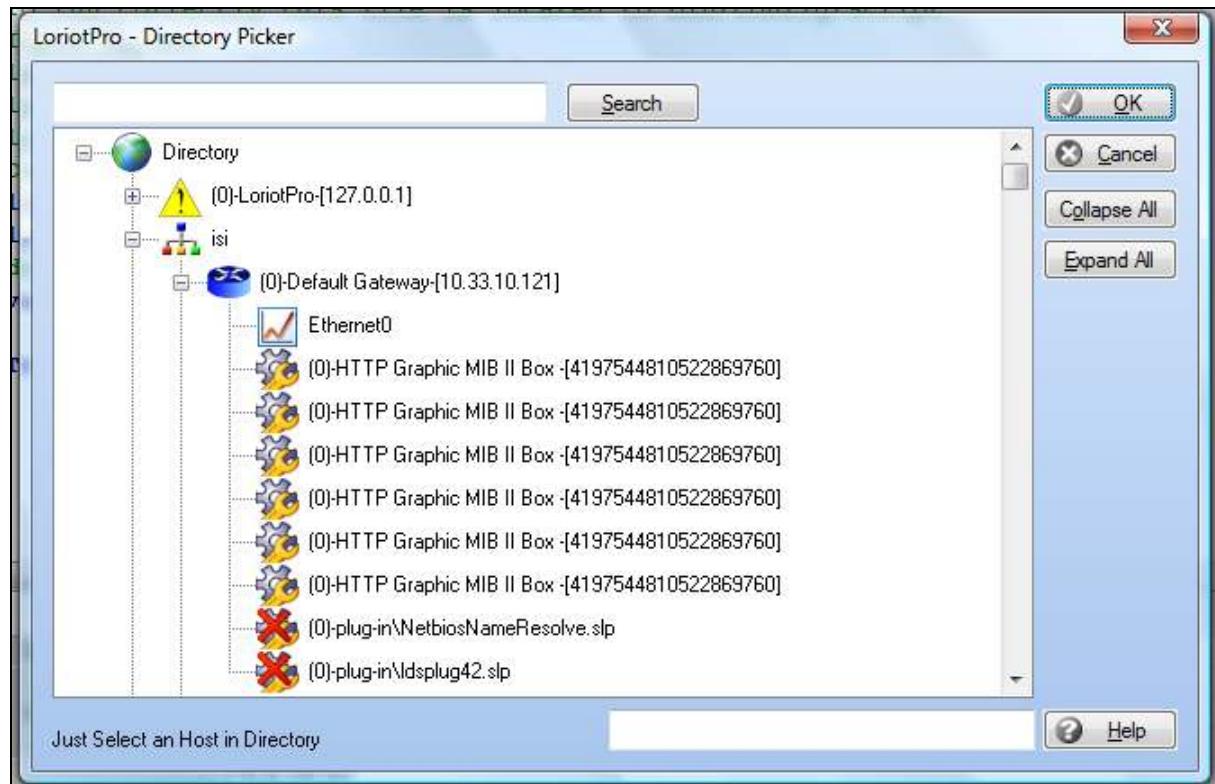
Sysname.0 a été inséré à l'emplacement du curseur et nous sommes sûrs que l'équipement 127.0.0.1 répond à la requête.

### 8.3.13.2 Options - Insert IP Host

Ce Wizard vous propose l'insertion d'une adresse IP de votre Directory dans le script à l'emplacement du curseur.

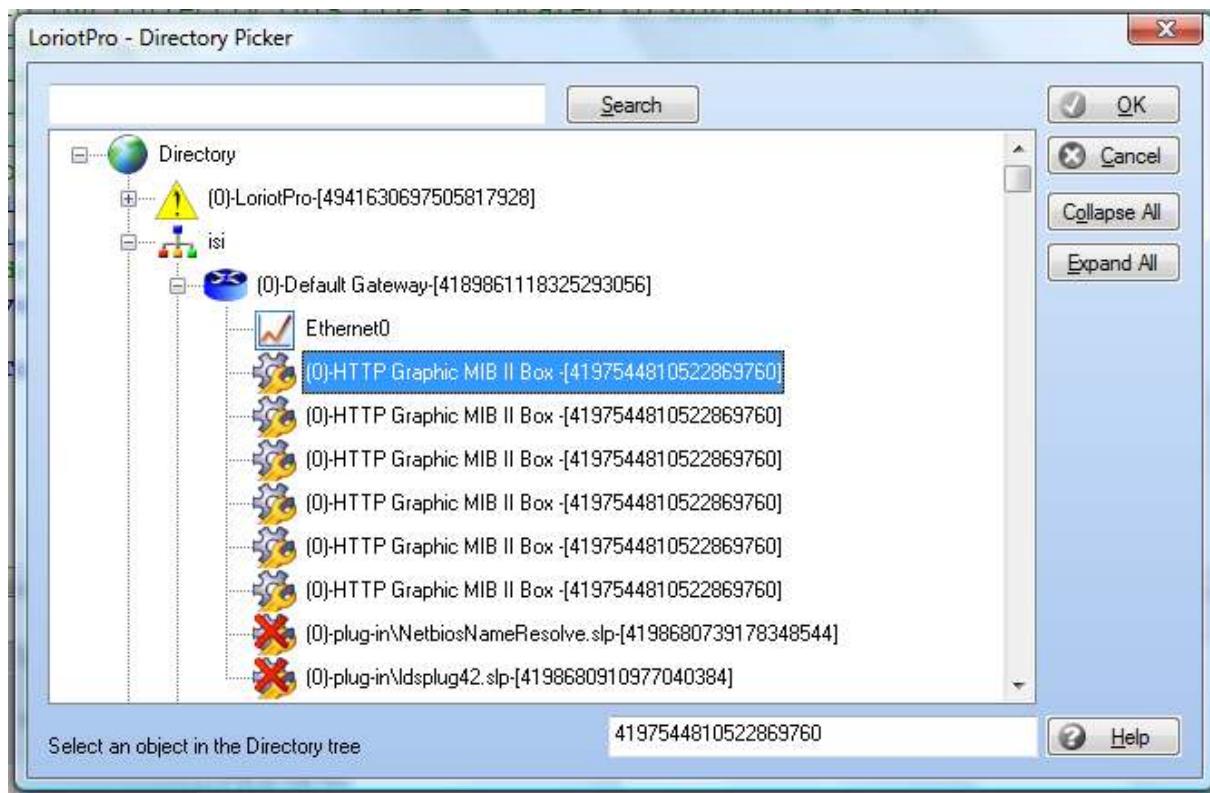
#### Attention

Les scripts ne peuvent réaliser des opérations SNMP que sur les équipements pré configurés dans votre Directory.



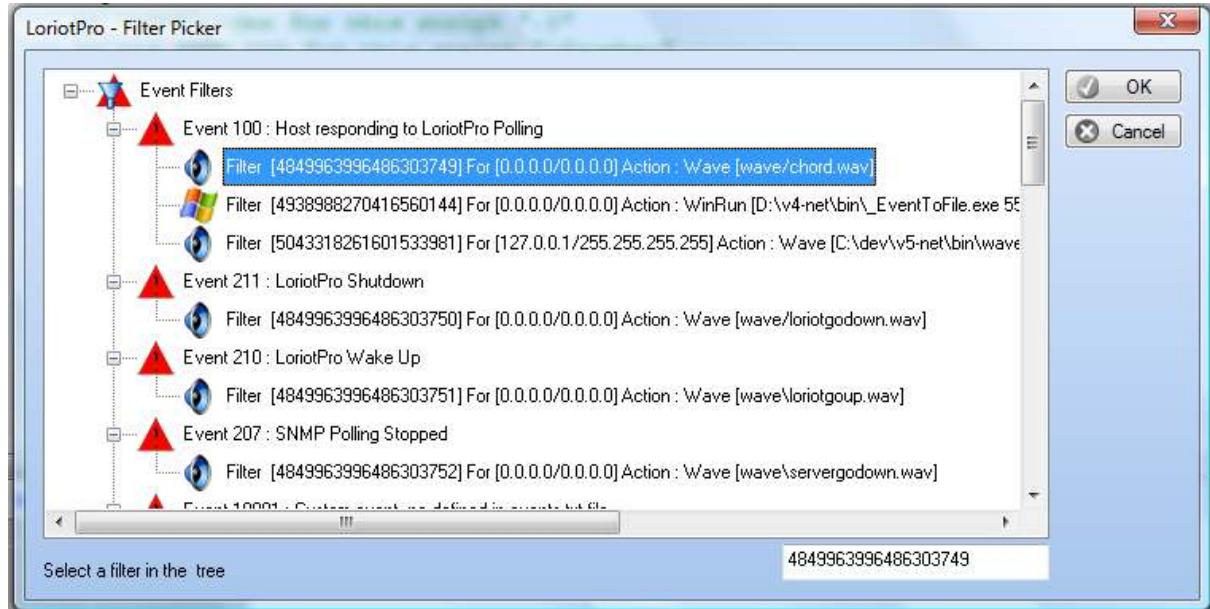
### 8.3.13.3 Options - Insert Directory UID Object

Tous les objets de la Directory possèdent un unique identifiant appelé UID qui est un nombre sur 64bits. Cet UID sert de référence à certaines fonctions des librairies de LoriotPro. Ce Wizard vous aide à trouver le bon UID pour l'objet qui vous concerne.



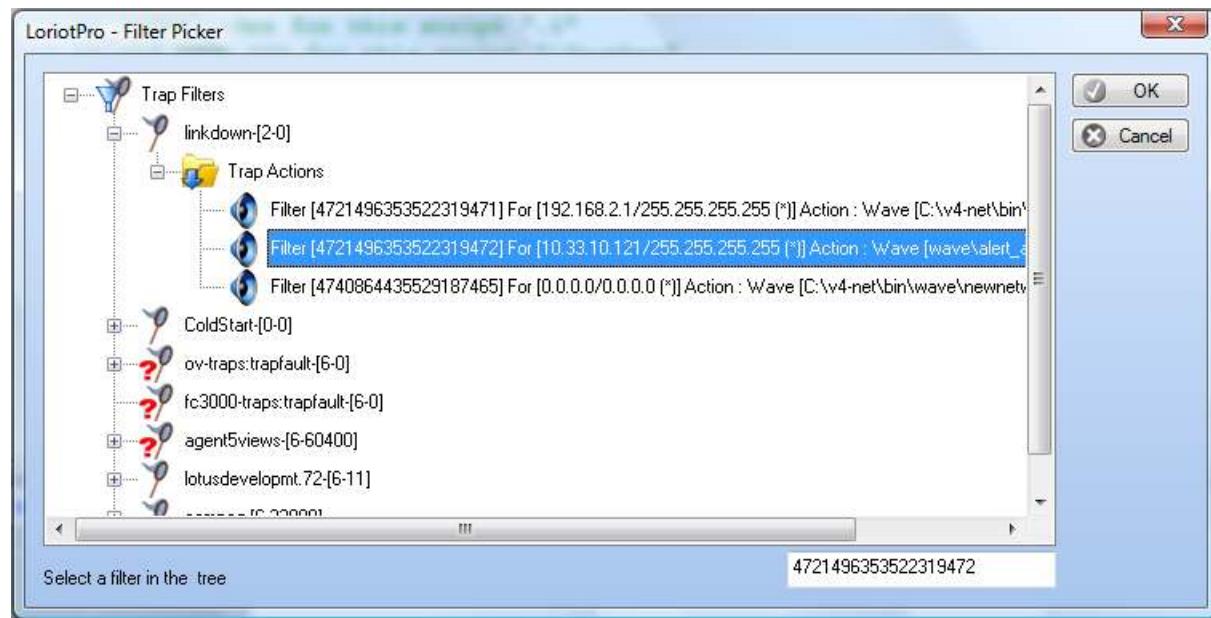
#### 8.3.13.4 Options - Insert Event Filter UID

Tous les filtres d'Events de la Directory possèdent un unique identifiant appelé UID qui est un nombre sur 64bits. Cet UID sert de référence à certaines fonctions des librairies de LoriotPro. Ce Wizard vous aide à trouver le bon UID pour l'objet qui vous concerne.



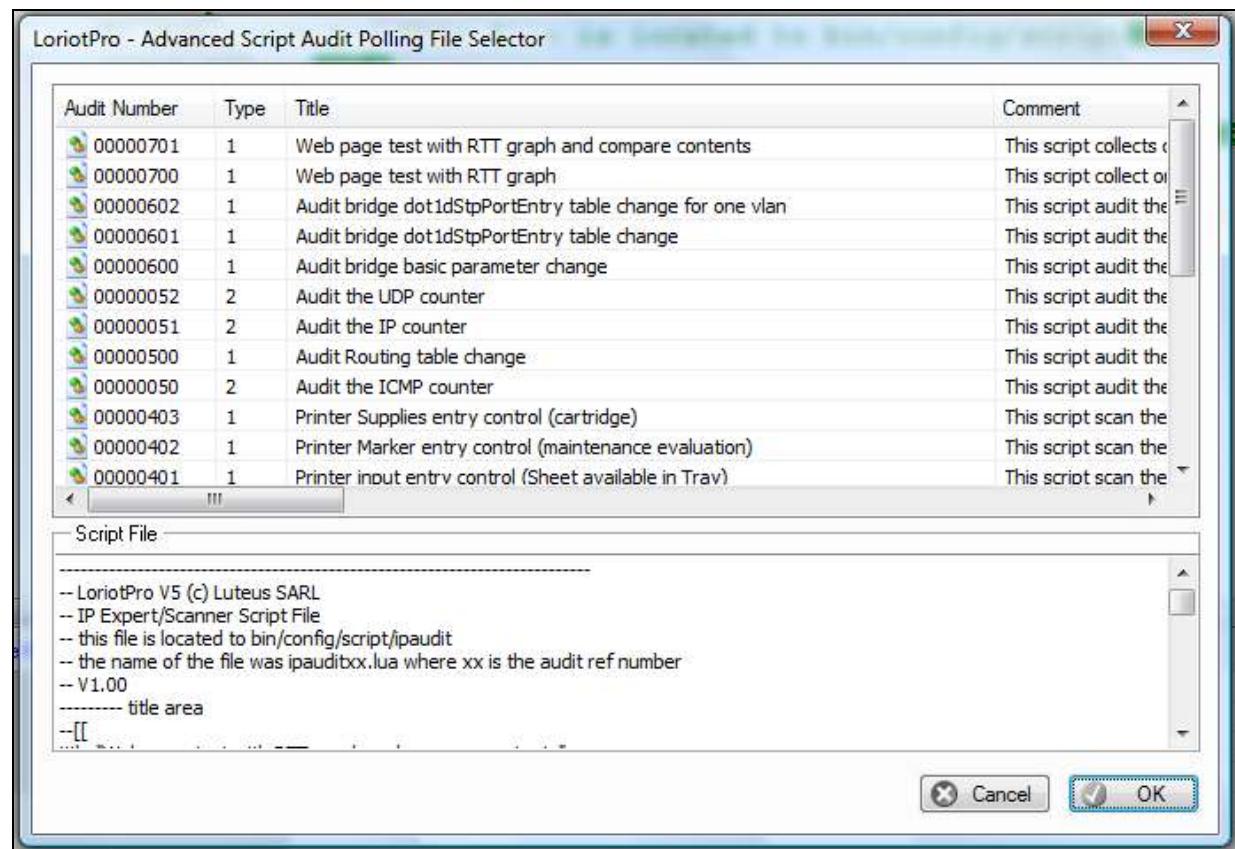
### 8.3.13.5 Options - Insert Trap Filter UID

Tous les filtres de Traps de la Directory possèdent un unique identifiant appelé UID qui est un nombre sur 64bits. Cet UID sert de référence à certaines fonctions des librairies de LoriotPro. Ce Wizard vous aide à trouver le bon UID pour l'objet qui vous concerne.



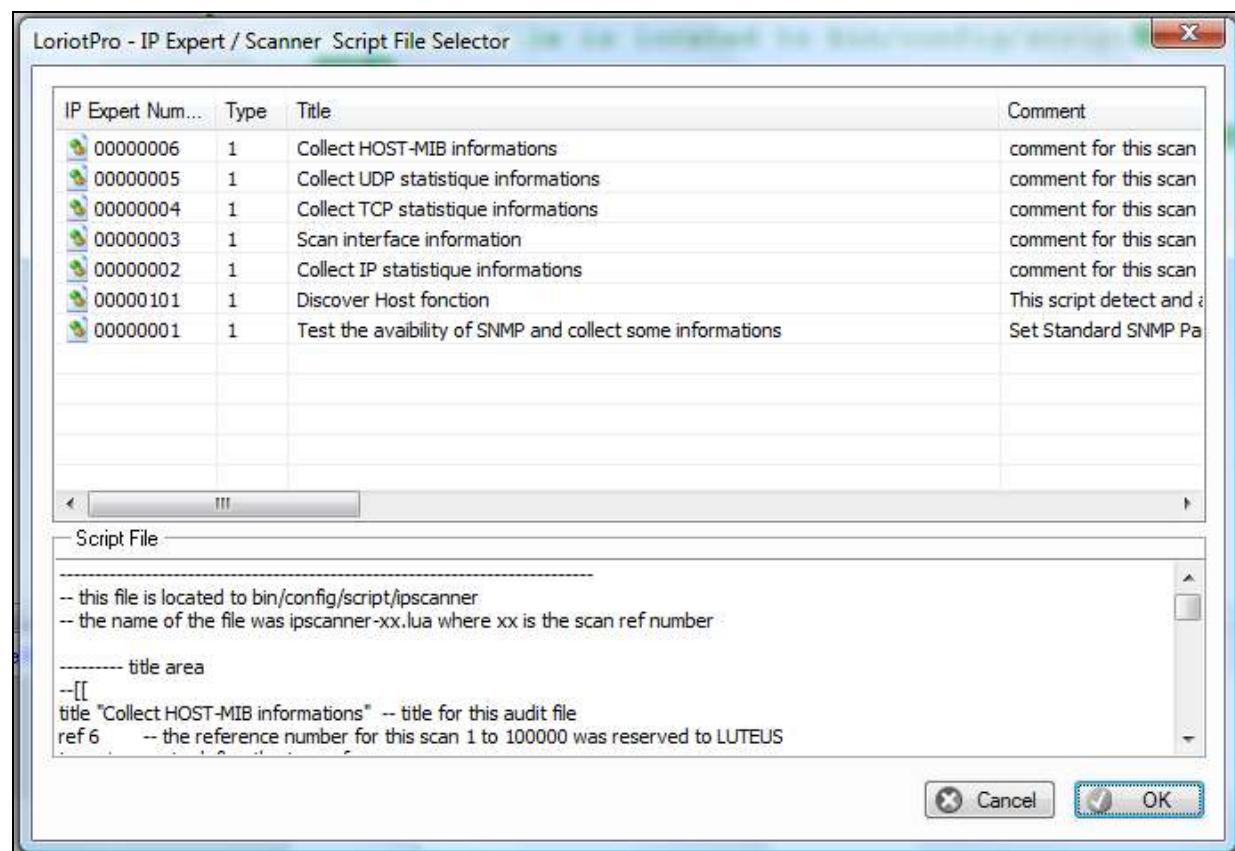
### 8.3.13.6 Options - Insert Audit Reference Number

Chaque script d'audit possède un numéro de référence qui peut être utilisé dans certaines fonctions LUA.



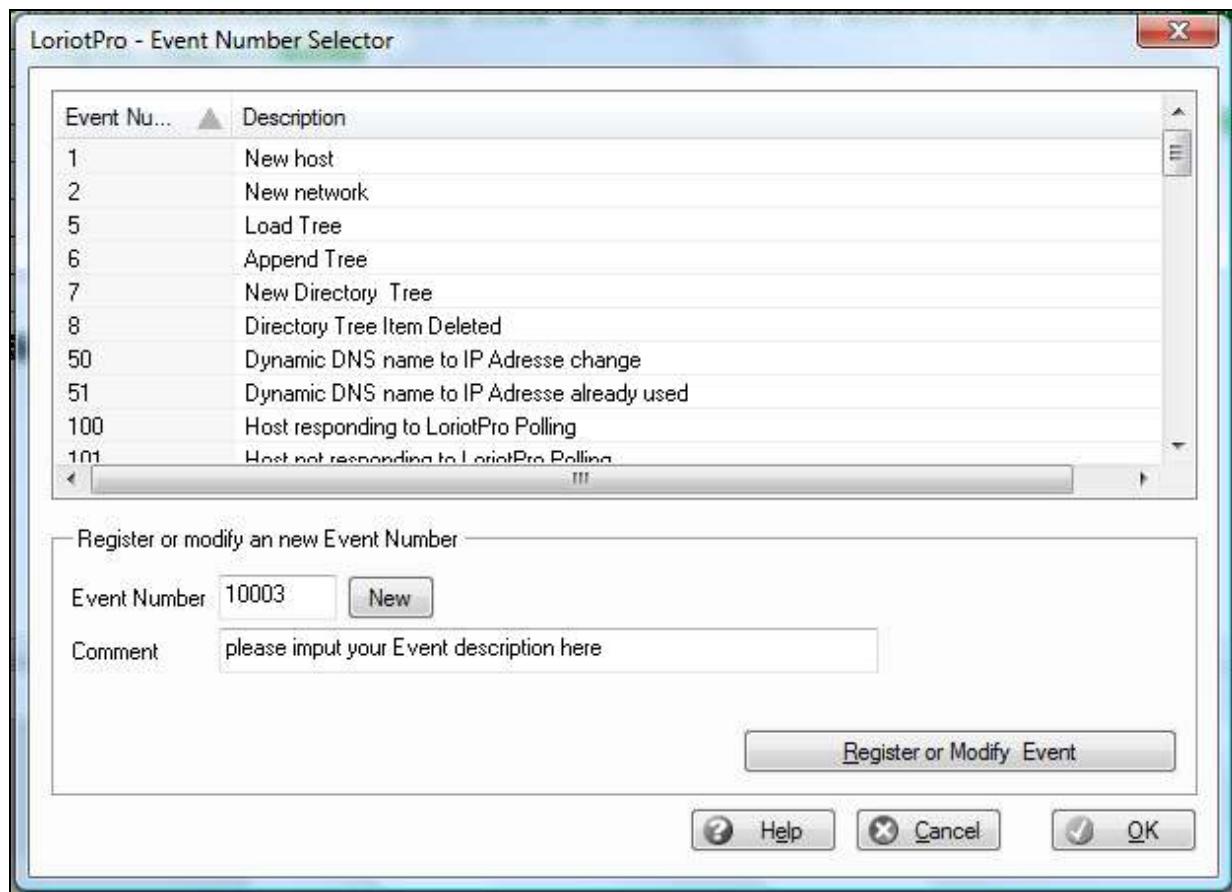
### 8.3.13.7 Options - Insert IPExpert Reference Number

Chaque script d'expertise posséde un numéro de référence qui peut être utilisé dans certaines fonctions LUA.



### 8.3.13.8 Options - Insert Event Reference Number

Il peut être nécessaire de connaître un numéro de référence d'événement qui peut être utilisé dans certaines fonctions LUA.



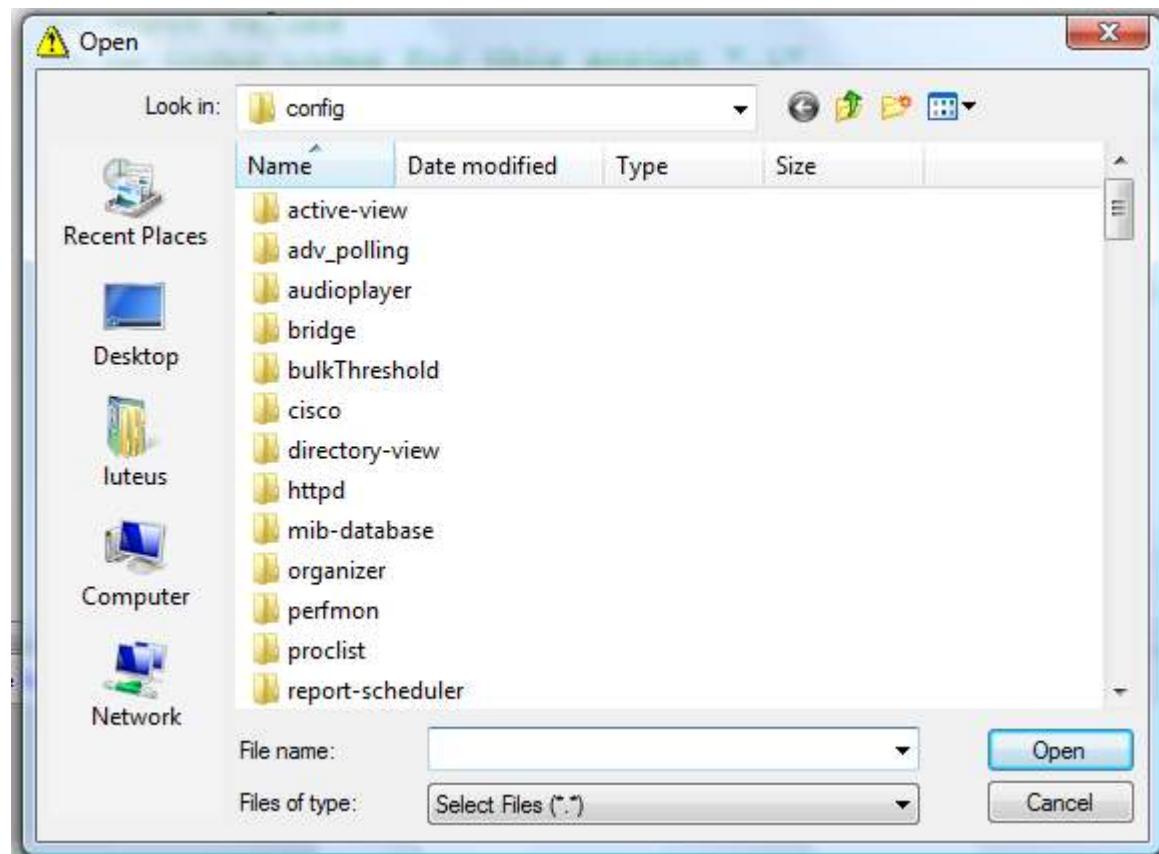
### 8.3.13.9 Options - Insert Color Reference

Il peut être nécessaire de connaître la référence d'une couleur qui peut être utilisée dans certaines fonctions LUA.



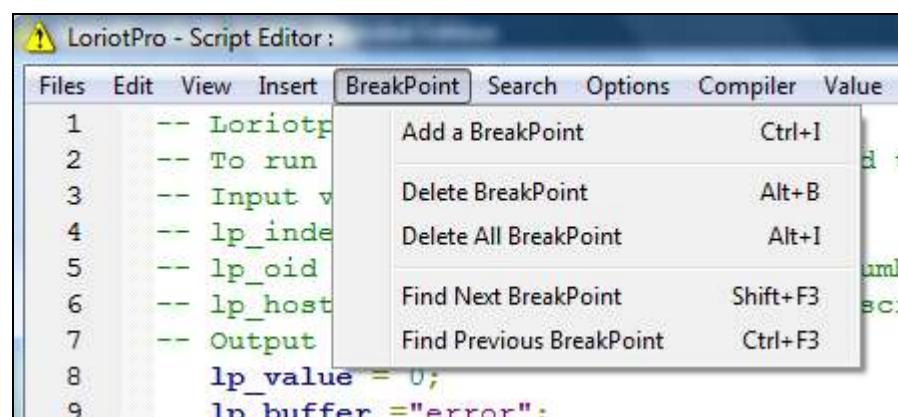
### 8.3.13.10 Options - Insert Path/file

Il peut être nécessaire de connaître le path ou un nom de fichier qui peut être utilisé dans certaines fonctions LUA.



### 8.3.13.11 Options - Menu BreakPoint

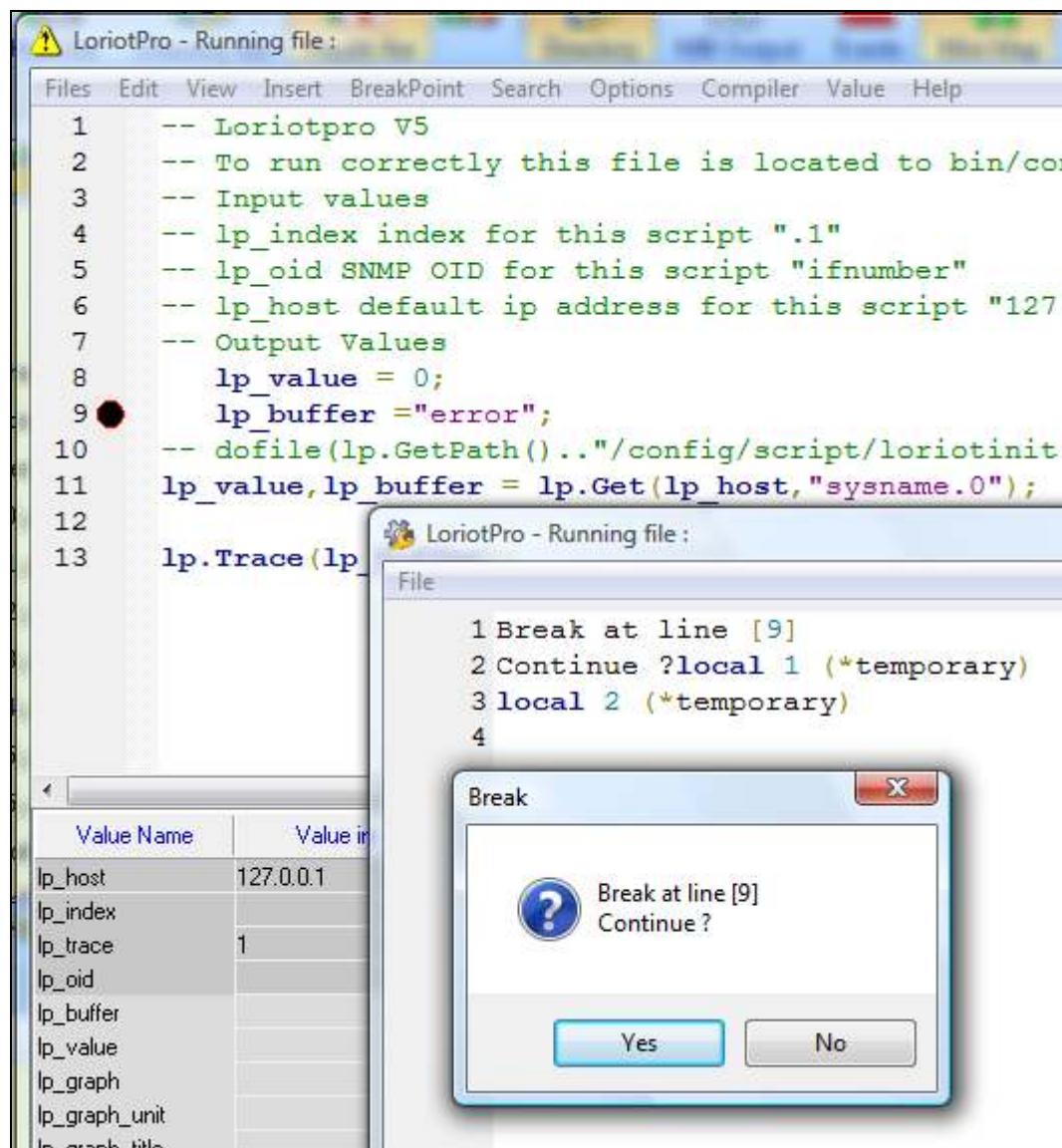
Ce menu gère les breaks points que l'on peut insérer dans le code pour mettre au point vos scripts. Les fonctions d'insertion ou de delete sont associées à l'apposition du curseur.



### 8.3.13.12 Options - Add a BreakPoint

On peut mettre un break point en marge du code pour faciliter la mise au point d'un code. Un break point est représenté par un rond rouge et noir sur la gauche du code.

```
6    -- lp_host default ip
7    -- Output Values
8    lp_value = 0;
9    lp_buffer ="error";
10   -- dofile(lp.GetPath())
11   lp_value,lp_buffer = 1
12
```



Lors de l'exécution du script, une fenêtre est affichée pour vous demandez si vous désirez continuer l'exécution du script.

### 8.3.13.13 Options - Delete BreakPoint

Elle delete le Break Point à l'emplacement du curseur.

### 8.3.13.14 Options - Delete All BreakPoint

Elle delete tous les Breaks Points

### 8.3.13.15 Options - Find Next BreakPoint

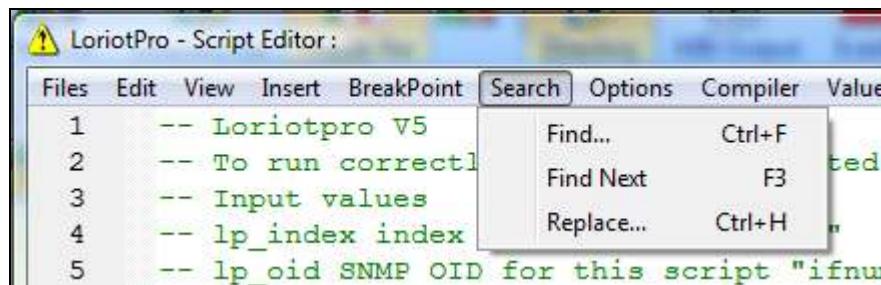
Elle déplace le curseur sur le prochain Break Point en partant de la position du curseur et en allant vers le bas.

### 8.3.13.16 Options - Find Previous BreakPoint

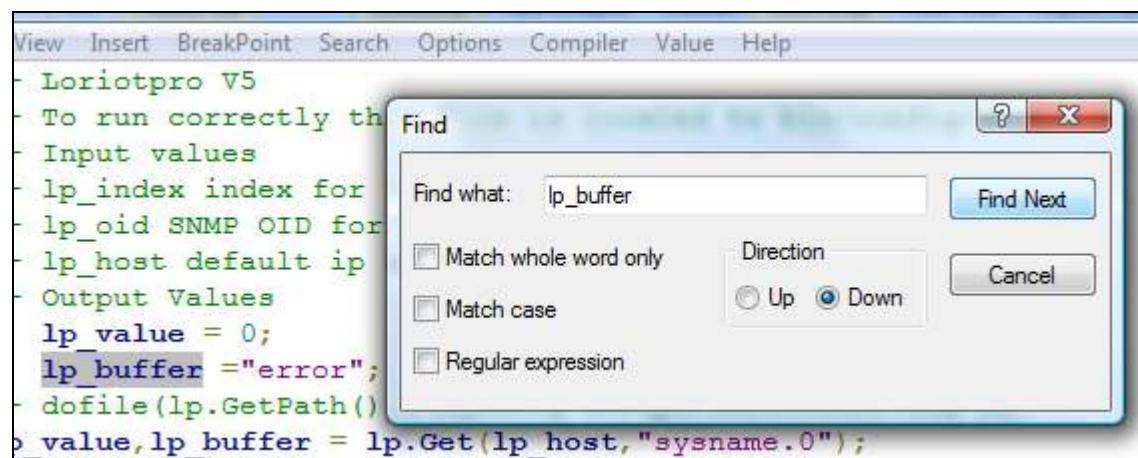
Elle déplace le curseur sur le prochain Break Point en partant de la position du curseur et en allant vers le haut.

## 8.3.14 Menu Search

Ce menu gère les fonctions standard de recherche / replace d'un éditeur.



### 8.3.14.1 Options - Find

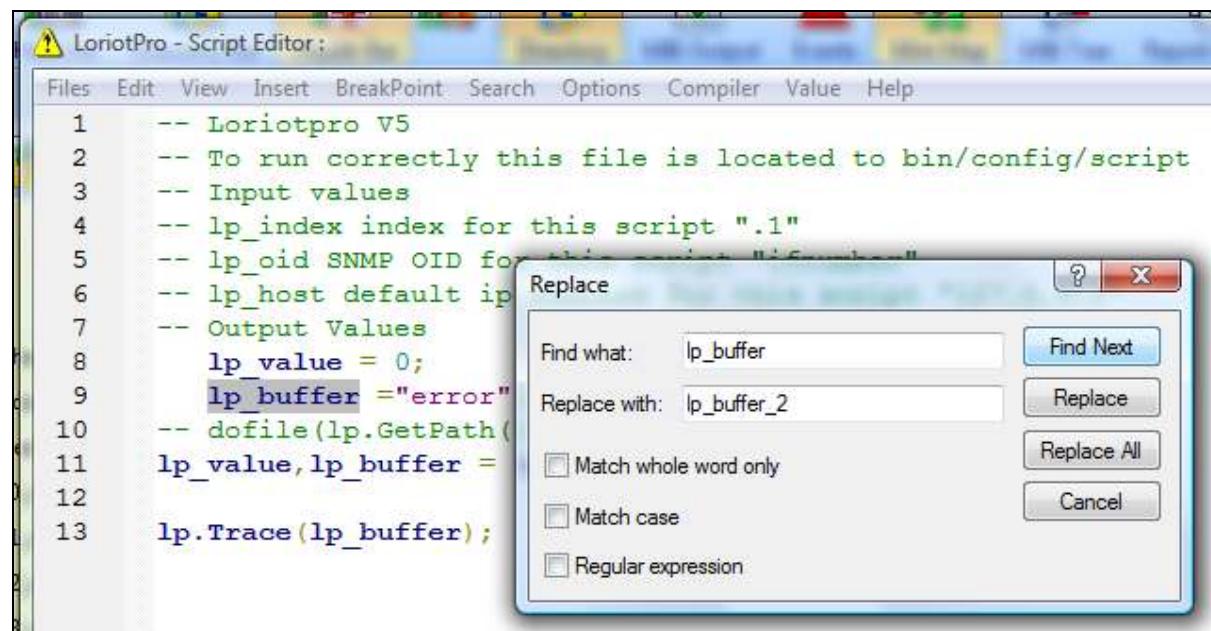


### 8.3.14.2 Options - Find Next

Elle cherche la prochaine occurrence du texte à trouver.

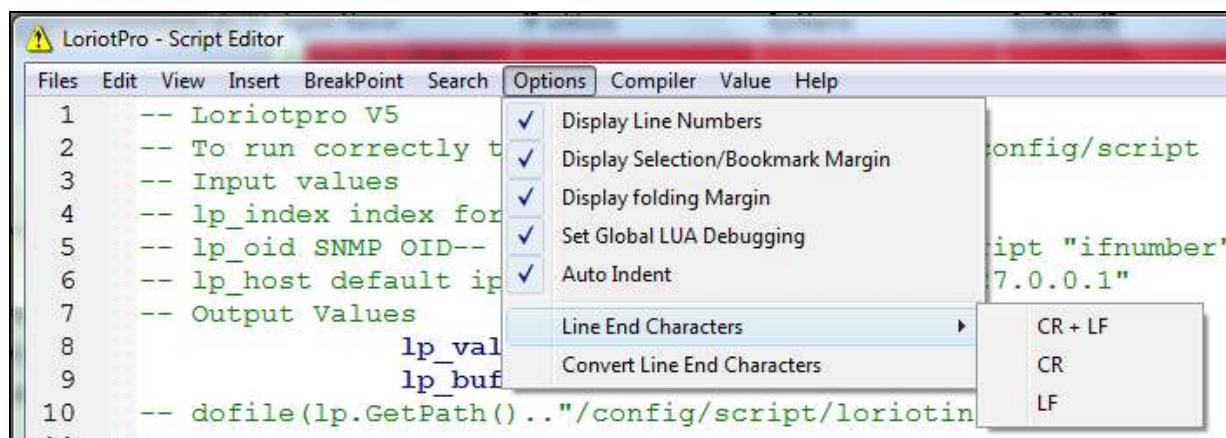
### 8.3.14.3 Options - Replace

Elle permet le remplacement d'une chaîne de caractères par une autre dans le texte du script.



## 8.3.15 Menu Options

Ce menu gère les options de l'éditeur.

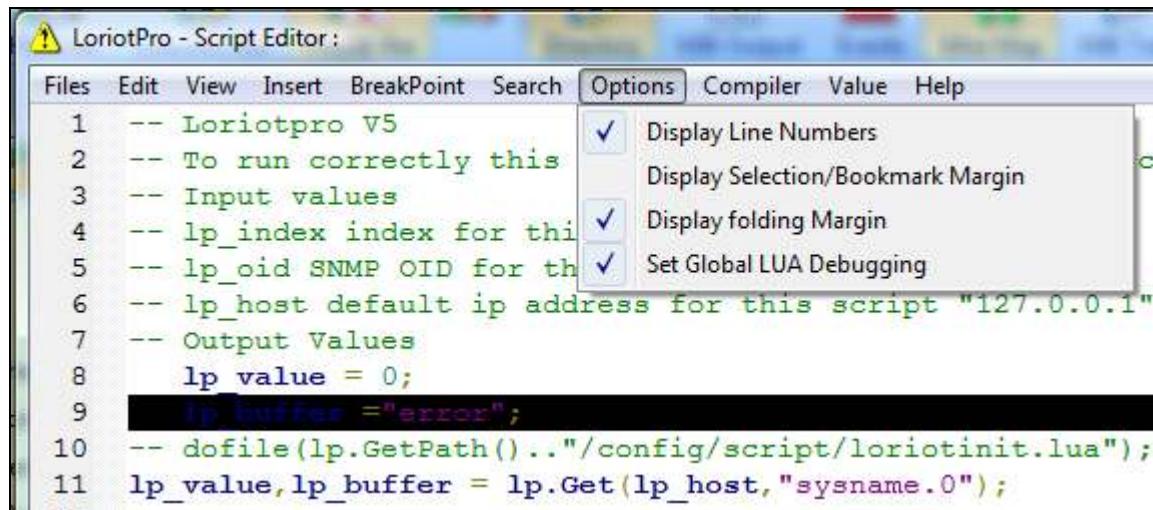


### 8.3.15.1 Options - Display Line Numbers

Affiche ou non les numéros de ligne à gauche de la zone de texte.

### 8.3.15.2 Options - Display Selection/Bookmark Margin

Affiche ou non la zone de Break Point



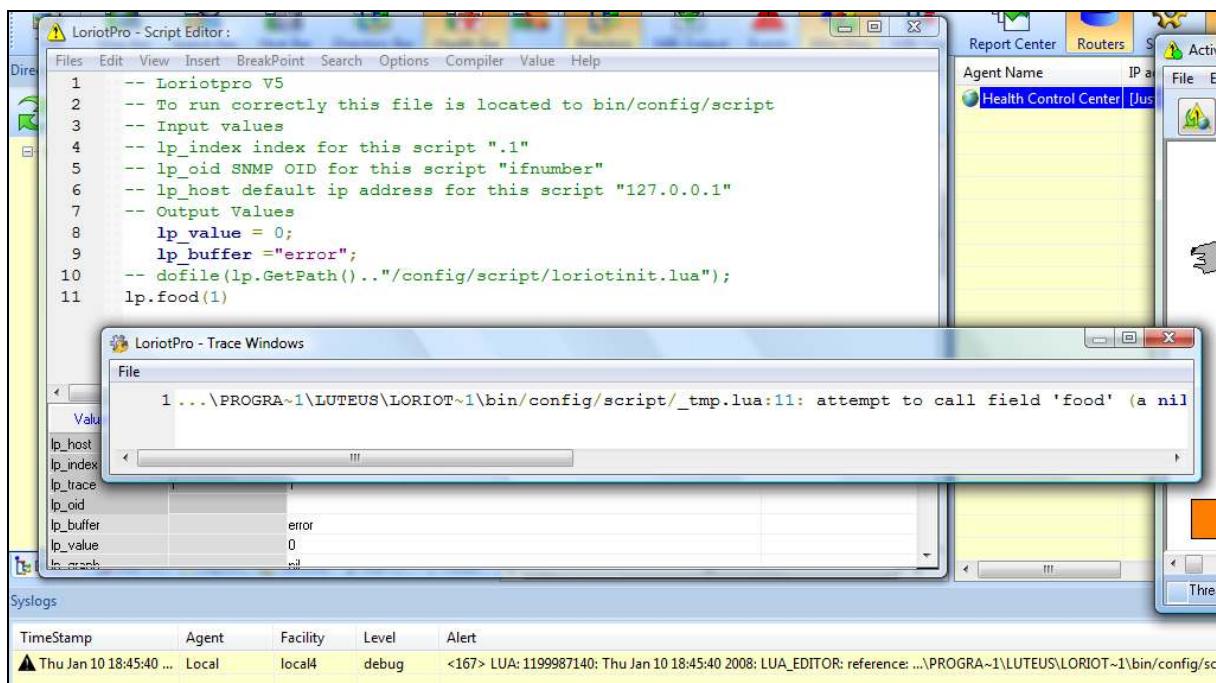
### 8.3.15.3 Options - Display folding Margin

Affiche ou non la zone de gestion des encapsulations de code

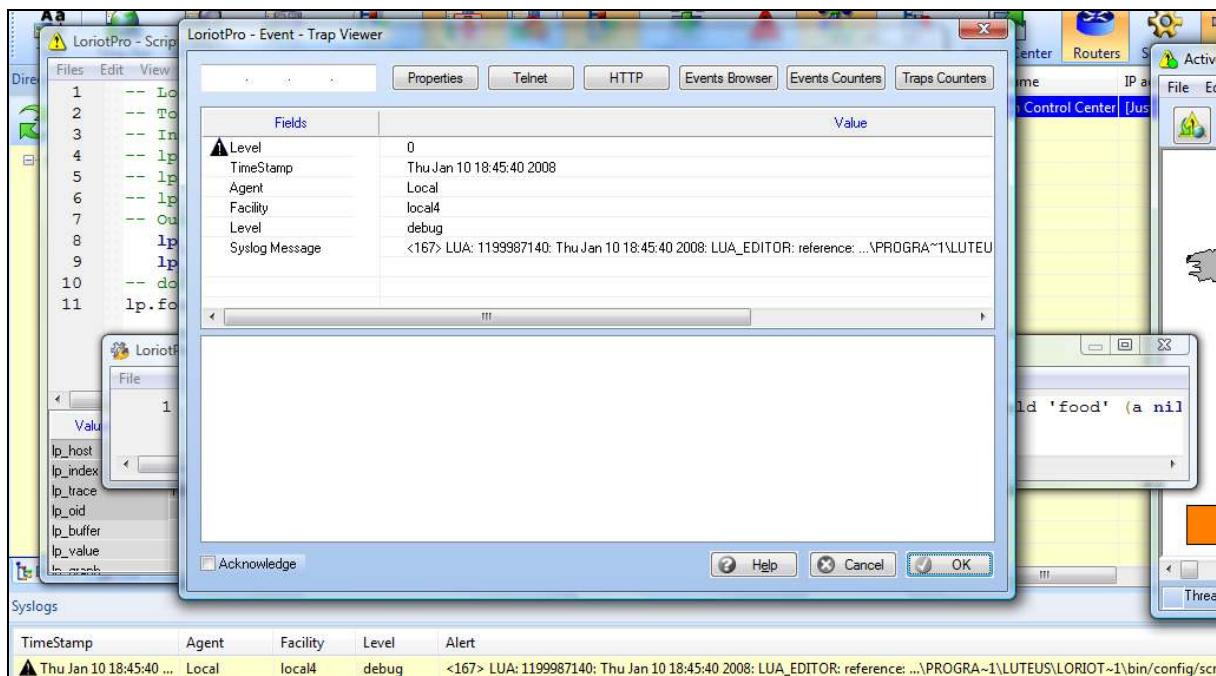
```
55  a=lp.ascii_Getui64(lp_host,"syservices");
56 +if (a) then
59  a=lp.ascii_GetBRCCui64(lp_host,"syservices",1)
60 -if (a) then
61  lp.Print(" lp.ascii_Getui64 :",a,"\n");
62 end
```

### 8.3.15.4 Options - Set Global LUA Debugging

Cette option est très importante, elle permet de rediriger de façon globale les messages d'erreur de l'interpréteur (noyau) LUA vers le module Syslog de LoriotPro. Cette option permet de prendre connaissance de problèmes de code liés à des scripts (objet SNMP virtuel, audit) ne possédant pas de sortie d'affichage.



Le syslog généré dans cet exemple contient des informations sur la ligne de code en défaut dans le script et les références du module qui a généré le syslog. Dans notre exemple, la commande `lp.food(1)` inexistante a généré une erreur reportée par la fenêtre de trace intégrée dans l'éditeur ; un message syslog structuré est parti vers le port UDP 514 de LoriotPro qui a intercepté l'information et l'a affichée dans le module de réception des Syslog.



Dans notre exemple la commande `lp.food(1)` inexistante a généré une erreur reportée par le Syslog :

```
<167> LUA: 1199987140: Thu Jan 10 18:45:40 2008: LUA_EDITOR: reference: ...\\PROGRA~1\\LUTEU
```

### 8.3.15.5 Options – Line End Characters

Cette option permet de définir le type de fin de ligne que le script utilise. Cette option est utile lorsque vous chargez des fichiers d'origine unix qui n'adoptent pas les mêmes normes que les fichiers LoriotPro.

### 8.3.15.6 Options – Convert Line End Characters

Les caractères de fin de ligne du script sont modifiés en fonction du choix de l'option 'Line End Characters'.

The screenshot shows the LoriotPro Script Editor interface. The menu bar includes Files, Edit, View, Insert, BreakPoint, Search, Options, Compiler, Value, and Help. The Options menu is open, showing several checked options: Display Line Numbers, Display Selection/Bookmark Margin, Display folding Margin, Set Global LUA Debugging, and Auto Indent. A submenu, "Line End Characters", is also open, with CR + LF selected. The main code editor window contains a script with various comments and variable definitions.

```
1 -- Loriotpro V5|CR
2 -- To run correctly t|CR
3 -- Input values|CR
4 -- lp_index index for|CR
5 -- lp_oid SNMP OID--|CR
6 -- lp_host default ip|CR
7 -- Output Values|CR
8     lp_val|CR
9     lp_buf|CR
10    -- dofile(lp.GetPath().."config/script/loriotin|CR
11    CR|CR
12    CR|CR
13    CR|CR
```

Avant

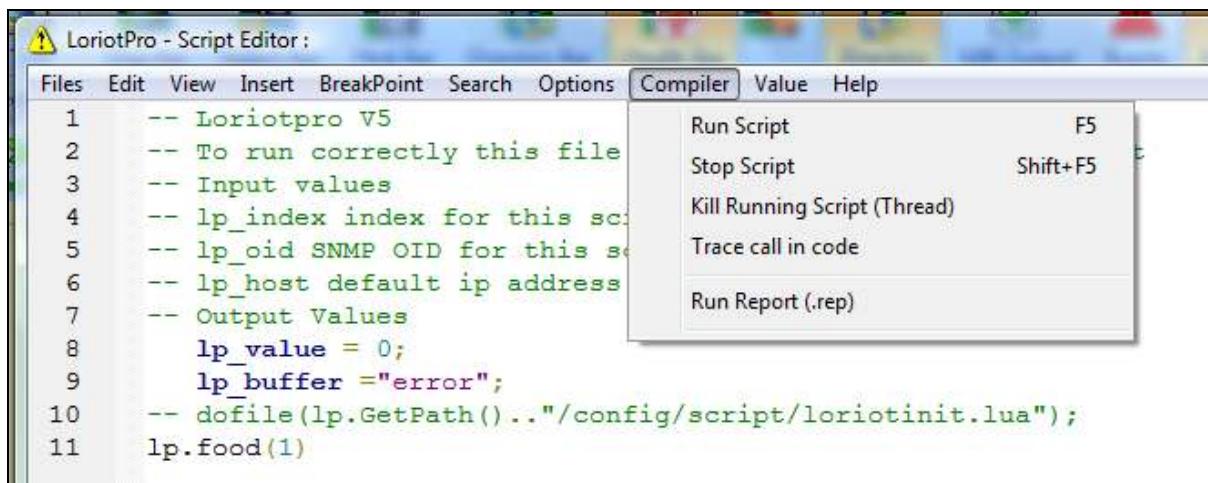
This screenshot shows the same script as above, but with all line endings converted to CRLF. The "Line End Characters" submenu in the Options menu is still open, with CR + LF selected. The code editor window displays the script with CRLF line endings throughout.

```
1 -- Loriotpro V5|CRLF
2 -- To run correctly this file is located to bin/config/script|CRLF
3 -- Input values|CRLF
4 -- lp_index index for this script ".1"|CRLF
5 -- lp_oid SNMP OID-- lp_oid SNMP OID for this script "ifnumber"|CRLF
6 -- lp_host default ip address for this script "127.0.0.1"|CRLF
7 -- Output Values|CRLF
8     lp_value = 0;|CRLF
9     lp_buffer ="error";|CRLF
10    -- dofile(lp.GetPath().."config/script/loriotinit.lua");|CRLF
11    |CRLF
12    |CRLF
13    |CRLF
```

Après

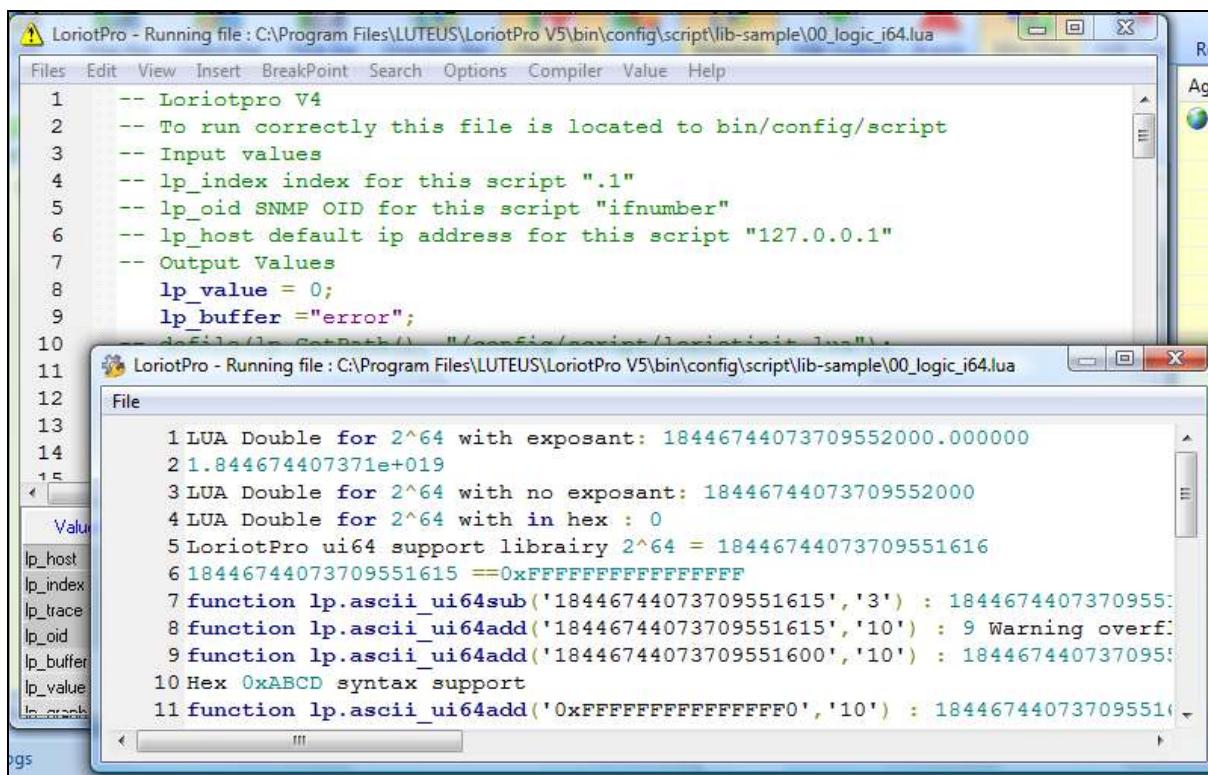
### 8.3.16      Menu Compiler

Ce menu permet de lancer le script par sa mise au point. Il ne s'agit pas réellement de compilation mais d'une exécution du code.



### 8.3.16.1 Options - Run Script

Elle exécute le script en cours d'écriture ; la fenêtre de trace est automatiquement affichée pour fournir des informations sur l'exécution du script.



Si le script contient des erreurs, la fenêtre d'exécution affiche l'emplacement de l'erreur rencontrée par l'interpréteur. Durant l'exécution, la fenêtre affiche le nom du script en cours et le terme « Running ». Lorsque l'exécution du script est terminé, la fenêtre affiche « Trace Windows ». Durant l'exécution, il n'est pas possible de scroller les informations contenues dans la fenêtre.

The screenshot shows a window titled "LoriotPro - Trace Windows". The output pane displays the following LUA code and its execution results:

```
1 LUA Double for 2^64 with exposant: 18446744073709552000.000000
2 1.844674407371e+019
3 LUA Double for 2^64 with no exposant: 18446744073709552000
4 LUA Double for 2^64 with in hex : 0
5 LoriotPro ui64 support librairy 2^64 = 18446744073709551616
6 18446744073709551615 ==0xFFFFFFFFFFFFFFFFF
7 function lp.ascii_ui64sub('18446744073709551615','3') : 18446744073709551612
8 function lp.ascii_ui64add('18446744073709551615','10') : 9 Warning overflow
9 function lp.ascii_ui64add('18446744073709551600','10') : 18446744073709551610
10 Hex 0xABCD syntax support
11 function lp.ascii_ui64add('0xFFFFFFFFFFFFFFF0','10') : 18446744073709551610
```

The screenshot shows a window titled "LoriotPro - Trace Windows". The output pane displays the following LUA code and an error message:

```
11
12     val1=="5085255619359801346";
13     val2="10";
14
```

1 C:\PROGRA~1\LUTEUS\LORIOT~1\bin\config\script\\_tmp.lua:12: '=' expected near '=='

### 8.3.16.2 Options - Stop Script

Vous pouvez utiliser cette option lors de l'exécution d'un script pour générer un break.

#### Attention

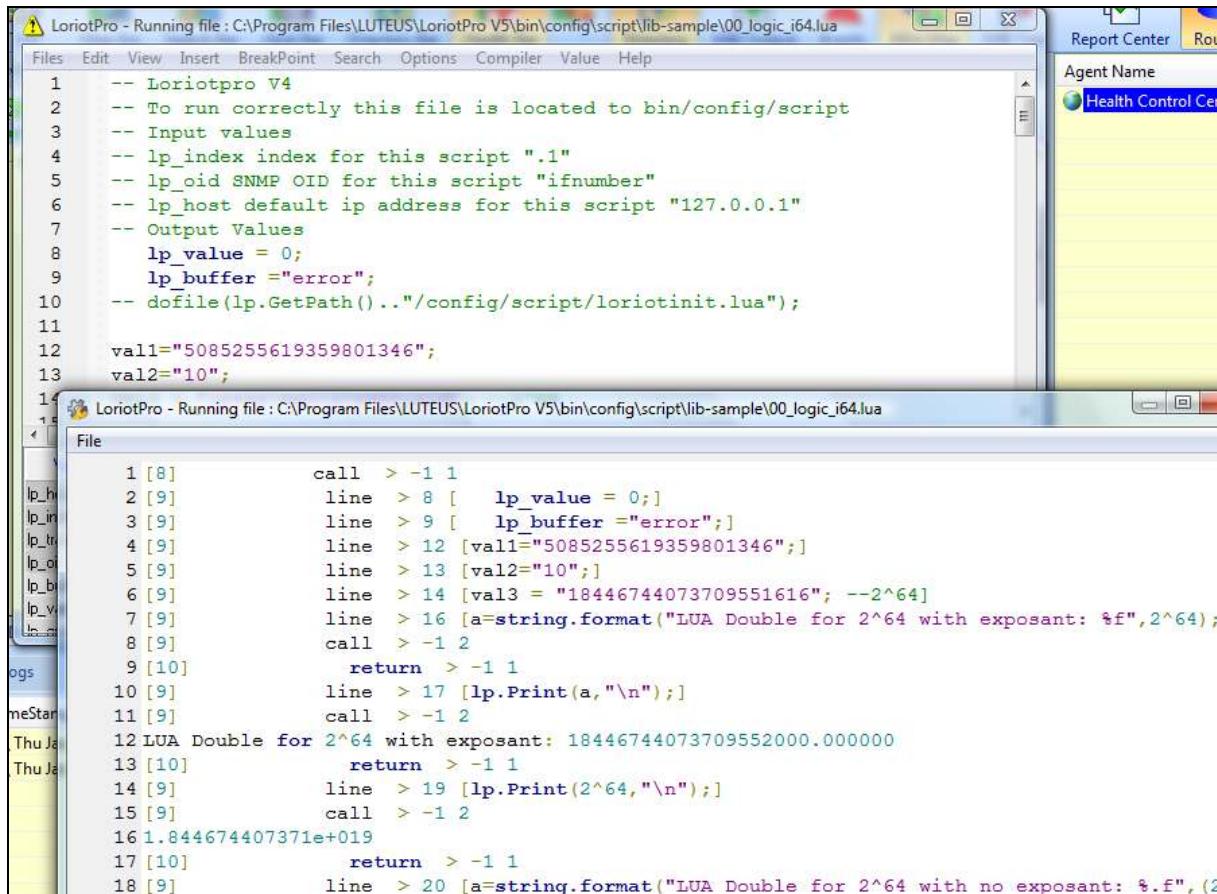
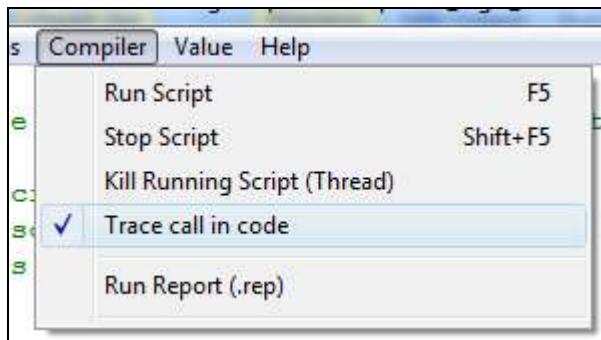
Cette option ne permet pas de sortir d'une boucle de codage.

### 8.3.16.3 Options - Kill Running Script

Cette option stoppe le thread associé à l'exécution du script, elle permet de sortir d'une boucle. Son utilisation provoque quelquefois un blocage de l'interpréteur ; dans ce cas, sauvegardez votre script et relancez une nouvelle fenêtre d'édition.

### 8.3.16.4 Options - Trace call in code

Cette option permet d'afficher dans la fenêtre de trace des informations sur le déroulement de l'exécution du script.



Cette option permet de mettre au point les scripts.

### 8.3.16.5 Options - Run Report

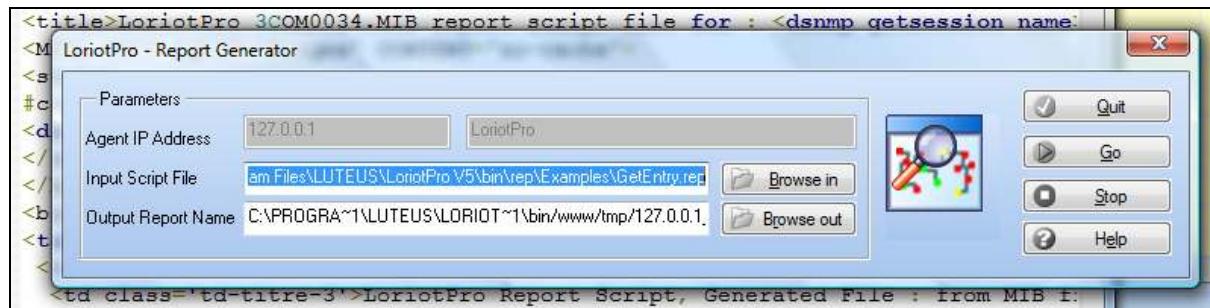
Si le fichier chargé est un fichier de report vous pouvez le tester à travers l'éditeur. L'adresse IP utilisée est la variable lp\_host.

## LoriotPro V5.00 Extended Edition – Supervision et gestion SNMP par scripts LUA Introduction au System Expert et aux scripts

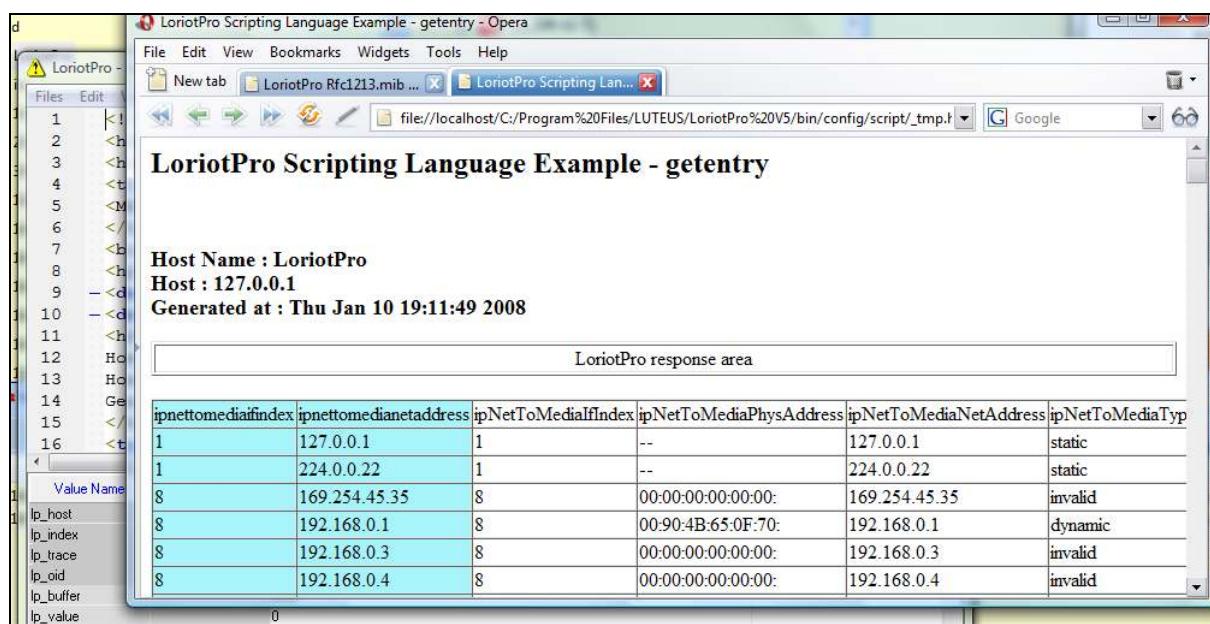
The screenshot shows the LoriotPro Script Editor window. The menu bar includes Files, Edit, View, Insert, BreakPoint, Search, Options, Compiler, Value, and Help. The main area displays a LUA script titled "LoriotPro Scripting Language Example - getentry.rep". The script contains various HTML tags and SNMP functions like <dsnmp>. Below the script is a table with columns "Value Name", "Value in", and "Value out". The table data is as follows:

Value Name	Value in	Value out
lp_host	127.0.0.1	127.0.0.1
lp_index		
lp_trace	1	1
lp_oid		
lp_buffer		error
lp_value		0
lp_graph		nil

La génération de ce type de fichier n'est pas traitée dans ce document mais dans la documentation générale du logiciel.

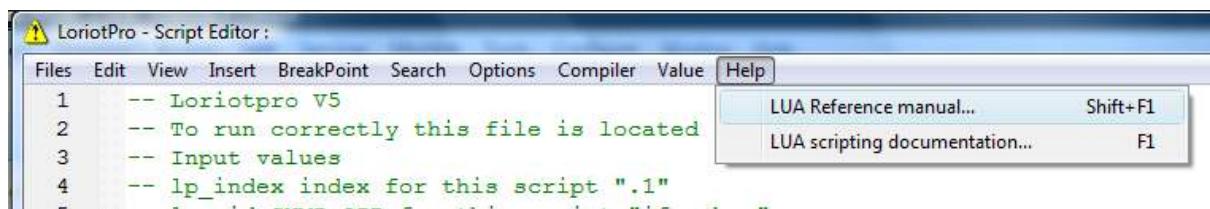


Le générateur de rapports est lancé et le fichier généré est affiché dans votre browser Web par défaut.

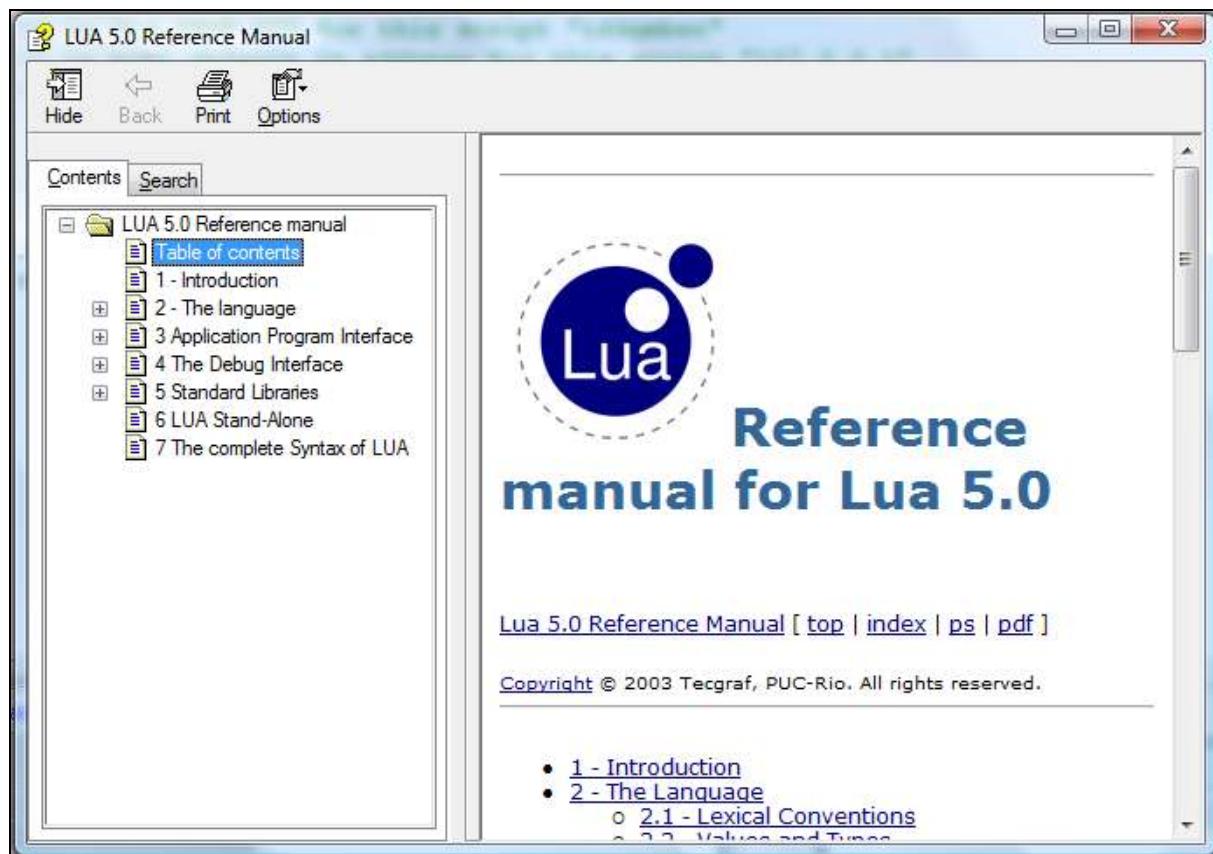


### 8.3.17 Menu Help

Ce menu permet d'avoir accès à la documentation de la syntaxe du langage LUA et la syntaxe des fonctions des librairies LoriotPro « Extended Edition »

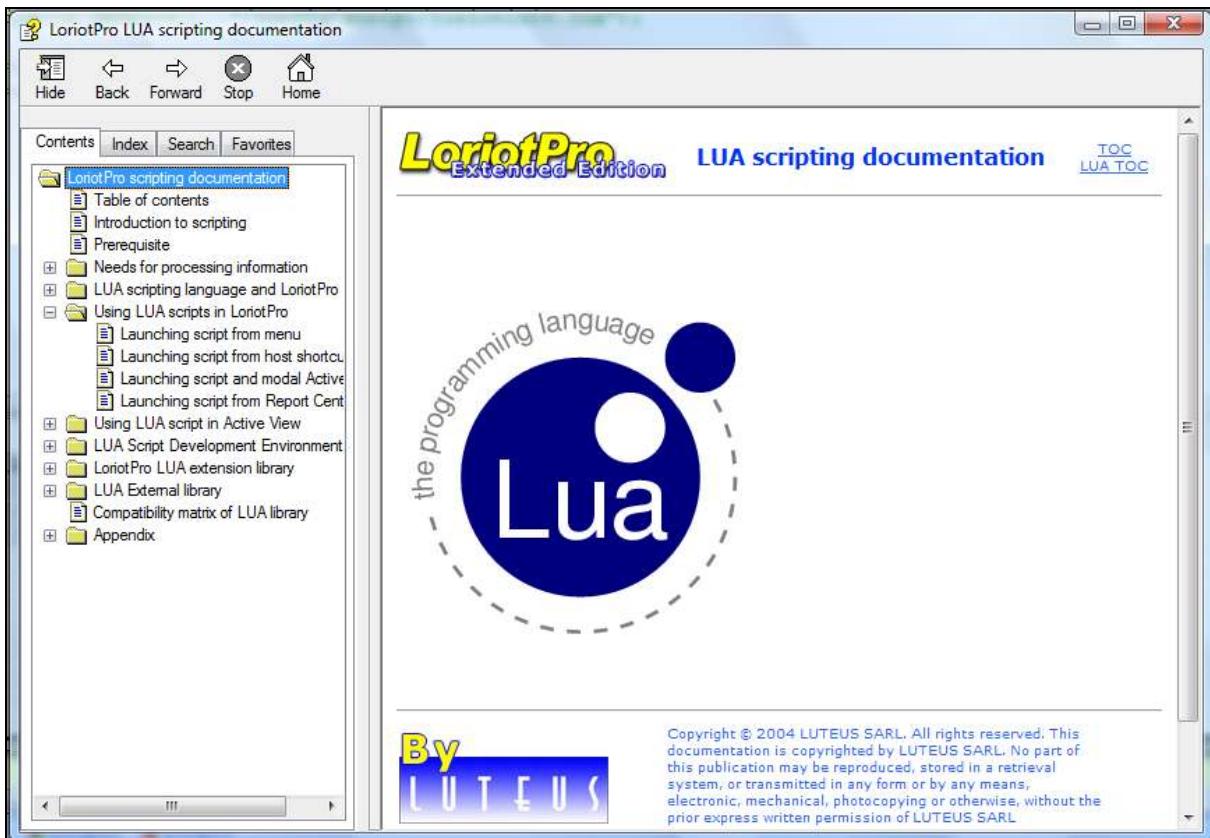


#### 8.3.17.1 Options - LUA Reference manual



### 8.3.17.2 Options - LUA Scripting documentation

Le help des fonctions LUA des librairies de LoriotPro « Extended Edition » en format CHM.



## 8.4 Zone d'édition des variables

Par défaut les scripts LUA LoriotPro utilisent 4 variables en entrée et 2 variables en sortie. Les scripts d'audit peuvent fournir au module d'audit des variables permettant la génération de graphes.

LoriotPro fournit au script les paramètres suivants:

Paramètres	Descriptions
lp_host	L'adresse IP de l'équipement sous la forme d'une chaîne de caractères

lp_oid	Le nom de l'objet snmp virtuel sous la forme d'une chaîne de caractères
lp_index	L'index associé à l'objet snmp virtuel sous la forme d'une chaîne de caractères incluant le point. « .2 »
lp_trace	1 active le mode trace, 0 le stoppe. Lorsque le script tourne en production, le mode trace est 0 par défaut.

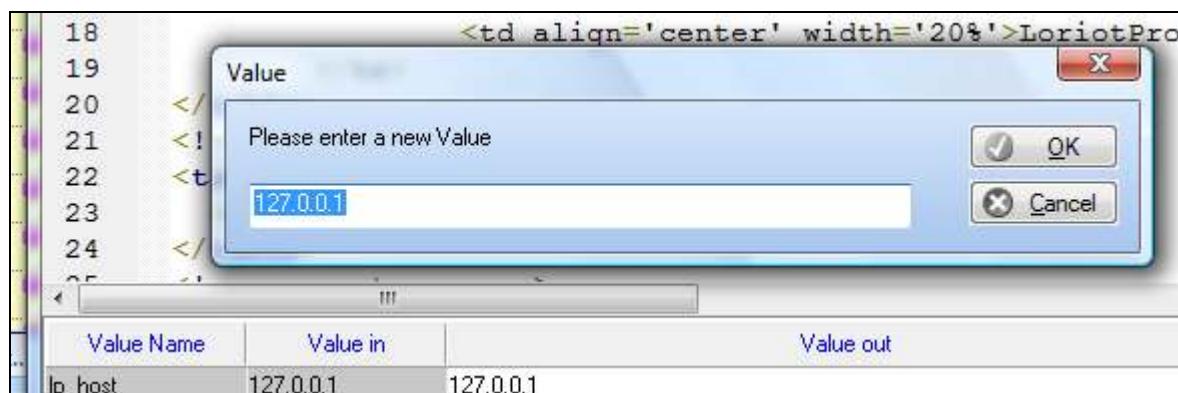
Le script fournit les paramètres suivant à LoriotPro :

Paramètres	Descriptions
lp_value	Une valeur (format double pour supporter le 64 bits) (si l'objet est de type « integer »)
lp_buffer	Une chaîne de caractères (si l'objet est de type string)
lp_graph	1 si un graphe est généré par ce script d'audit
lp_graph_unit	L'unité à utiliser pour le graphe
lp_graph_title	Le titre du graphe
lp_graph_type	Le graphe type (counter ou gauge) 0 ou 1
lp_graph_value	Le valeur renournée par le script d'audit au module d'audit.

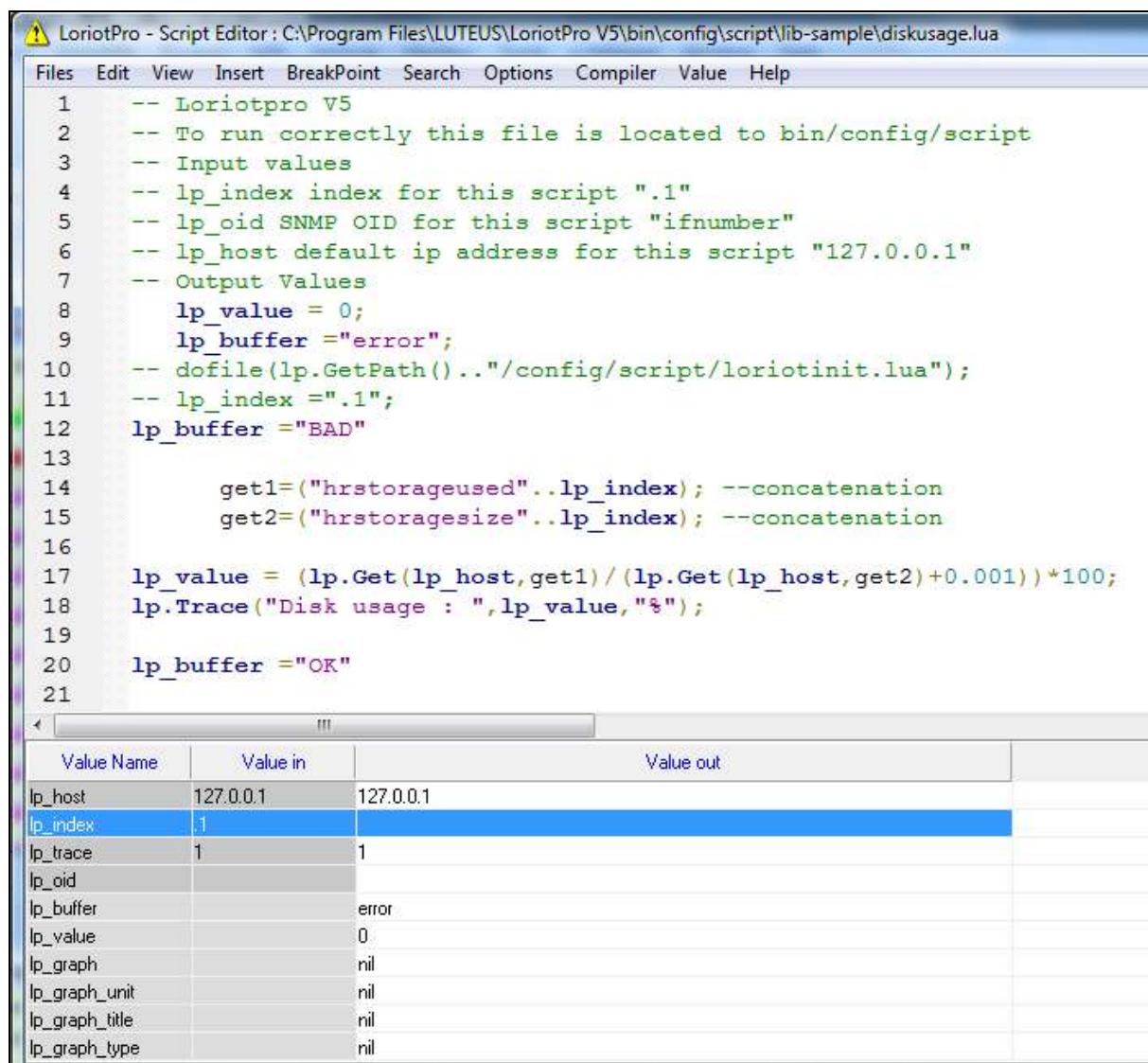
Il est possible de simuler ces variables dans la zone variable et de voir l'état de ces mêmes variables durant le déroulement du script.

Value Name	Value in	Value out
lp_host	127.0.0.1	127.0.0.1
lp_index		
lp_trace	1	1
lp_oid		
lp_buffer		error
lp_value		0
lp_graph		nil
lp_graph_unit		nil
lp_graph_title		nil
lp_graph_type		nil
lp_graph_value		nil

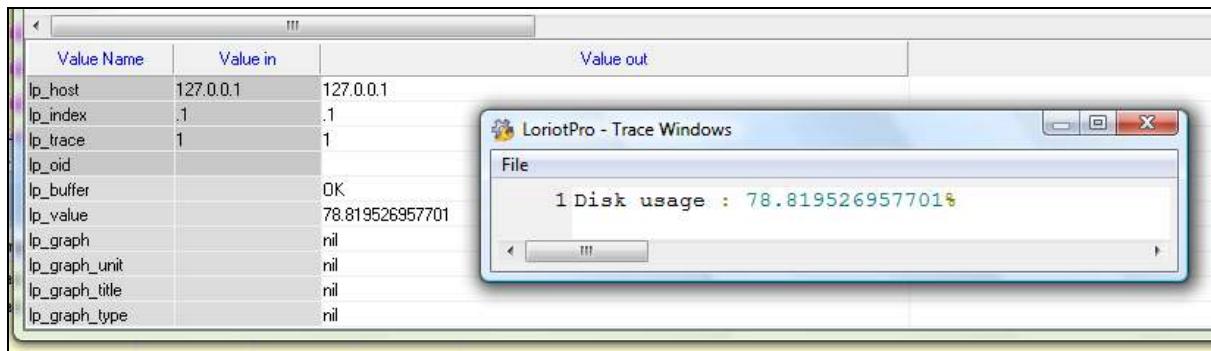
Pour changer la valeur en entrée d'une variable, il faut double-cliquer sur le champ contenant la valeur.



La touche F5 lance le script et les variables sont mises à jour dans la colonne « Value out » après l'exécution du script.



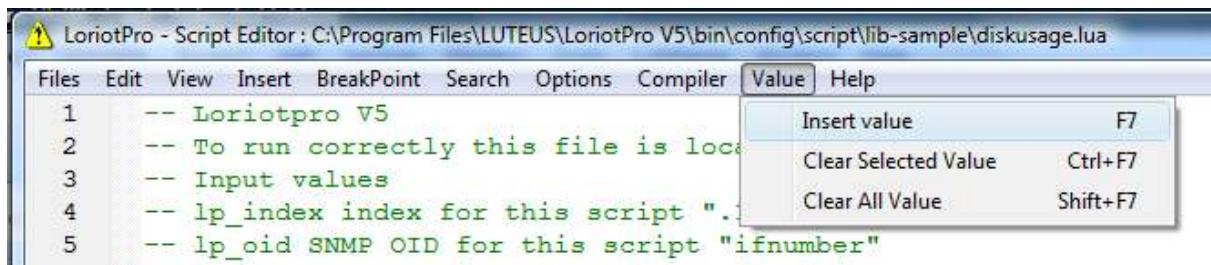
Avant



Après

#### 8.4.1 Menu Value

Ce menu permet de définir d'autres variables managées pour l'exécution du script



##### 8.4.1.1 Options - Insert Value

Dans cet exemple les variables value1 et value2 ont été ajoutées dans la zone de gestion des variables. A la fin de l'exécution du script, le champ « value out » de ces deux variables a été renseigné.

The screenshot shows the LoriotPro Script Editor interface. The top menu bar includes Files, Edit, View, Insert, BreakPoint, Search, Options, Compiler, Value, and Help. The main window displays a Lua script:

```
1 -- Loriotpro v5
2 -- To run correctly this file is located to bin/config/script
3 -- Input values
4 -- lp_index index for this script ".1"
5 -- lp_oid SNMP OID for this script "ifnumber"
6 -- lp_host default ip address for this script "127.0.0.1"
7 -- Output Values
8     lp_value = 0;
9     lp_buffer ="error";
10    -- dofile(lp.GetPath().."config/script/loriotinit.lua");
11
12    value1=2
13    value2=4
14
```

Below the script is a table titled "Value" with columns "Value Name", "Value in", and "Value out". The table contains the following data:

Value Name	Value in	Value out
lp_graph		nil
lp_graph_unit		nil
lp_graph_title		nil
lp_graph_type		nil
lp_graph_value		nil
value1	2	
value2	4	

#### 8.4.1.2 Options - Clear All Value

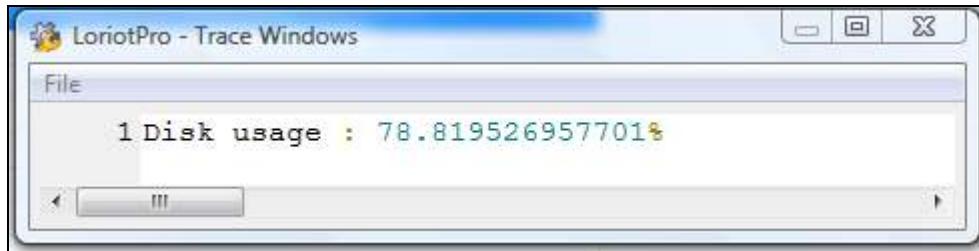
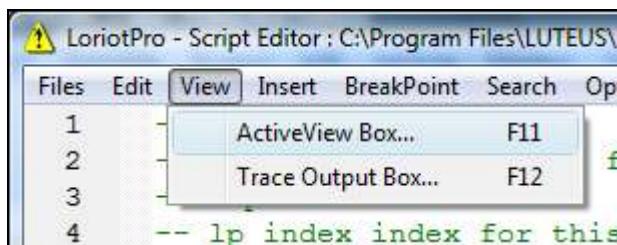
Elle efface toutes les variables non système préalablement définies par l'administrateur.

The screenshot shows the "Value" table from the previous screenshot, but now all rows except the header have been cleared, resulting in empty "Value in" and "Value out" columns.

Value Name	Value in	Value out
lp_host	127.0.0.1	
lp_index		
lp_trace	1	
lp_oid		
lp_buffer		
lp_value		
lp_graph		
lp_graph_unit		
lp_graph_title		
lp_graph_type		
lp_graph_value		

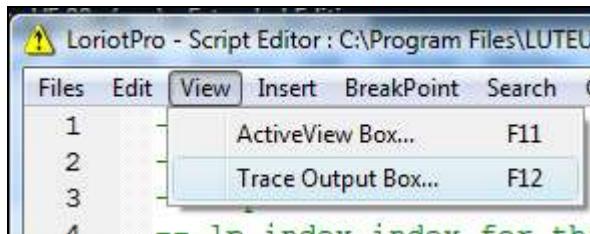
## 8.5 Fenêtre des messages systèmes

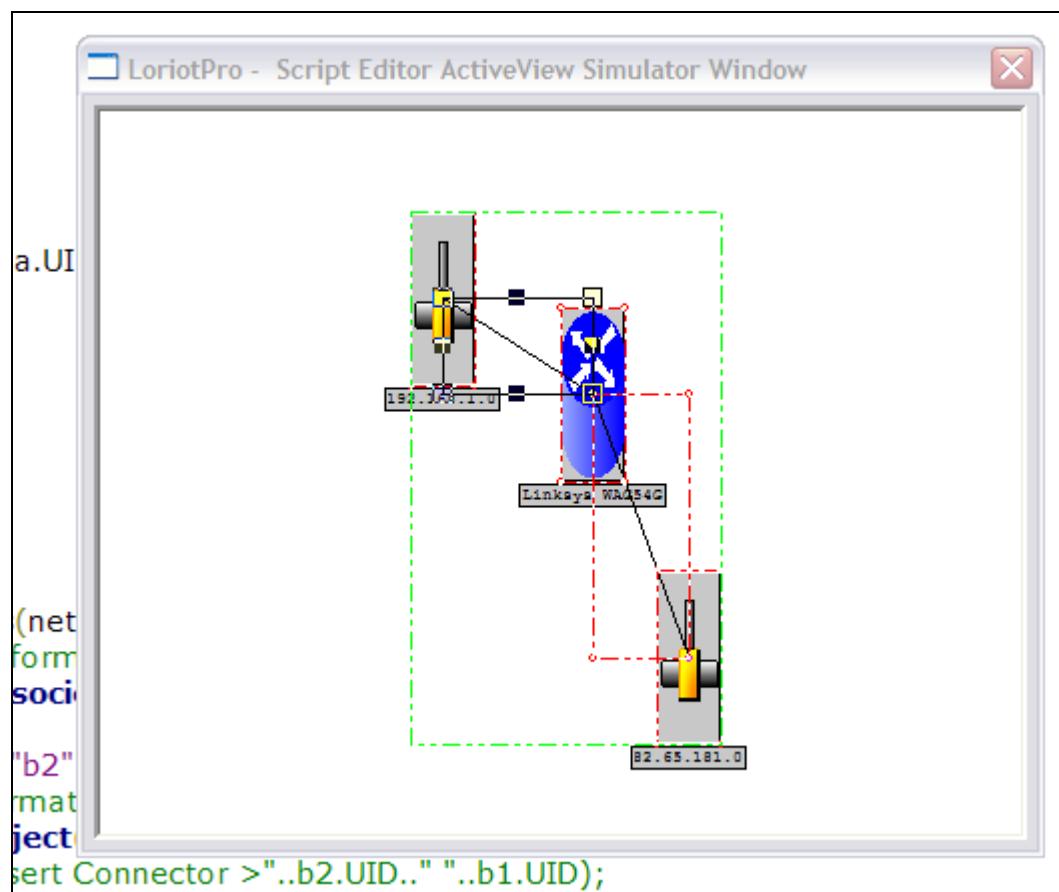
De base, un script n'est pas fait pour afficher du texte mais uniquement pour réaliser un traitement, il est pourtant indispensable de pouvoir afficher des commentaires durant la réalisation d'un script pour aider au « débogage ». La librairie offre des fonctions d'affichage de texte dans cette zone (**Ip.Trace(« text »)**) .



## 8.6 Fenêtre d'affichage des graphiques (ActiveView)

Cette zone permet de simuler une ActiveView, un grand nombre de fonctions de la librairie **Ipav** utilise comme variables des objets vectoriels d'ActiveView.





## 9 Le module IP Expert

### 9.1 Introduction

La version 5 de LoriotPro intègre un nouveau module attaché au noyau du logiciel qui permet de réaliser des scans élaborés des équipements découverts par le module Discover déjà présent dans les versions précédentes du logiciel. Le but du module IP Expert est de finaliser et d'automatiser la configuration des équipements détectés. Ce module est évolutif et constitué d'un ensemble structuré de scripts de scans permettant une configuration très fine des équipements. Chaque script de scan dispose d'une logique de configuration simple permettant à l'administrateur de sélectionner le type de configuration qu'il désire. Il est possible de réaliser ses propres scripts de scan avec ses propres options qui viendront s'ajouter aux scripts fournis par défaut avec le logiciel.

Le module IP Expert est contrôlé par le module Discover qui l'utilise pour réaliser des scans de ports TCP et lancer des scripts sur les équipements découverts dans un but d'audit et d'auto-configuration. L'administrateur peut contrôler les paramètres d'exécution du module IP Expert avec une grande finesse mais le lancement de ce module est réalisé lors des phases d'exécution du module Discover. Pour une utilisation unitaire de ce processus, il est possible d'utiliser le module IP Scanner qui offre les mêmes fonctionnalités avec des options supplémentaires.

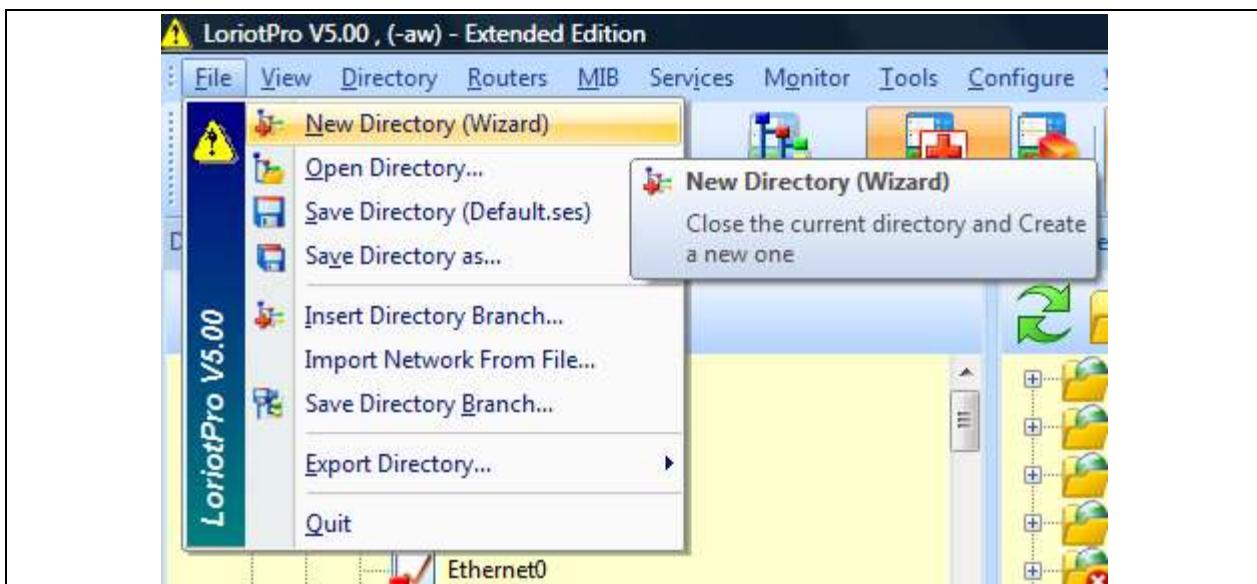
### 9.2 Logique de fonctionnement du module IP Expert

Le module IP Expert est contrôlé par le module Discover qui lui-même peut être lancé à travers un Wizard de découverte. Pour une meilleure compréhension du fonctionnement et de l'enchaînement des différents modules, les chapitres suivants détaillent le principe de fonctionnement en partant d'une Directory vide et d'une utilisation du Wizard de découverte des ressources.

### 9.3 Découverte de topologie réseau et des ressources

#### 9.3.1 Introduction

LoriotPro intègre un WIZARD de Découverte associé à un système expert ouvert permettant d'aider à la découverte et la configuration de son environnement. Le module est lancé automatiquement après l'installation du logiciel ou est accessible via le menu.



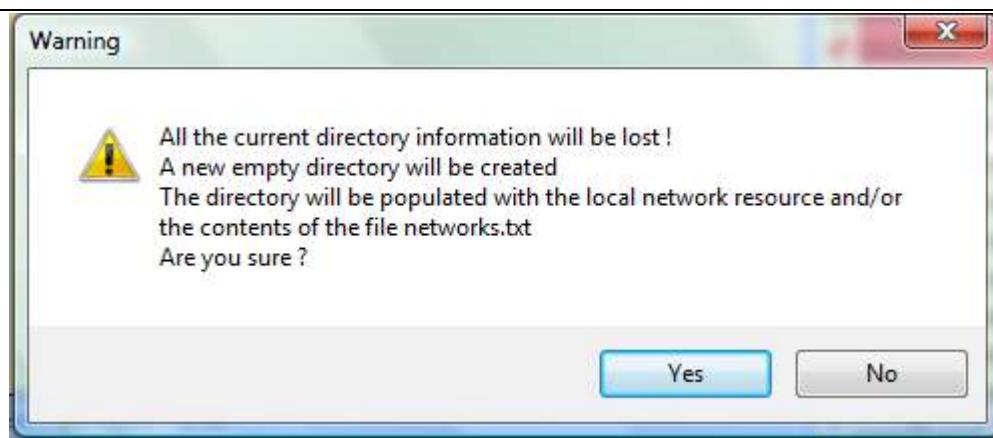
Le processus de découverte analyse la structure du système d'information et en construit une image fidèle. Pour ce faire, LoriotPro V5 intègre des stratégies de découverte et d'auto configuration des équipements détectés. Il est possible de positionner automatiquement des scripts d'audit sur des équipements pré ciblés. La version 5 contient plusieurs modules de « scan » interdépendants.

- Discover (scan furtif basé sur SNMP)
- Discover IP Expert (scan force brute basé sur ICMP, SNMP, TCP) contrôlé par le module Discover et utilisant des stratégies configurables
- En version « Extended Edition » IP Expert dispose d'un module de scan scripté (open source) pour autoconfigurer et/ou auditer les équipements.
- IPScanner (scan force brute basé sur ICMP, SNMP, TCP, script LUA)
- En Extended Edition, il est possible de développer des scripts (LUA) de scans qui interagissent avec la Directory et les moteurs d'événements du logiciel.

### 9.3.2 Lancement du Wizard de découverte

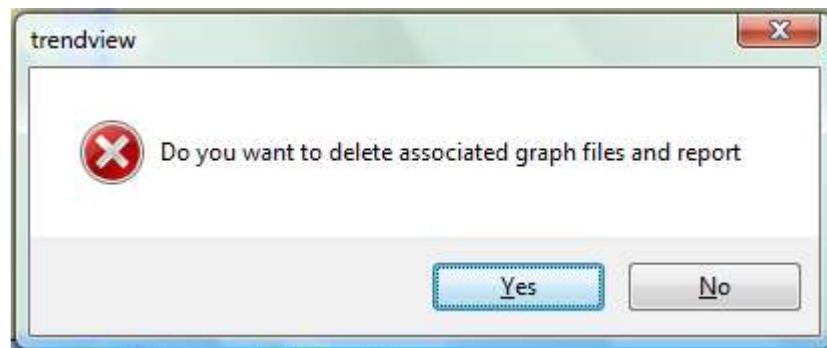
#### Attention

Si vous utilisez le module Wizard sur une Directory existante, le module efface au préalable l'ancienne Directory. Prévoyez de faire une sauvegarde de votre Directory avant d'utiliser cette option.



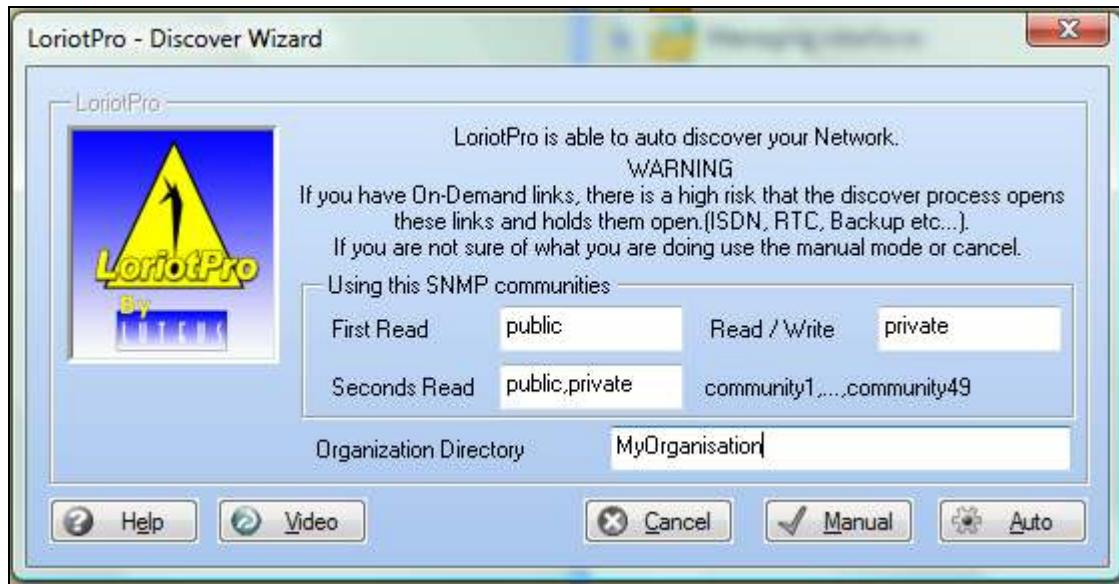
Vous pouvez réaliser une sauvegarde de votre Directory avant l'opération et procéder à un insert de votre ancienne Directory dans la nouvelle si vous le désirer.

Certains plugins vous demanderont si vous désirez nettoyer les fichiers de datas associés, si vous désirez conserver l'intégrité de votre ancienne Directory ; il faut répondre No.



Au lancement du Wizard, quelques paramètres optionnels sont demandés pour faciliter la découverte.

- Les communities SNMP utilisées par l'entreprise
- le nom de l'entreprise



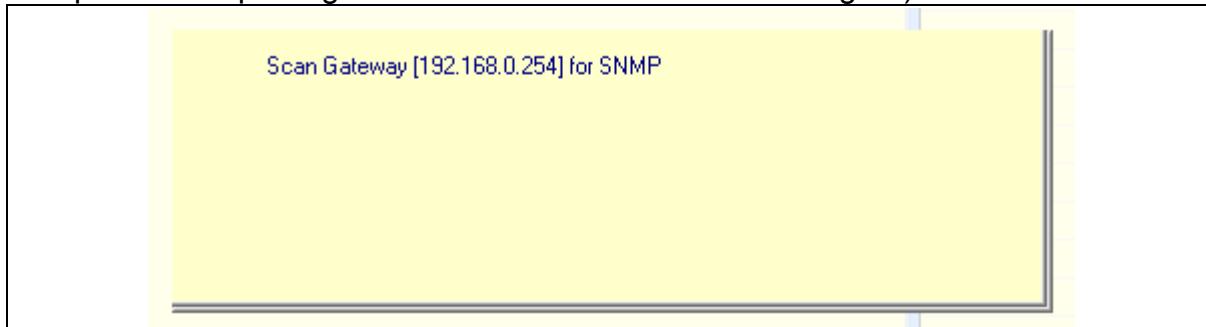
Si les équipements de votre entreprise dispose de plusieurs communities SNMP vous pouvez les renseigner dans le champ « Seconds Read » en les séparant par des virgules (ne mettez pas de caractère blanc dans la liste des communities).

Choisir l'options Auto pour poursuivre notre démonstration.

### 9.3.3 Collecte local de donnée

Le processus d'auto découverte suit la logique suivante :

- (1) Collecte des ressources locales à l'équipement où est installé le logiciel. (il est préférable que l'agent SNMP local soit installé et configuré)

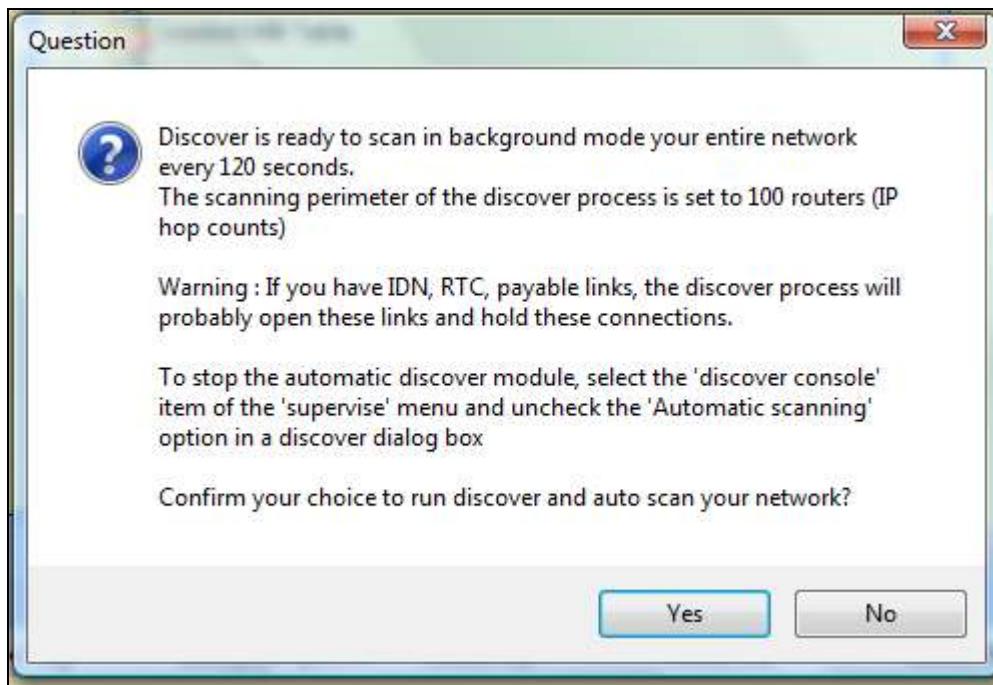


- Collecte des adresses IP et interfaces locales
- Collecte de la Gateway par défaut
- Collecte des tables de routage

- Collecte de la table ARP locale
- Recherche d'information sur les Next-Hop, la Gateway et les équipements connectés au PC local. Ce processus utilise SNMP mais les ressources locales peuvent être détectées même si le service SNMP n'a pas démarré.

### 9.3.4 Lancement des modules de découvertes

(2) L'Auto découverte va être lancée



- Les données collectées durant la phase 1 seront utilisées pour découvrir le reste des équipements.

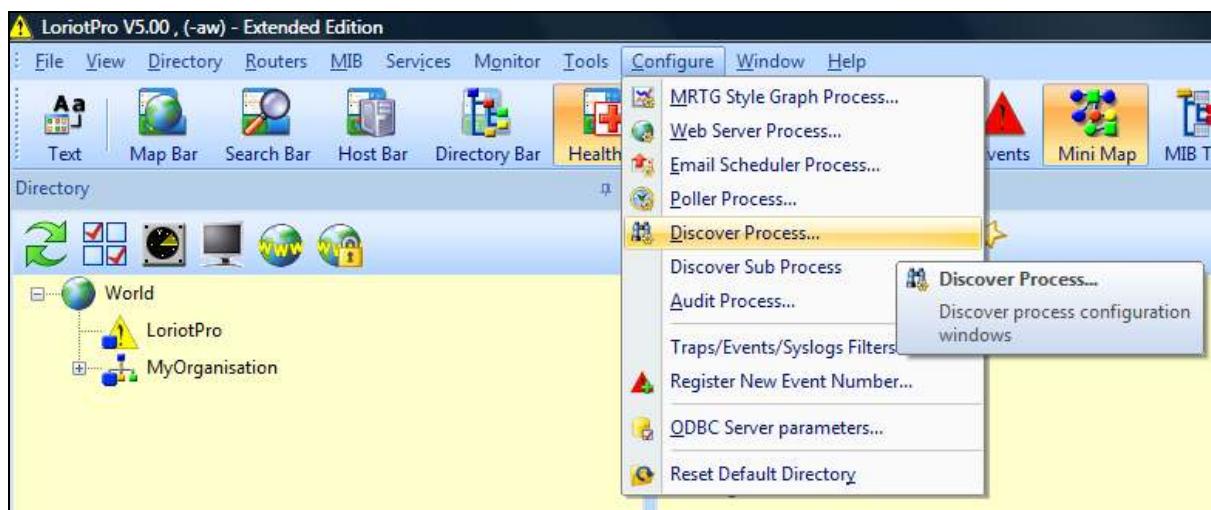
(3) Les modules Discover et IP Expert sont lancés en tâche de fond

#### 9.3.4.1 Module Discover

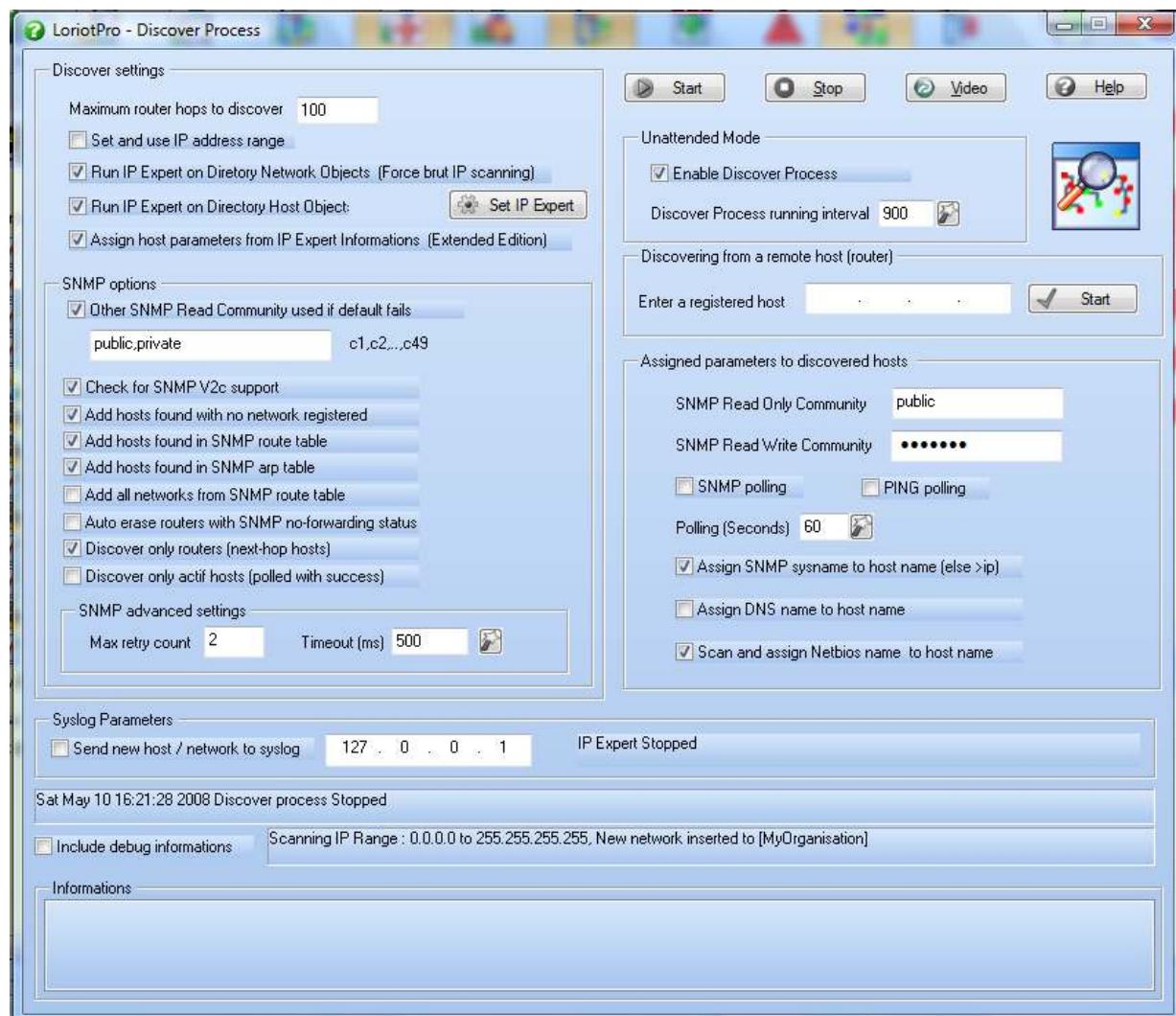
Discover collecte des données SNMP sur les next-hop (routeur), construit et pré-configurera la Directory à partir des informations collectées.

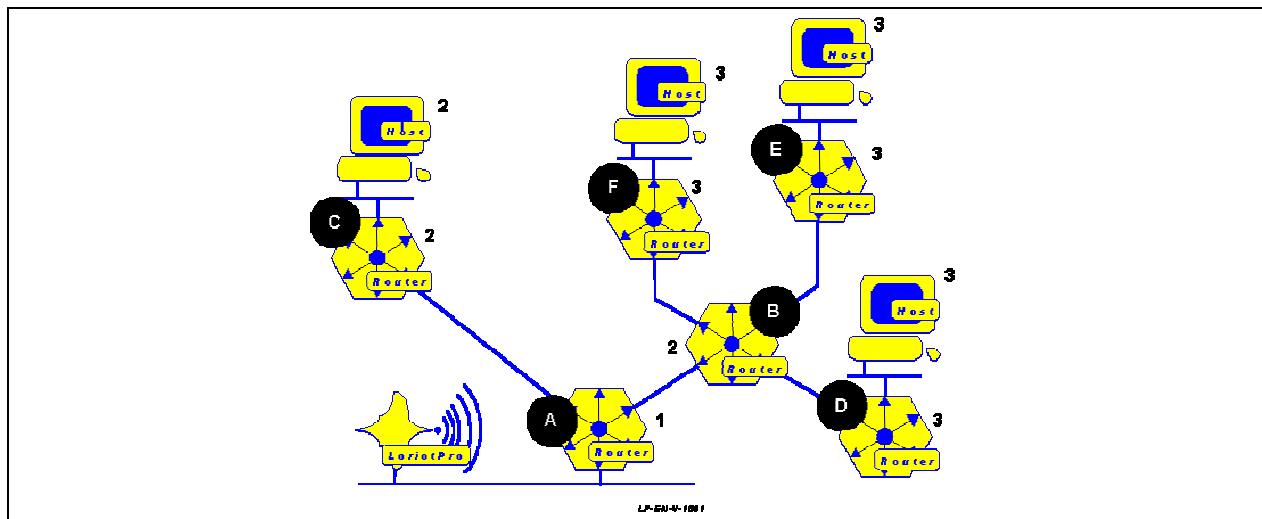
- Collecte des adresses IP et interfaces locales
- Auto-configuration du routeur (dans la Directory)
- Collecte des tables de routage
- Découverte des networks et routeurs (next-hop)
- Collecte de la table ARP du routeur
- Découverte furtive des équipements du réseau (bypass des firewall locaux)
- Recherche d'information sur les Next-Hop

- Extension de la découverte par (hop)



Il est possible de configurer et visualiser les paramètres du module Discover (voir la documentation du logiciel).





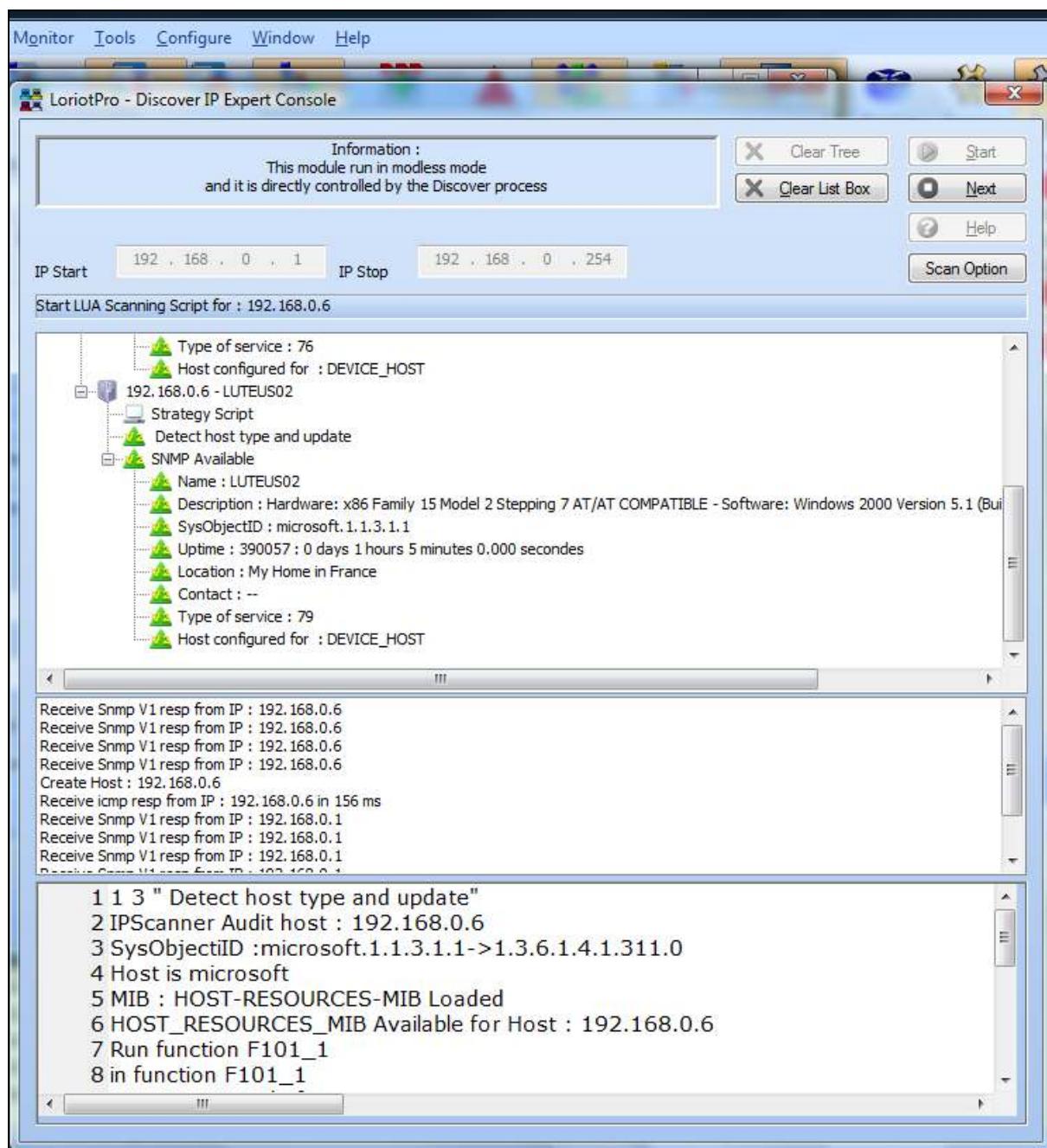
Les équipements et les réseaux sont découverts à travers les tables SNMP collectées sur les routeurs d'interconnexion du réseau d'entreprise. La découverte est donc réalisée par hops à partir du premier routeur inséré dans la Directory. A chaque cycle de balayage des ressources, le module Discover « avance » dans la découverte de l'environnement.

#### 9.3.4.2 Module IP Expert

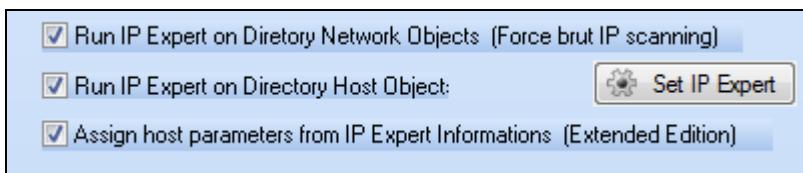
IP Expert (lancer par discover) collecte des données sur les équipements détectés et cherche de nouveaux équipements sur les networks découverts. Configure les éléments de la Directory et positionne des audits (Extended Edition).

- (1) Scan ICMP SNMP force brute sur les networks découverts
  - Découvre d'autres équipements
- (2) Scan TCP sur les équipements découverts
  - Détermine les services disponibles sur les équipements
- (3) Audit des équipements découverts (Extended Edition)
  1. Classifie les équipements et leur appliquent des stratégies.
- (4) Auto-Configuration des équipements par le système Expert (Extended Edition).

LoriotPro V5.00 Extended Edition – Supervision et gestion SNMP par scripts LUA  
Introduction au System Expert et aux scripts



Le module Discover fournit les paramètres de contrôle du module IP Expert.



#### **9.3.4.2.1 Options - Run IP Expert on Directory Networks**

Discover lance un scan force brute sur l'ensemble des équipements des réseaux trouvés dans les tables SNMP des routeurs.

Par exemple, si discover trouve le réseau 192.168.1.0 / 255.255.255.0 dans la table de routage d'un équipement et que cette option est cochée, alors IP Expert fera un scan de découverte des équipements 192.168.1.1 à 192.168.1.254. L'objectif de ce process est de détecter de nouveaux équipements, simplement en faisant des requêtes ICMP et SNMP de base.

#### **9.3.4.2.2 Options - Run IP Expert on Directory Hosts Object**

Après les scans force brute réalisés (ou pas) sur les réseaux fournis par le module Discover, Discover lance un IP Expert sur chaque équipement de la Directory si cette option est cochée.

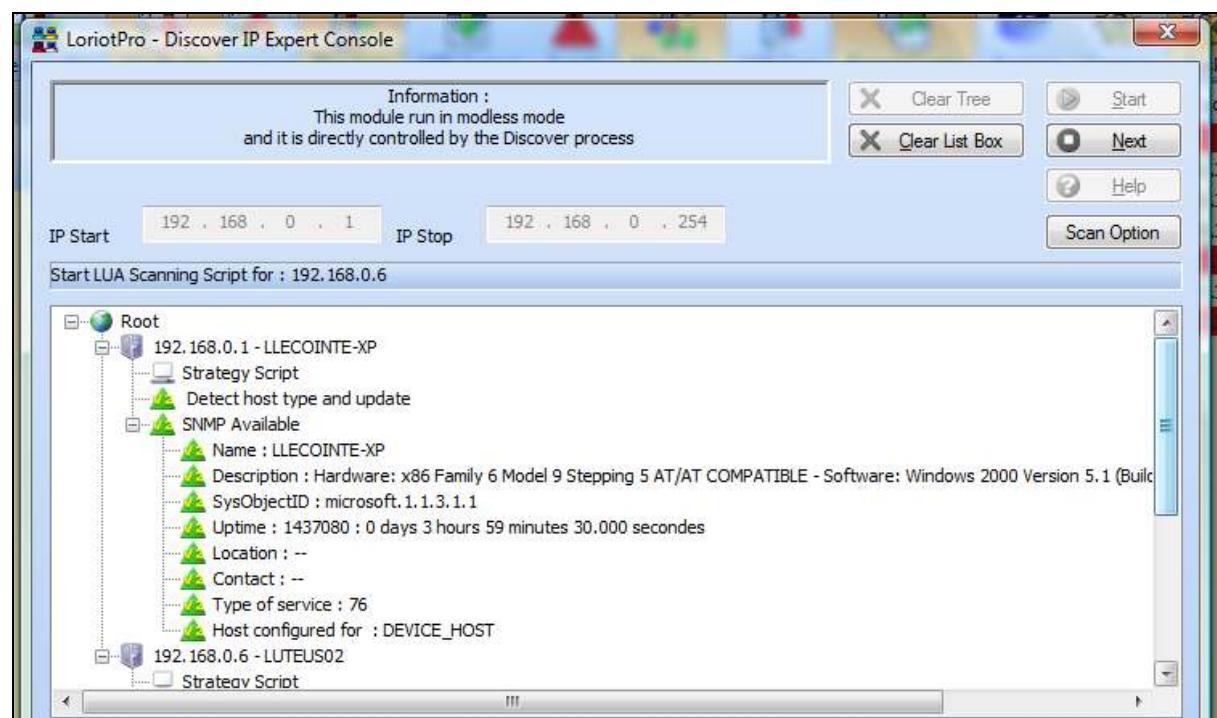
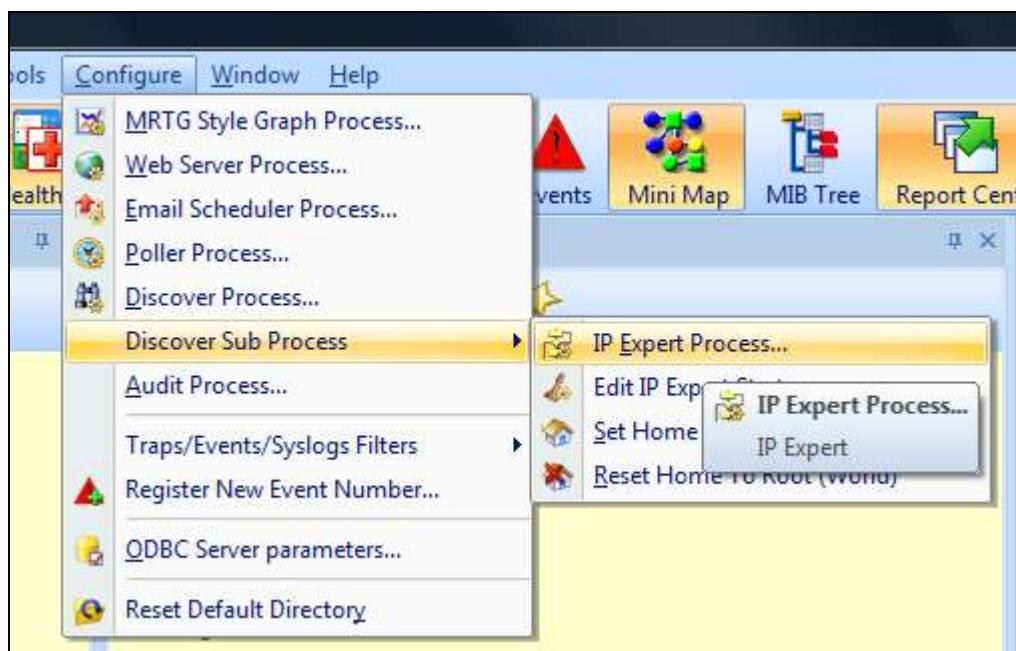
Lors de cette phase, IP Expert utilise les paramètres IP Expert associés à l'équipement concerné par le scan d'expertise. On peut associer manuellement ou par script à tout équipement de la Directory une stratégie d'expertise. De base, le module utilise une stratégie globale qui permet de configurer des stratégies spécifiques sur les équipements en fonction des résultats collectés de façon automatique.

#### **9.3.4.2.3 Options - Assign Host parameters from IP Expert Informations**

Si cette case est cochée, le script est informé que l'administrateur désire une configuration automatique des équipements. Ce paramètre est global. Il est possible de « bypass » ce paramètre pour un équipement ou un réseau en utilisant les paramètres associés aux réseaux ou aux équipements. Voir pour cela la documentation détaillant les options associées à ces types d'objets.

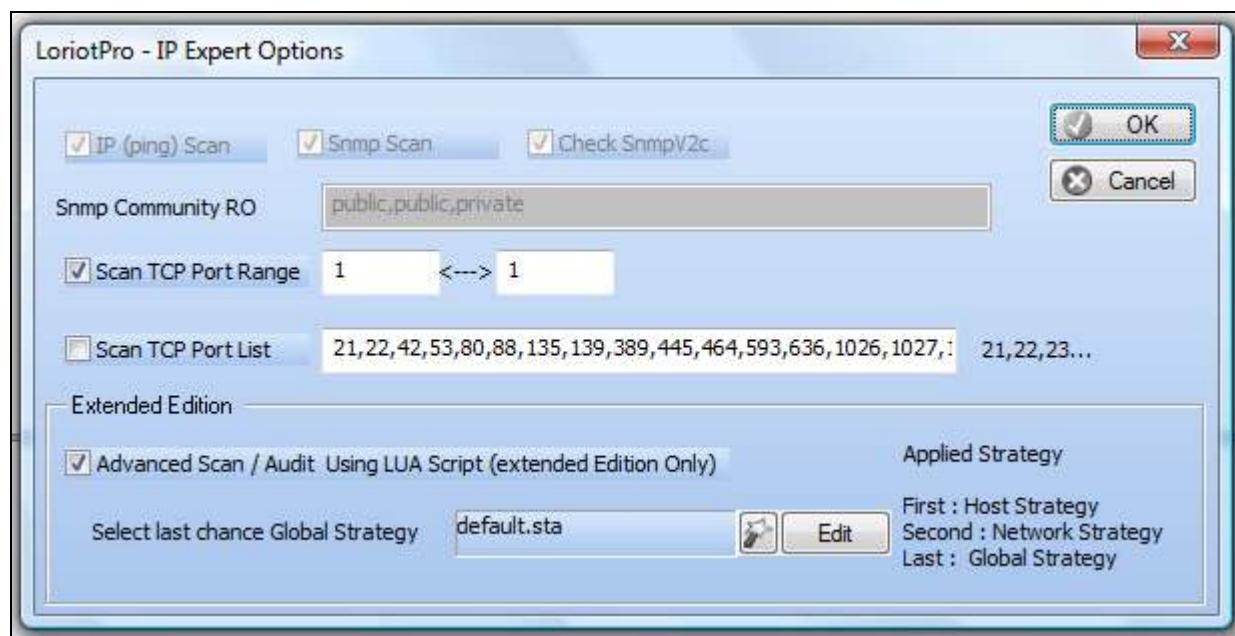
#### **9.3.4.3 Paramètres du module IP Expert**

Le module IP Expert est contrôlé par le module Discover mais il est possible de configurer certains paramètres de base de ce module.



La fenêtre de configuration est lancée par le bouton « Scan Option ».

#### 9.3.4.4 Fenêtre - Scan Option



Les paramètres en grisé sont les mêmes que ceux définis dans le module Discover.

#### 9.3.4.5 Options - Scan TCP

Les options « Scan TCP port Range » et « Scan TCP Port List » permettent de générer un scan TCP sur l'équipement avant de lancer les scripts d'expertise. Les scripts pourront avoir accès au résultat du scan TCP pour réaliser des opérations d'auto-configuration ou autres.

Un script de type IP Expert peut forcer un scan TCP en utilisant les paramètres associés au script (voir la partie création d'un script IP Expert).

##### 9.3.4.5.1 Script IP Expert et stratégie associées

La case « Advanced Scan / Audit Using Script » permet de sélectionner une stratégie d'utilisation des scripts de type IP Expert.

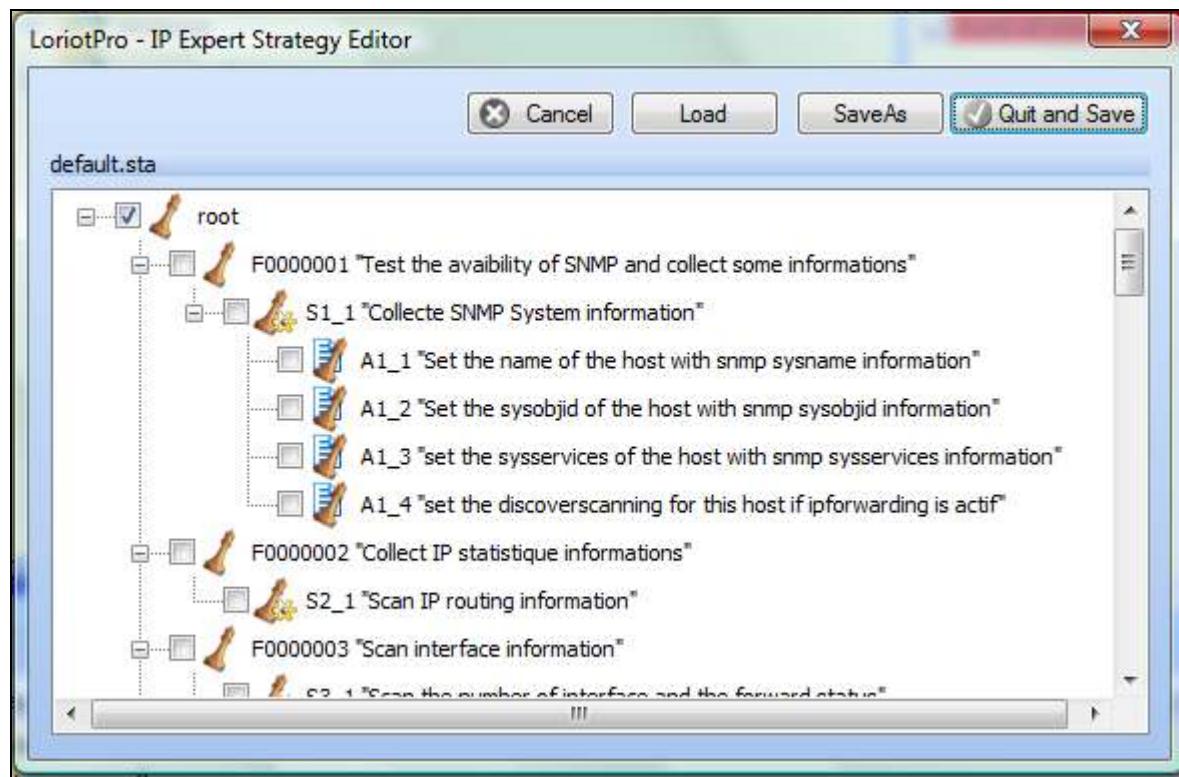
Les « scans » sont réalisés à partir de scripts situés dans le répertoire **bin/config/script/ipsscanner**. Les scripts sont open source et peuvent être modifiés avec l'éditeur de script fourni avec le logiciel. Chaque fichier de script contient des fonctions diverses d'audits, d'expertises, de configurations etc.. Les fonctions incluses dans les scripts sont structurées et auto répertoriées par le logiciel. LoriotPro fournit un module de gestion de ses fonctions et permet de définir des stratégies d'exploitation associables à un équipement ou un network. Basiquement le module « IP Expert » utilise ces fonctions pour auto configurer les équipements et réaliser quelques audits basiques.

Le module « IP Expert » utilise la stratégie « **default.sta** » que vous pouvez bien entendu modifier selon les besoins de votre entreprise. Ou bien vous pouvez créer une stratégie différente à partir des outils de gestion.

**Attention** 

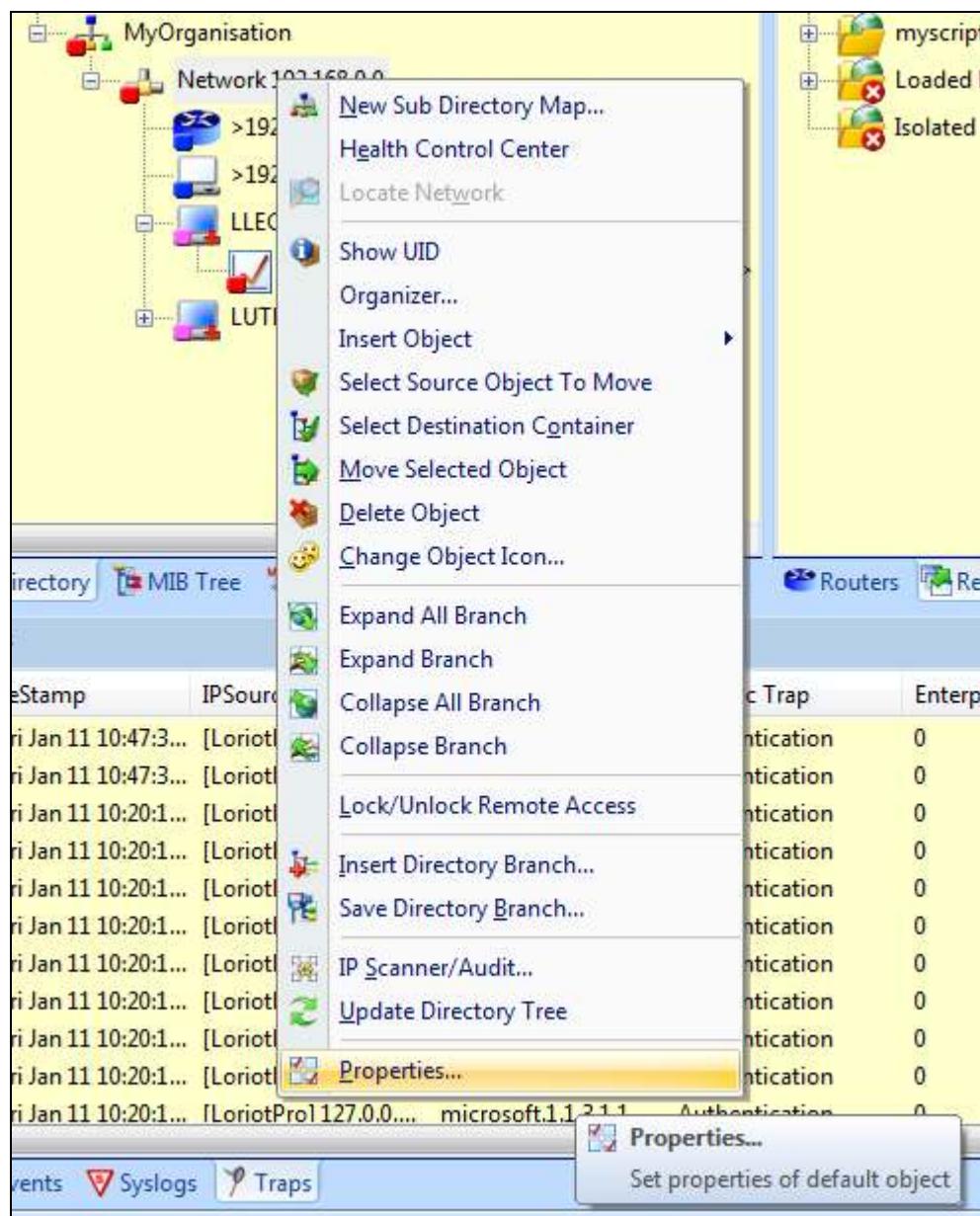
Si un équipement ou un réseau possède une stratégie spécifique, le module utilisera en priorité la stratégie qui lui est associé dans ses paramètres ou bien la stratégie associée au réseau de l'équipement si elle est configurée.

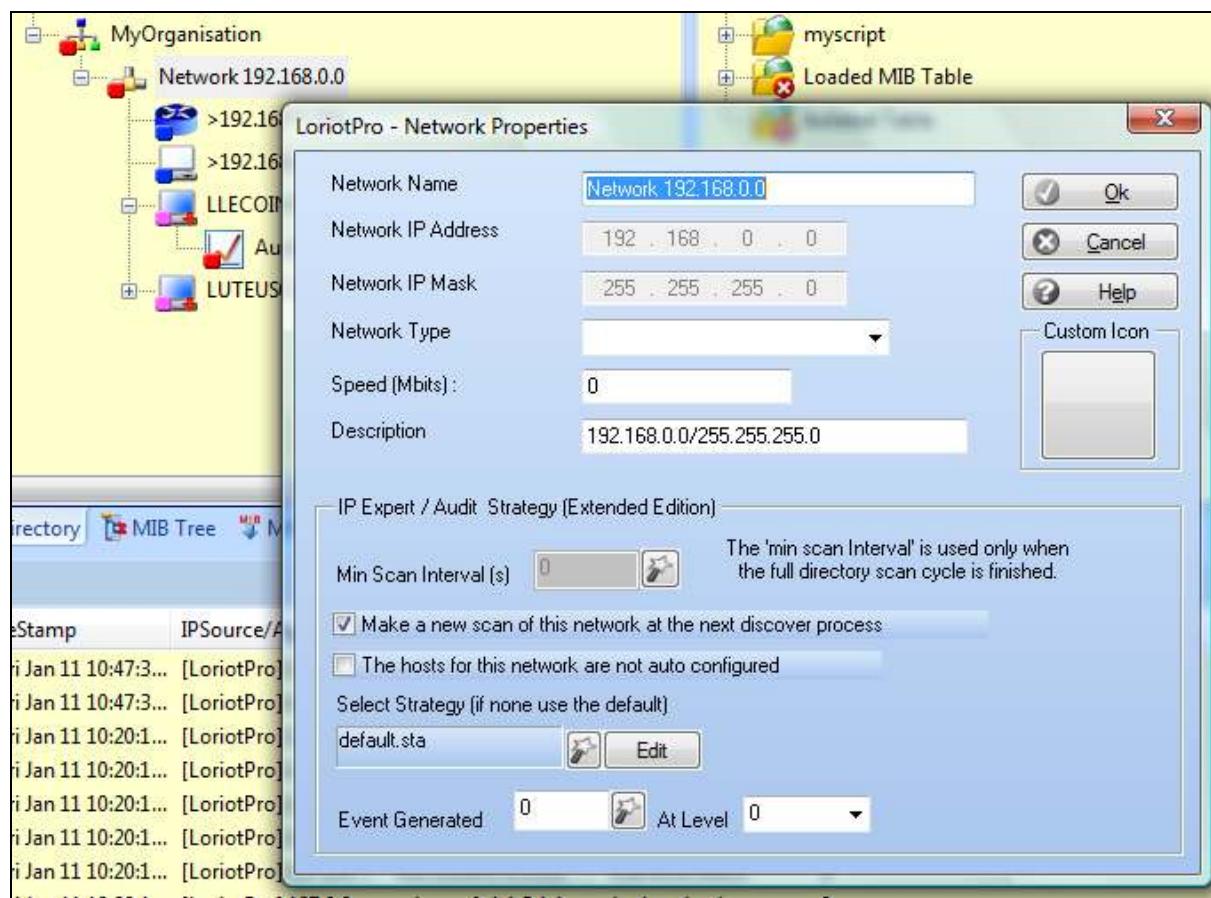
Le bouton « Edit » permet d'afficher la boîte de dialogue de sélection de votre stratégie « IP Expert »



#### 9.3.4.6 Stratégie IP Expert associé au réseaux

Il est possible de modifier les stratégies utilisées par le module Discover / IP Expert pour les équipements d'un réseau donné. Pour cela, il faut utiliser la boîte de propriété d'un réseau à partir de la Directory.





Les paramètres de stratégie « IP Expert » associés à un réseau sont prioritaires sur les paramètres généraux utilisés par le module Discover.

#### 9.3.4.6.1 Options - Min Scan Interval

Le module Discover peut tourner en tâche de fond à intervalles réguliers, cette option permet de définir l'intervalle de temps entre deux Discovers sur ce réseau.

##### Attention

SI le module Discover n'est pas en mode tâche de fond, cette option n'est pas utilisée.

Si le module Discover est programmé pour tourner toutes les 24 heures (ou autre valeur de temps) en tâche de fond et que ce paramètre possède une valeur inférieure, alors l'intervalle de scan du module Discover pour ce réseau sera celui du module Discover.

Si le module Discover est programmé pour tourner toutes les 5 minutes (ou autre valeur de temps) en tâche de fond et que ce paramètre possède une valeur supérieure, alors l'intervalle de scan du module Discover pour ce réseau sera celui défini ici modulo l'intervalle du module Discover.

Le module Discover traite les réseaux dans l'ordre de sa structure chaînée.

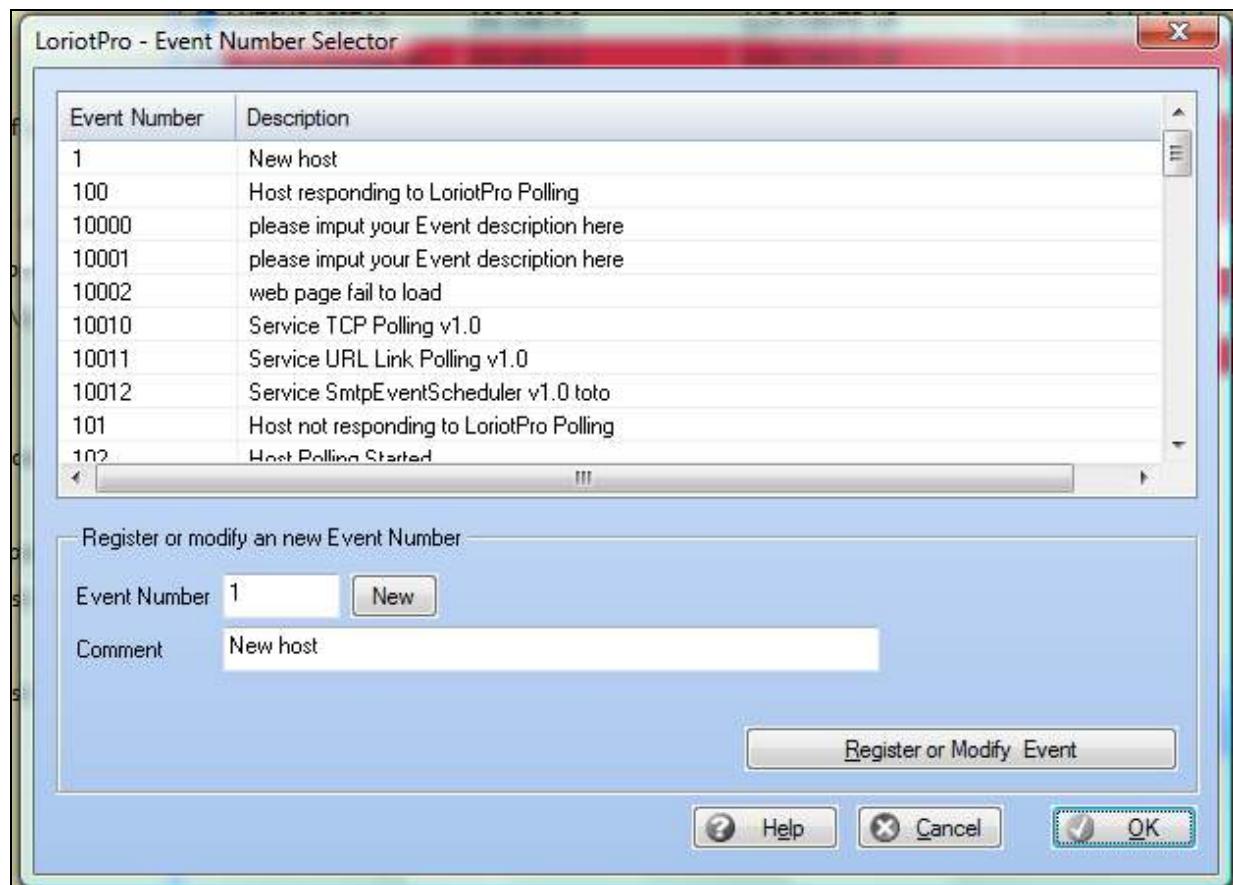
Si le temps de traitement des réseaux présents dans la Directory est plus long que le « Min Scan interval » alors cette valeur pourra ne pas être honorée et le scan du réseau se fera à la suite des autres (il n'y a pas de priorité et le module travaille dans un mode « best effort »).

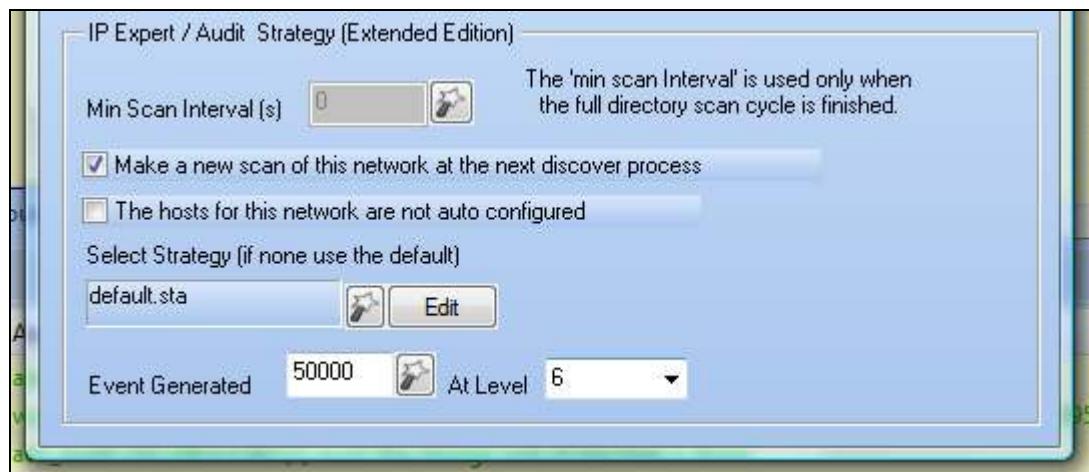
### 9.3.4.6.2 Options - Event Generated

Les scripts « IP Expert » peuvent réaliser des audits ou détecter des anomalies ; dans ce cas, par défaut un Event 600 est envoyé à LoriotPro en level 2.

Events				
TimeStamp	LoriotPro ...	Ref N°	IP Ref	Alert
▲ Fri Jan 11 11:28:10 2...	Local	600	[LUTEUS02]192.168.0.6	adv_audit: 192.168.0.6 100(1) audit the storage unit of the host is down
▲ Fri Jan 11 11:28:10 2...	Local	600	[LUTEUS02]192.168.0.6	warning host [192.168.0.6] as low disk c:\ labelboot serial number c4cefa82 : reached 86% 65963/75995 mo
▲ Fri Jan 11 11:26:10 2...	Local	600	[LLECOINTE-XP]192.168...	adv_audit: 192.168.0.1 100(4) audit the storage unit of the host is down
▲ Fri Jan 11 11:26:10 2...	Local	600	[LLECOINTE-XP]192.168...	warning host [192.168.0.1] as low disk c:\ labelwindows serial number 2e2241b : reached 95% 14616/15335 mo
▲ Fri Jan 11 11:22:40 2...	Local	600	[LUTEUS02]192.168.0.6	adv_audit: 192.168.0.6 100(2) audit the storage unit of the host is down
▲ Fri Jan 11 11:22:40 2...	Local	600	[LUTEUS02]192.168.0.6	warning host [192.168.0.6] as low disk c:\ labelboot serial number c4cefa82 : reached 86% 65963/75995 mo
▲ Fri Jan 11 11:21:05 2...	Local	600	[LLECOINTE-XP]192.168...	adv_audit: 192.168.0.1 100(0) audit the storage unit of the host is down
▲ Fri Jan 11 11:21:05 2...	Local	600	[LLECOINTE-XP]192.168...	warning host [192.168.0.1] as low disk c:\ labelwindows serial number 2e2241b : reached 95% 14616/15335 mo
▲ Fri Jan 11 11:17:36 2...	Local	600	[LUTEUS02]192.168.0.6	adv_audit: 192.168.0.6 100(0) audit the storage unit of the host is down

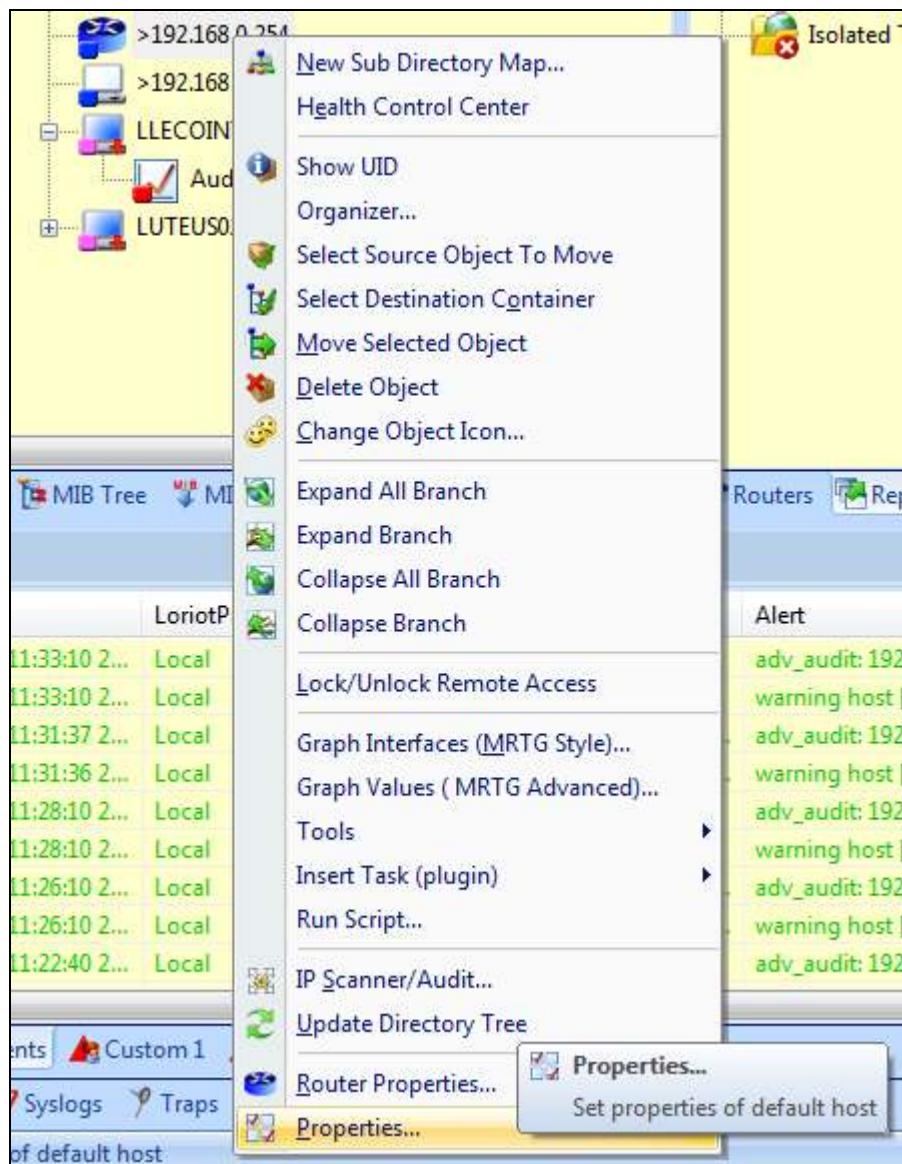
Il est possible de modifier cette valeur pour isoler les Events IP Expert / Audit d'un Host ou d'un réseau donné.

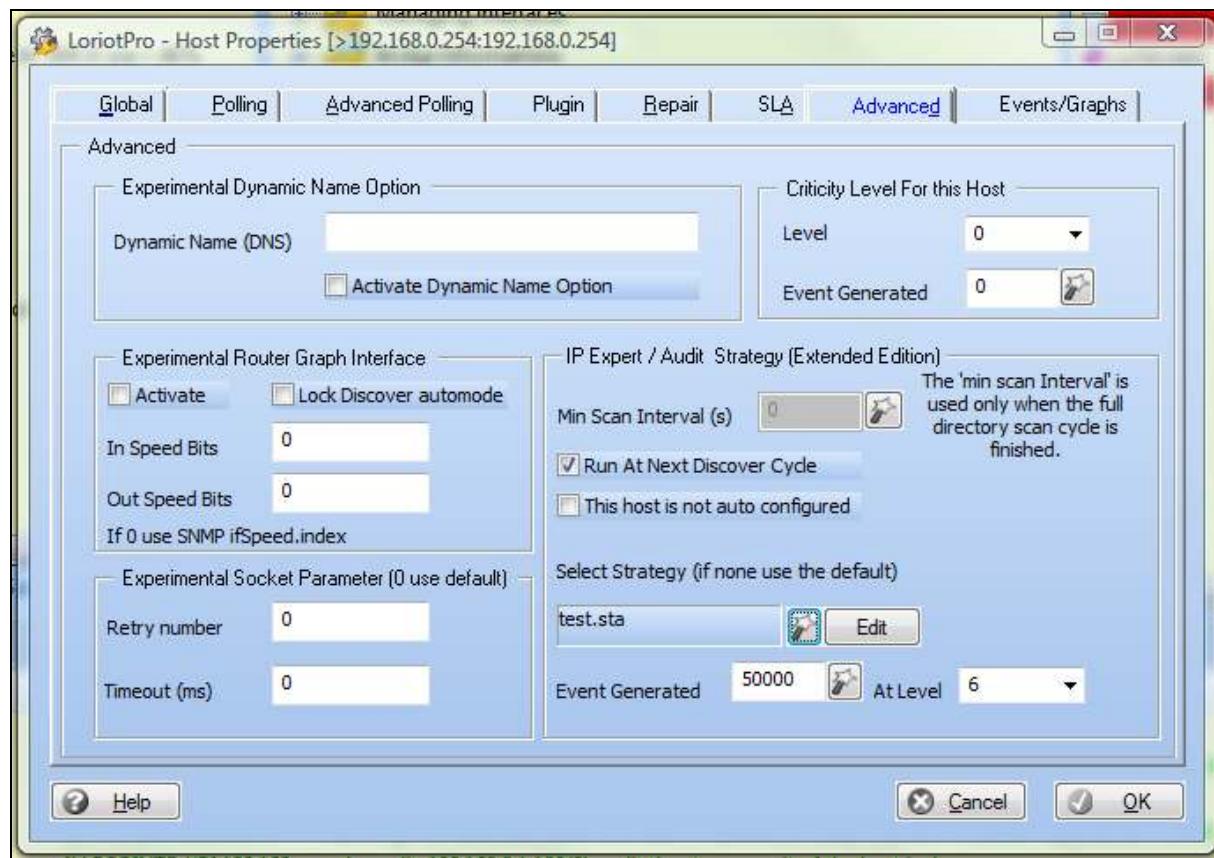




#### 9.3.4.7 Stratégie IP Expert associé aux équipements

Il est possible de modifier les stratégies utilisées par le module Discover / IP Expert pour les équipements. Pour cela, il faut utiliser la boîte de propriété d'un équipement à partir de la Directory.

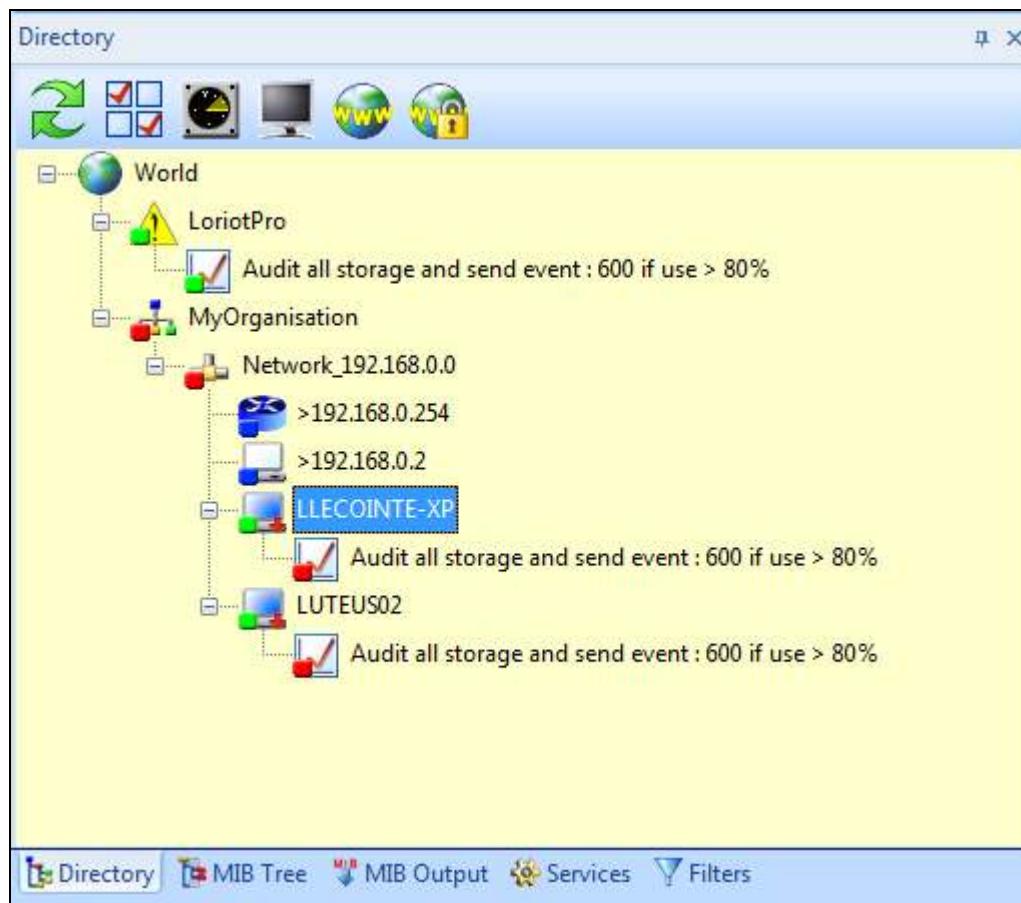




La logique des paramètres IP Expert d'un équipement est la même que pour un réseau. Les paramètres de stratégie « IP Expert » associés à un équipement sont prioritaires sur les paramètres généraux utilisés par le module Discover ou les paramètres de stratégie d'un réseau.

#### 9.3.4.8 cycle de découverte

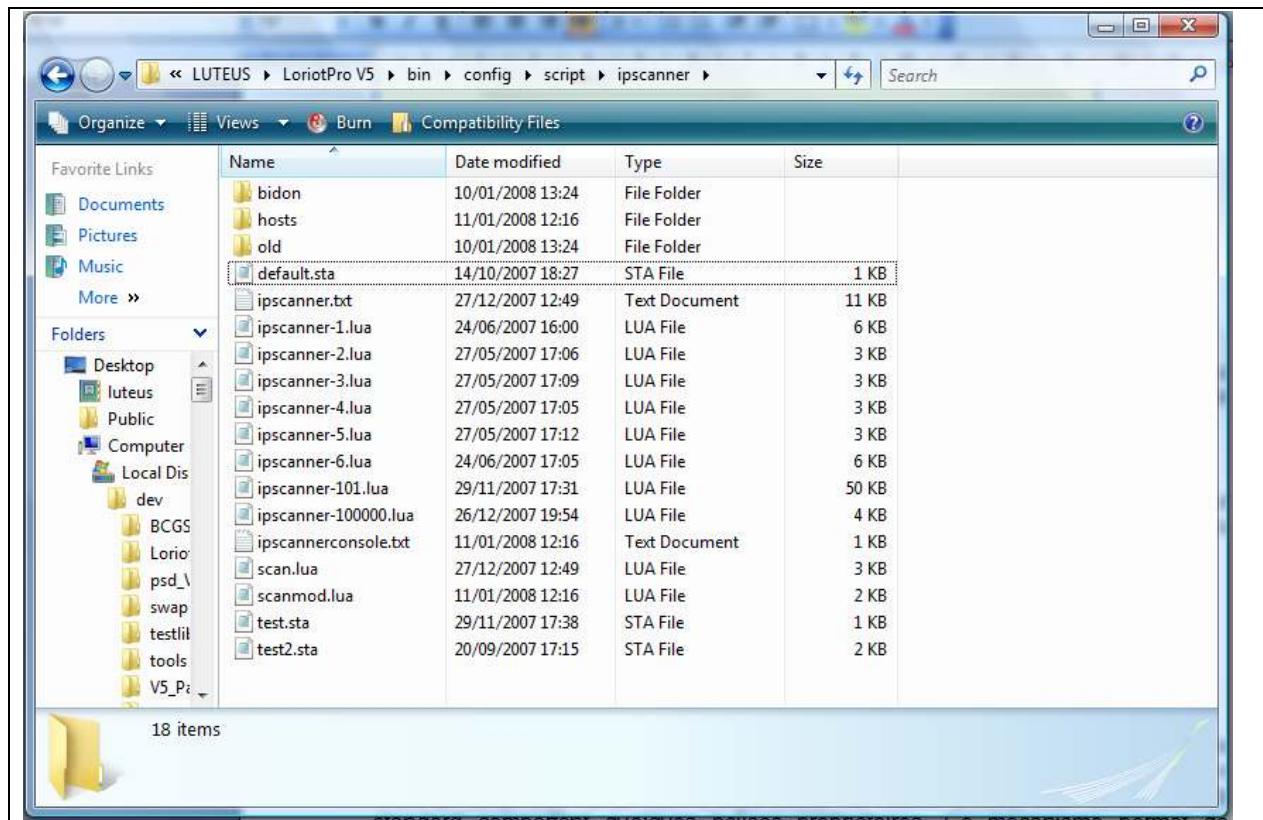
Le premier cycle de découverte a permis d'auto configurer les quelques machines de notre réseau de tests. Le module IP Expert a configuré les « polling » des équipements et a placé automatiquement des scripts d'audits de disque dur sur des équipements supportant la **HOST-RESOURCES-MIB**.



## 9.4 Génération de Scripts pour le module IP Expert ou IP Scanner

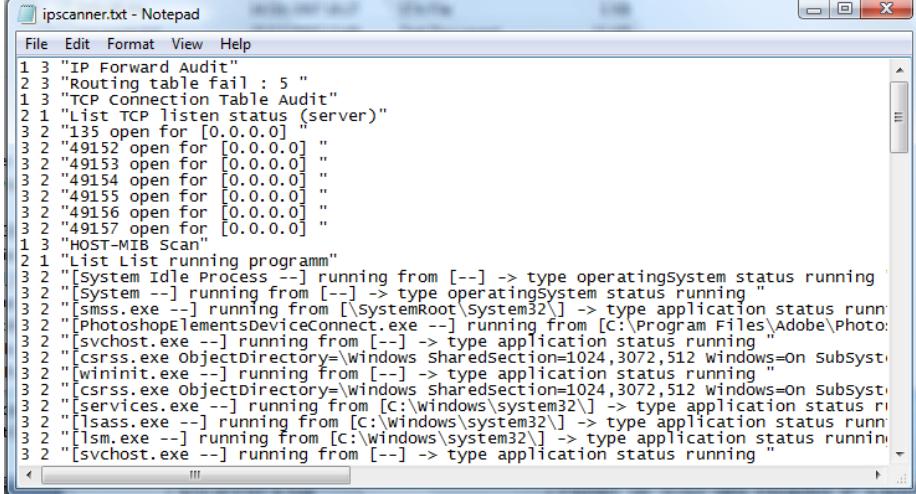
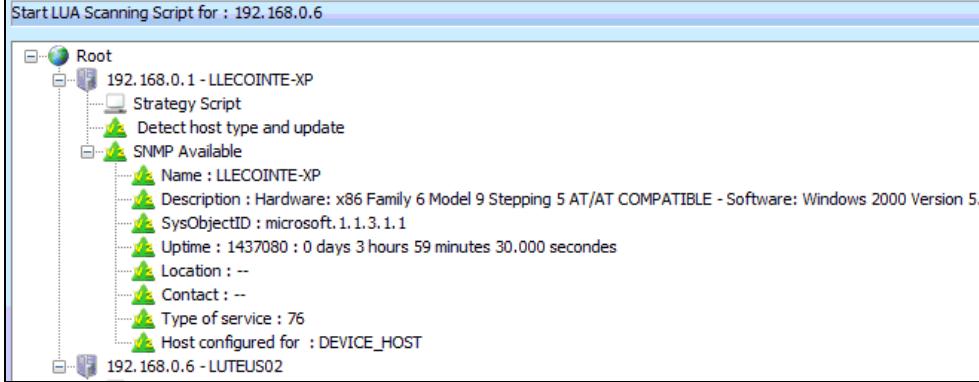
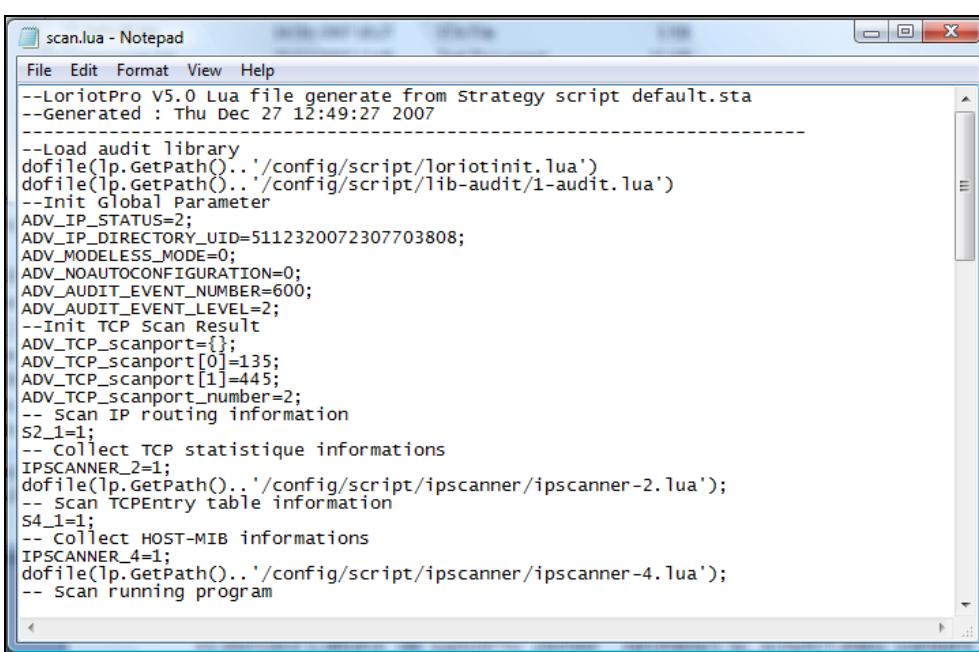
Les scripts utilisés par le module IP Expert sont structurés pour permettre au module de sélection de stratégie d'être opérationnel et faciliter la tâche de l'administrateur.

Tous les scripts des modules IP Expert ou IP Scanner sont situés dans le répertoire : **Bin/config/script/ipscanner** du logiciel. Tous les fichiers ont un nom structuré pour permettre au module de gestion des stratégies une analyse des ressources proposées par le script et un affichage permettant une sélection aisée des options.



Ce répertoire contient différents types de fichiers.

Extention	Usage
ipscanner-x.lua	<p>Ce son les fichiers de script des modules « IP Expert » ou « IP Scanner ».</p> <p>Le X représente le numéro unique de référence assigné au script.</p> <pre> ipscanner-101.lua - Notepad File Edit Format View Help ----- -- this file is located to bin/config/script/ipsscanner -- the name of the file was ipscanner-xx.lua where xx is the scan ref number -- LoriotPro V5 (c) Luteus SARL -- v1.04 ----- title area [[[ title "Discover Host fonction" -- title for this audit file ref 101 -- the reference number for this scan 1 to 100000 was reserved to LUTEUS type 1 -- to define the type of scan comment "This script detect and assign host function" -- somme comment (just one line) param "a list of parameter" -- somme param (just one line)  scan 1 "Discover function" audit 1 "Set default parameters for the host if option autoconfiguration is set" audit 20 "Generate Server list" audit 21 "Insert Storage Disk Audit of server" audit 30 "Generate Bridge list" audit 40 "Generate Router list" audit 50 "Generate WINS Serverlist" audit 51 "Insert Poll WINS Services" audit 60 "Generate DNS Server list" audit 61 "Insert Poll DNS Services" audit 70 "Generate HTTP Server list" audit 71 "Insert Poll HTTP service" audit 80 "Generate FTP Server list" audit 81 "Insert Poll FTP service" audit 90 "Generate LDAP Server list" audit 91 "Insert Poll LDAP service" </pre>
*.sta	Ce sont les fichiers de stratégies, ils son visualisés avec l'éditeur de stratégies.
ipscanner*.txt	Ces sont les fichiers générés par les scripts et qui sont utilisés par

	<p>« IP Expert » ou « IP Scanner » pour remplir l'arbre d'affichage des résultats.</p>  
Scan*.txt	<p>Ce sont les fichiers de scripts générés dynamiquement par les modules « IP Expert » ou « IP Scanner ».</p> 

### 9.4.1 Principe de lancement des scripts IP Expert

Le module Discover lance le module IP Expert en lui fournissant en paramètres soit un réseau à scanner, soit un équipement à expertiser. Dans le cas d'un réseau le module réalise un envoi de requête de type ICMP et/ou SNMP simple vers l'ensemble des adresses IP du réseau mais ne réalise pas de scan TCP ou de scripts d'expertise.

Dans le cas d'un équipement, le Module IP Expert réalise en premier un ping ou/et une requête SNMP vers l'équipement pour vérifier qu'il est disponible, il réalise ensuite un scan de port TCP et finalement génère dynamiquement un fichier temporaire de script qui va contenir les informations collectées par les deux phases précédentes. Ce fichier de scripts générés dynamiquement va contenir la liste des opérations à réaliser lors de l'expertise de l'équipement (la stratégie associée aux scripts d'expertise). Pour réaliser cette liste de tâches, le module IP Expert va ouvrir le fichier de stratégie configuré et conserver les options cochées. Ces options vont permettre la création du fichier de script dynamique.

En résumé, dans le cas d'un équipement (toutes les options son cochées) :

Phases	Action	commentaire
1	Le module Discover lance le module IP Expert avec un équipement de la Directory en variable et attend le retour du module IP Expert.	L'équipement est déjà dans la Directory et ses paramètres de stratégie sont fournis au module IP Expert.
2	Le module IP Expert Ping et/ou SNMP polling, pour vérifier que l'équipement est vivant.	Si l'équipement ne répond, pas le module IP Expert rend la main au module Discover sinon le module IP Expert affiche les résultats dans son arbre (affichage) en créant une entrée pour l'équipement .
4	Le module IP Expert réalise un Scan de port TCP pour commencer la collecte d'information.	Le module IP Expert affiche les résultats dans son arbre (affichage) en créant une entrée pour chaque port TCP valide (réponse positive de l'équipement).
5	Le module IP Expert génère un fichier de scripts dynamiques en utilisant les informations collectées et les options cochées de la stratégie associée à l'équipement.	Le fichier généré est dans le répertoire <i>bin/config/script/ipscanner/</i> et porte le nom : <b>scanmod.lua</b>

```

LoriotPro - Script Editor : C:\Program Files\LUETUS\LoriotPro V5\bin\config\script\ipscanner\scan.lua
Files Edit View Insert BreakPoint Search Options Compiler Value Help
1 --LoriotPro V5.0 Lua file generated from Strategy script default.sta
2 --Generated : Thu Dec 27 12:49:27 2007
3
4 --Load audit library
5 dofile(lp.GetPath()..'/config/script/loriotinit.lua')
6 dofile(lp.GetPath()..'/config/script/lib-audit/i-audit.lua')
7 --Init Global Parameter
8 ADV_IP_STATUS=2;
9 ADV_IP_DIRECTORY_UID=5112320072307703808;
10 ADV_MODELESS_MODE=0;
11 ADV_NOAUTOCONFIGURATION=0;
12 ADV_AUDIT_EVENT_NUMBER=600;
13 ADV_AUDIT_EVENT_LEVEL=2;
14 --Init TCP Scan Result
15 ADV_TCP_scanport={};
16 ADV_TCP_scanport[0]=135;
17 ADV_TCP_scanport[1]=445;
18 ADV_TCP_scanport_number=2;
19 -- Scan IP routing information

```

6	Le module IP Expert exécute le script.	<p>Le fichier <b><i>ipscannerconsole.txt</i></b> est généré par le script dans le répertoire <b><i>bin/config/script/ipscanner/</i></b>; il contient les informations à ajouter à la structure de l'arbre du module IP Expert.</p> <p>En fonction des sélections réalisées dans la stratégie associée à l'équipement, différents fichiers de script <b><i>ipscanner-x.lua</i></b> seront lancés avec passage de paramètres. L'équipement est configuré en fonction des options ou bien on réalise des audits et/ou expertises sur l'équipement. Durant cette phase, la partie inférieure de la boîte de dialogue du module IP Expert affiche les informations générées par les scripts.</p> <p style="color: red;"><b>Le module IP Expert utilise les paramètres système <b><i>lp_host</i></b> et <b><i>lp_oid</i></b> pour fournir au script l'adresse IP de l'équipement et le nom du fichier de résultat (ici <b><i>ipscannerconsole.txt</i></b>)</b></p>
7	Le module IP Expert affiche les résultats fournis par le script dans son arbre (affichage).	Le fichier <b><i>ipscannerconsole.txt</i></b> généré est utilisé par le module IP Expert pour mettre à jour l'affichage de son arbre de résultats.
8	Le module IP Expert envoie une information de fin au module Discover qui lui fournit un nouveau travail.	Le cycle recommence pour un autre équipement.

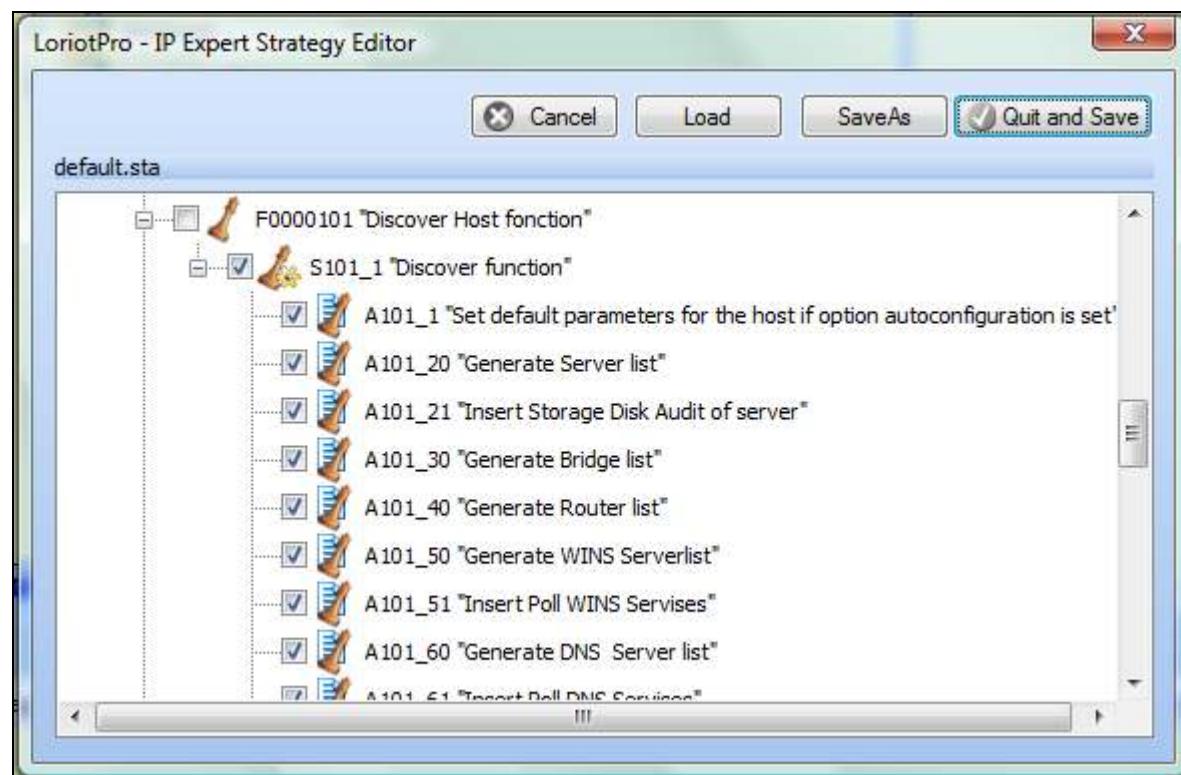
#### 9.4.2 Structure des fichiers ***scanmod.lua*** et ***scan.lua***

Ces fichiers sont générés dynamiquement par les modules IP Expert et/ou IP Scanner après le scan de port TCP en utilisant les informations contenues dans le fichier de stratégie associé à l'équipement en traitement. Le module IP Expert va dans un premier temps initialiser dans le fichier ***scanmod.lua*** un ensemble de variables LUA globales qui pourront être réutilisées par les scripts. Le module IP Scanner génère lui le fichier ***scan.lua***. **Les deux modules peuvent tourner en même temps, les noms de ces fichiers étant différents.**

Dans ce premier exemple, le fichier de stratégie ne contient que quelques sélections et l'équipement n'a pas répondu au scan de port TCP.

<b>Scanmod.lua</b>	<b>Commentaire</b>
--LoriotPro V5.0 Lua file generate from Strategy script default.sta --Generated : Fri Jan 11 12:16:22 2008	C'est l'entête du fichier, il fournit l'heure de création et le fichier de stratégie utilisé.
--Load audit library dofile(ip.GetPath()..'/config/script/loriotinit.lua') dofile(ip.GetPath()..'/config/script/lib-audit/1-audit.lua')	Par défaut, le script charge les fichiers d'initialisation de LUA contenant les fonctions de base et les définitions de variables.
--Init Global Parameter	Nous rendrons dans la zone des variables contenant les informations collecté sur l'équipement.
ADV_IP_STATUS=2;	Le statut actuel de l'équipement dans la Directory
ADV_IP_DIRECTORY_UID=5154148020069074411;	L'UID de l'équipement dans la Directory.
ADV_MODELESS_MODE=1;	Ce script est généré par le module IP Expert
ADV_NOAUTOCONFIGURATION=0;	L'équipement pourra être auto-configuré.
ADV_AUDIT_EVENT_NUMBER=50000;	Les scripts utiliseront l'Event 50000 pour cet équipement. Si =0 il n'y aura pas émission d'Event.
ADV_AUDIT_EVENT_LEVEL=6;	Les Events 50000 émis serons au level 6.
--Init TCP Scan Result ADV_TCP_scanport={}; ADV_TCP_scanport_number=0;	La zone de résultats des ports TCP scannés avec succès. Dans cet exemple aucun.
--Init TCP Scan Result ADV_TCP_scanport={}; ADV_TCP_scanport[0]=135; ADV_TCP_scanport[1]=445; ADV_TCP_scanport_number=2;	La zone de résultats des ports TCP scannés avec succès. Dans cet exemple 2.
lp_host	L'adresse IP de l'équipement
lp_oid	Le nom du fichier de résultat <b>Ici ipscannerconsole.txt</b>

La suite du fichier contient des variables représentants les sélections effectuées par l'administrateur dans la stratégie utilisée (ici default.sta).



-- Discover function S101_1=1;	La fonction 1 « Discover function » du fichier de script IP Expert référence 101 (ipscanner-101.lua) a été sélectionnée.
-- Set default parameters for the host if option autoconfiguration is set A101_1=1;	L'option 1 « Set default parameters for the host if option autoconfiguration is set» du fichier de script IP Expert référence 101 (ipscanner-101.lua) a été sélectionnée.
-- Generate Server list A101_20=1;	L'option 20 « Generate Server list » du fichier de script IP Expert référence 101 (ipscanner-101.lua) a été sélectionnée.
-- Insert Storage Disk Audit of server A101_21=1;	L'option 21 « Insert Storage Disk Audit of server» du fichier de script IP Expert référence 101 (ipscanner-101.lua) a été sélectionnée.
-- Generate Bridge list A101_30=1;	L'option 30 « Generate Bridge list» du fichier de script IP Expert référence 101 (ipscanner-101.lua) a été sélectionnée.
-- Generate Router list	L'option 40 « Generate

A101_40=1;	Router list» du fichier de script IP Expert référence 101 (ipscanner-101.lua) a été sélectionnée.
-- Generate WINS Serverlist A101_50=1;	L'option 50 « Generate WINS Serverlist» du fichier de script IP Expert référence 101 (ipscanner-101.lua) a été sélectionnée.
....	Etc..
....	
dofile(lp.GetPath()..'/config/script/ipscanner/ipscanner-101.lua');	Le fichier de script <b>ipscanner-101.lua</b> est lancé par le script principal <b>scanmod.lua</b> lui-même lancé par le module IP Expert. Le fichier de script ipscanner-101.lua pourra exploiter les paramètres insérés dans le fichier scanmod.lua et réaliser les tâches programmées.

#### 9.4.3 Structure d'un fichier de script IP Expert

Un fichier de script IP Expert devra tenir compte des informations (variables) placées dans le fichier **scanmod.lua** pour être opérationnel.

Chaque fichier de script IP Expert possède une référence unique qui permet au système de générer le fichier dynamique scanmod.txt et les fichiers de stratégie. Les références 1 à 99999 sont réservées à la société Luteus et ne doivent pas être utilisées pour créer de nouveaux scripts. Le wizard de création de scripts IP Expert intégrés à l'environnement de développement gère automatiquement la numérotation des nouveaux scripts.

L'entête d'un fichier **ipscanner-x.txt** est structurée pour permettre au module de stratégie de construire son arbre de choix.

##### 9.4.3.1 Entête d'un fichier ipscanner-x.lua

Prenons en exemple le fichier ipscanner-101.lua déjà utilisé :

Ipscanner-101.lua	commentaires
-- this file is located to bin/config/script/ipscanner -- the name of the file was ipscanner-xx.lua where xx is the scan ref number -- LoriotPro V5 (c) Luteus SARL -- V1.04	Cette zone de commentaire est libre et contient les copyrights du script.

----- title area --[[	Ici commence la zone de définition des fonctions et options disponibles avec ce script.
title "Discover Host fonction" -- title for this audit file	title "Un titre pour votre script entre guillemets »
ref 101 -- the reference number for this scan 1 to 100000 was reserved to LUTEUS	ref xx xx le numéro de référence du script. Le fichier porte aussi le numéro de référence dans son nom.
type 1 -- to define the type of scan	Information réservée
comment "This script detect and assign host function" -- somme comment (just one line)	comment "Un commentaire pour votre script entre guillemets »
param "a list of parameter" -- somme param (just one line)	param "des paramètres de type texte »  réservé pour l'instant
scan 1 "Discover function"	La fonction numéro 1 et son titre
audit 1 "Set default parameters for the host if option autoconfiguration is set"	L'option numéro 1 et son titre
audit 20 "Generate Server list"	L'option numéro 20 et son titre
audit 21 "Insert Storage Disk Audit of server"	L'option numéro 21 et son titre
...	
...	
--tcp list used with this script by the tcp scanner tcp 21,22,23,42,53,80,88,135,139,280,389,443 tcp 445,464,515 tcp 631,631,636,1026,1027,1059,1430,1720,2000,2001 tcp 2002,2103,6050,9100,3389, ,593	Une liste de port TCP à réaliser avant de lancer ce script.  Ces ports TCP seront ajoutés à la liste visible dans la dialog box d'option de l'IP Expert avant le lancement du Scan TCP.
]]--	La fin de l'entête

Soit ipscanner-101.lua :

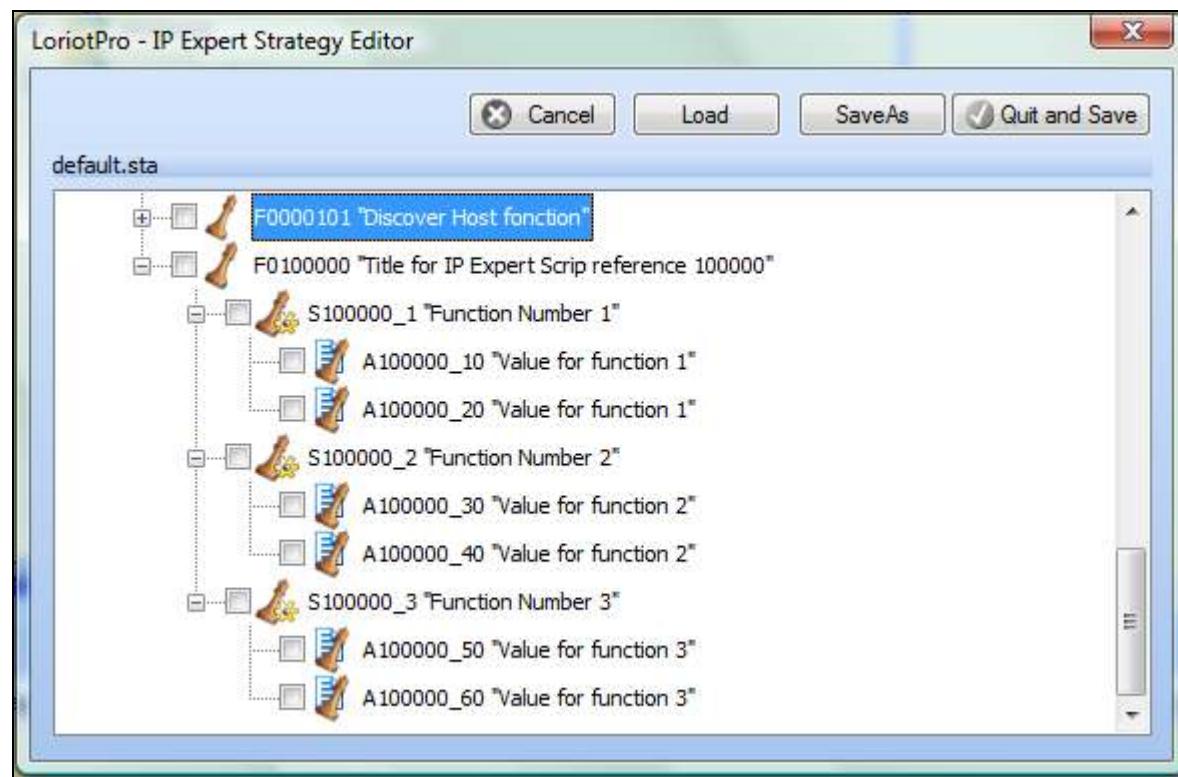
```
-- this file is located to bin/config/script/ipsscanner
-- the name of the file was ipscanner-xx.lua where xx is the scan ref number
-- LoriotPro V5 (c) Luteus SARL
-- V1.04
----- title area
--[][
title "Discover Host fonction" -- title for this audit file
ref 101      -- the reference number for this scan 1 to 100000 was reserved to LUTEUS
type 1       -- to define the type of scan
comment "This script detect and assign host function" -- somme comment (just one line)
param "a list of parameter"   -- somme param (just one line)

scan 1 "Discover function"
audit 1 "Set default parameters for the host if option autoconfiguration is set"
audit 20 "Generate Server list"
audit 21 "Insert Storage Disk Audit of server"
audit 30 "Generate Bridge list"
audit 40 "Generate Router list"
audit 50 "Generate WINS Serverlist"
audit 51 "Insert Poll WINS Services"
audit 60 "Generate DNS Server list"
audit 61 "Insert Poll DNS Services"
audit 70 "Generate HTTP Server list"
audit 71 "Insert Poll HTTP service"
audit 80 "Generate FTP Server list"
audit 81 "Insert Poll FTP service"
audit 90 "Generate LDAP Server list"
audit 91 "Insert Poll LDAP service"
audit 100 "Generate SSH Server list"
audit 101 "Insert Poll SSH Services"
audit 110 "Generate TELNET Server list"
audit 111 "Insert Poll TELNET service"
audit 120 "Generate CMM Server list and set associed phone"
audit 121 "Insert Poll CMM Server service 1720"
audit 130 "Generate printer list "
audit 131 "Insert Poll Printer services 515 631 9100"
audit 140 "Generate microsoft list "
audit 150 "Insert Arcserve (TCP 6050) service and port monitoring"
--tcp list used with this script by the tcp scanner
tcp 21,22,23,42,53,80,88,135,139,280,389,443,445,464,515,593
tcp 631,631,636,1026,1027,1059,1430,1720,2000,2001,2002,2103,6050,9100,3389
]]-
```

Soit l'entête du fichier ipscanner-100000.lua

```
-- Loriotpro V5
-----
-- LoriotPro V5 (c) Luteus SARL
-- IP Expert/Scanner Script File
-- this file is located to bin/config/script/ipsscanner
-- the name of the file was ipscanner-xx.lua where xx is the scan ref number
-- V1.00
----- title area
--[][
title "Title for IP Expert Scrip reference 100000"
ref 100000
type 1
comment "This is the IP Expert Scrip reference 100000"
param "No Parameters"
scan 1 "Function Number 1"
audit 10 "Value for function 1"
audit 20 "Value for function 1"
scan 2 "Function Number 2"
audit 30 "Value for function 2"
audit 40 "Value for function 2"
scan 3 "Function Number 3"
audit 50 "Value for function 3"
```

```
audit 60 "Value for function 3"  
tcp 23,56  
]]--
```

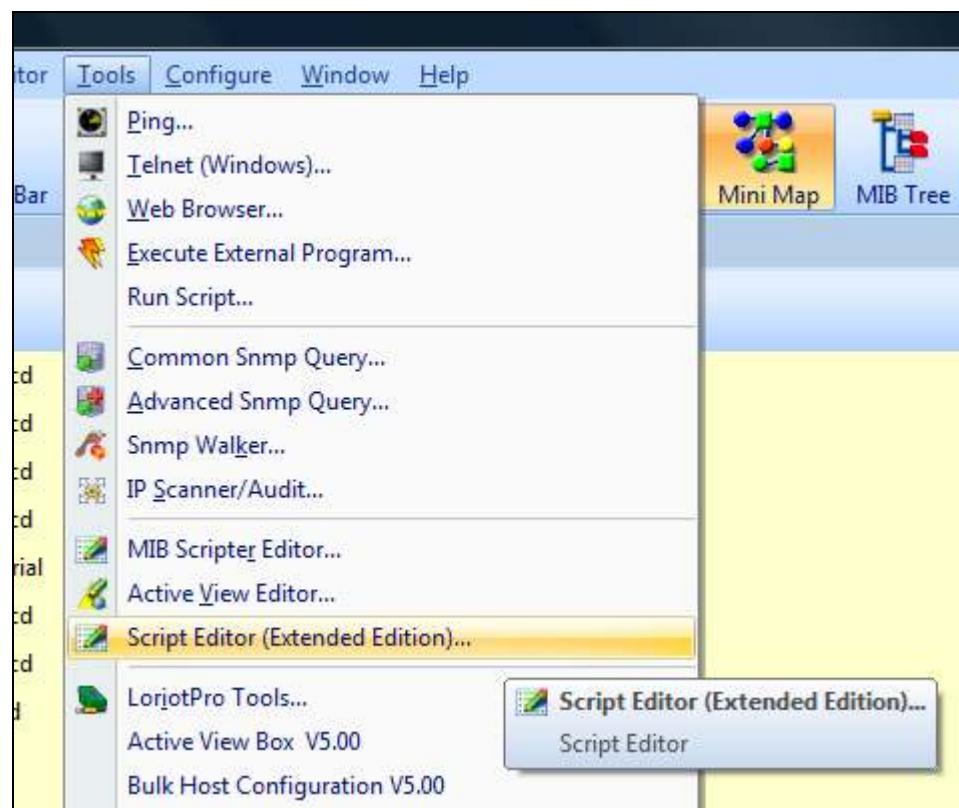


Visualisation de l'entête dans la boîte de configuration des stratégies.

#### 9.4.4 Wizard de création d'un fichier de script IP Expert

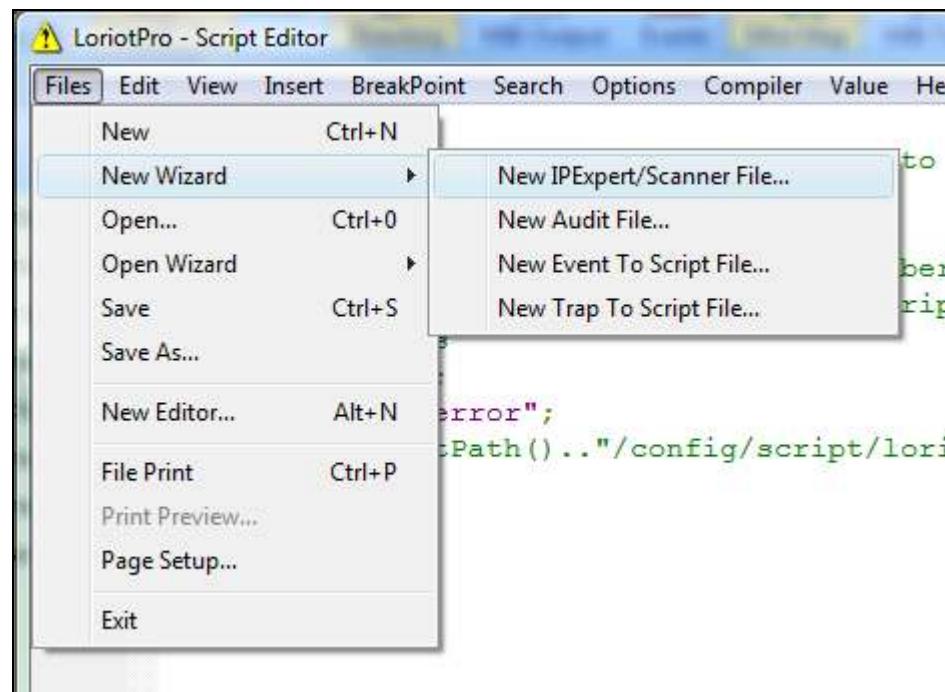
L'environnement de développement vous propose un Wizard de génération de script pour le module IP Expert ou IP Scanner.

Le lancement de l'Editor se fait à partir du menu principal du logiciel :

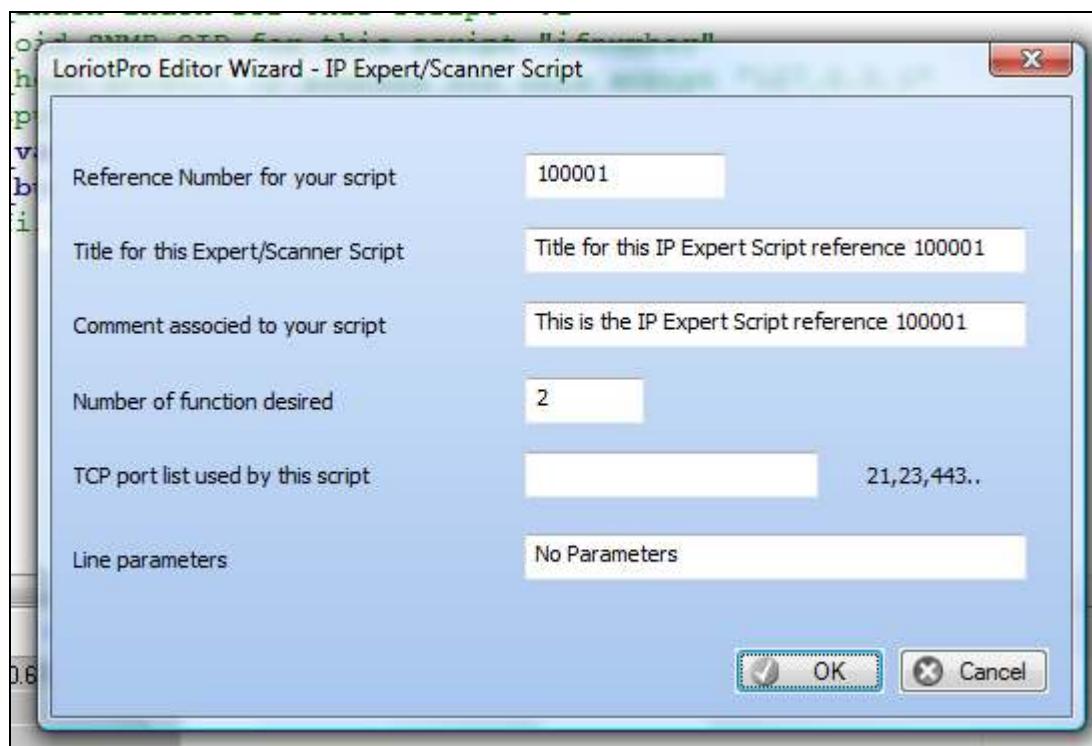


Il faut choisir l'option :

**Files/New Wizard/New IP Expert/Scanner File...**



Une boîte de dialogue vous demande quelques paramètres.



#### 9.4.4.1 Options - Reference Number for your script

Un nombre de référence supérieur à 99999.

**Attention**

Il est préférable de conserver le numéro de référence proposé par le Wizard, il est logiquement libre.

#### 9.4.4.2 Options - Title for this Expert/Scanner Script

Un titre pour votre script.

#### 9.4.4.3 Options - Comment associed to your script

Le commentaire pour votre script

#### 9.4.4.4 Options - Number of function desired

Le wizard ne gère pas les noms des fonctions et les éventuelles options associées, il faudra modifier le fichier généré manuellement.

#### 9.4.4.5 Options - TCP port list used by this script

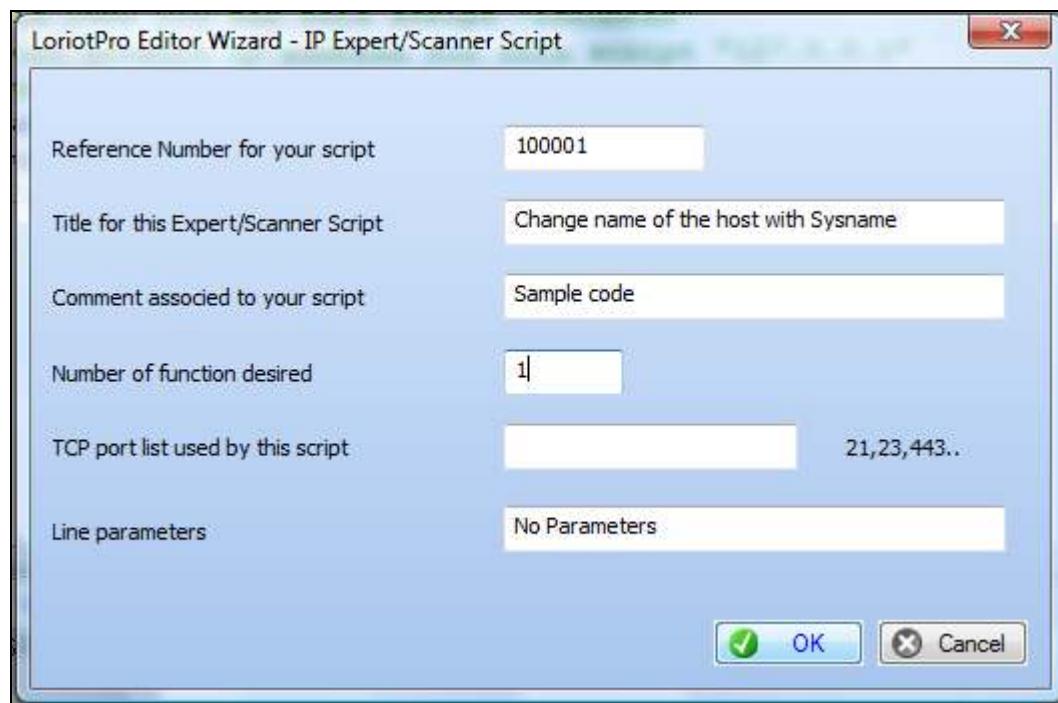
Eventuellement les ports TCP à scanner avant de lancer votre script. Les numéros de ports doivent être séparés par des virgules sans blanc.

#### 9.4.4.6 Options - Line parameters

Cette option est réservée.

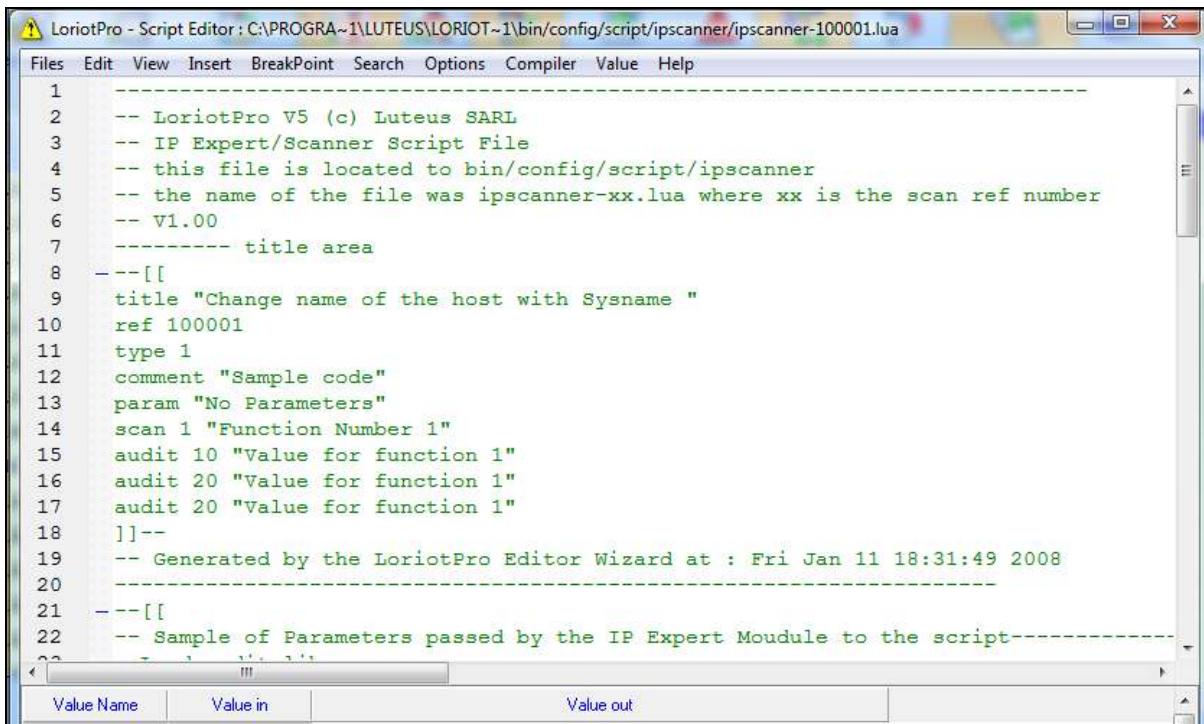
#### 9.4.4.7 Exemple de script

Dans cet exemple, nous allons faire un script qui scanne le paramètre SNMP sysname d'un équipement et met à jour ce paramètre dans la Directory.



Après avoir renseigné les champs de saisie, on valide par « OK ».

## LoriotPro V5.00 Extended Edition – Supervision et gestion SNMP par scripts LUA Introduction au System Expert et aux scripts



The screenshot shows the LoriotPro Script Editor window with the file 'ipscanner-100001.lua' open. The code is a LUA script for an IP Scanner. It includes comments explaining the purpose of the script, its version (V1.00), and the generated date (Fri Jan 11 18:31:49 2008). The script defines a title area and contains audit commands for function 1. A section at the bottom provides a sample of parameters passed by the IP Expert Module to the script.

```
-- LoriotPro V5 (c) Luteus SARL
-- IP Expert/Scanner Script File
-- this file is located to bin/config/script/ipscanner
-- the name of the file was ipscanner-xx.lua where xx is the scan ref number
-- V1.00
----- title area
-- [[
title "Change name of the host with Sysname "
ref 100001
type 1
comment "Sample code"
param "No Parameters"
scan 1 "Function Number 1"
audit 10 "Value for function 1"
audit 20 "Value for function 1"
audit 20 "Value for function 1"
]]--
-- Generated by the LoriotPro Editor Wizard at : Fri Jan 11 18:31:49 2008
-----
-- [[
-- Sample of Parameters passed by the IP Expert Moudule to the script-----
```

Le script est généré avec les paramètres entrés dans le wizard.

Il n'est pas ici question d'expliquer le fonctionnement de la syntaxe du langage LUA mais uniquement la structure d'un fichier de script IP Expert ou IP Scanner et son rapport avec la génération de stratégie.

```
-- LoriotPro V5 (c) Luteus SARL
-- IP Expert/Scanner Script File
-- this file is located to bin/config/script/ipscanner
-- the name of the file was ipscanner-xx.lua where xx is the scan ref number
-- V1.00
----- title area
-- [[
title "Change name of the host with Sysname "
ref 100001
type 1
comment "Sample code"
param "No Parameters"
scan 1 "Function Number 1"
audit 10 "Value for function 1"
audit 20 "Value for function 1"
audit 20 "Value for function 1"
]]--
-- Generated by the LoriotPro Editor Wizard at : Fri Jan 11 18:31:49 2008
-----
-- [[
-- Sample of Parameters passed by the IP Expert Moudule to the script-----
```

--Load audit library  
dofile(lp.GetPath()..'/config/script/loriotinit.lua')  
dofile(lp.GetPath()..'/config/script/lib-audit/1-audit.lua')  
--Init Global Parameter  
ADV\_IP\_STATUS=0;  
ADV\_IP\_DIRECTORY\_UID=5112358460725395478;  
ADV\_MODELESS\_MODE=1;  
ADV\_NOAUTOCONFIGURATION=0;  
ADV\_AUDIT\_EVENT\_NUMBER=600;  
ADV\_AUDIT\_EVENT\_LEVEL=2;  
--Init TCP Scan Result  
ADV\_TCP\_scanport={};  
ADV\_TCP\_scanport[0]=135;

## LoriotPro V5.00 Extended Edition – Supervision et gestion SNMP par scripts LUA

### Introduction au System Expert et aux scripts

```
ADV_TCP_scanport[1]=139;
ADV_TCP_scanport[2]=445;
ADV_TCP_scanport_number=3;
-- lp_host default ip address for this script
-- lp_oid on file name that content value concerning the host
lp_oid='c:/testipexpert.txt';
-- test the function and value
S100001_1=1;
A100001_10 =1;
A100001_20 =1;
]]--  
-----  
-- Start of functions -----  
-----  
function F100001_1()
    lp.Print('in function F100001_1()\n');
-- TODO: Add your message handler code here and/or call default
    if A100001_10 == 1 then
        -- TODO: Add your message handler code here and/or call default
        end
    if A100001_20 == 1 then
        -- TODO: Add your message handler code here and/or call default
        end
end;  
  
-----  
-- Start of your program -----  
-----  
fp=lp.Fopen(lp_oid,'at');      --open the result file (to populate the IP Expert Tree)
if fp then    --the file is open
str=string.format('1 3 %cMy Audit%c\n',34,34); -- create on root for the tree
-- tree format
--  
-- x y text
-- x = tree level
-- y = icon reference on the tree (IP Expert or IP Scanner Tree)
-- text = my text displayed on the Tree
--  
lp.Print(str);
lp.Fprintf(fp,str);
if S100001_1==1 then
    lp.Print('Run function F100001_1\n');
    str=string.format('2 3 %cMy Function information%c\n',34,34);
-- create on branch for the tree
    lp.Fprintf(fp,str);
        F100001_1();
end;
lp.Fclose(fp);
end
```

L'entête est générée et vous pouvez modifier le nom des fonctions et supprimer ou ajouter des variables d'audit.

```
----- title area
--[][]
title "Change name of the host with Sysname "
ref 100001
type 1
comment "Sample code"
param "No Parameters"
scan 1 "Function Number 1"
audit 10 "Value for function 1"
audit 20 "Value for function 1"
audit 20 "Value for function 1"
]]--
```

Une zone de variable vous permettant de simuler le script dans l'environnement de développement est générée.

## LoriotPro V5.00 Extended Edition – Supervision et gestion SNMP par scripts LUA

### Introduction au System Expert et aux scripts

```
-- [[
-- Sample of Parameters passed by the IP Expert Moudule to the script-----
--Load audit library
dofile(lp.GetPath().'/config/script/loriotinit.lua')
dofile(lp.GetPath().'/config/script/lib-audit/1-audit.lua')
--Init Global Parameter
ADV_IP_STATUS=0;
ADV_IP_DIRECTORY_UID=5112358460725395478;
ADV_MODELESS_MODE=1;
ADV_NOAUTOCONFIGURATION=0;
ADV_AUDIT_EVENT_NUMBER=600;
ADV_AUDIT_EVENT_LEVEL=2;
--Init TCP Scan Result
ADV_TCP_scanport={};
ADV_TCP_scanport[0]=135;
ADV_TCP_scanport[1]=139;
ADV_TCP_scanport[2]=445;
ADV_TCP_scanport_number=3;
-- lp_host default ip address for this script
-- lp_oid on file name that content value concerning the host
lp_oid='c:/testipexpert.txt';
-- test the function and value
S100001_1=1;
A100001_10 =1;
A100001_20 =1;
]]--
```

Par défaut cette zone est en commentaire, car logiquement elle n'est pas dans le fichier de script mais dans le fichier **scanmod.lua** généré dynamiquement. Pour tester le script, il faut rentrer cette zone active en supprimant les balises -[[ et ]]-.

Une zone contenant le squelette de votre fonction 1 est générée.

```
-----[[
-- Start of functions
-----[[
function F100001_1()
    lp.Print('in function F100001_1()\n');
    -- TODO: Add your message handler code here and/or call default
    if A100001_10 == 1 then
        -- TODO: Add your message handler code here and/or call default
    end
    if A100001_20 == 1 then
        -- TODO: Add your message handler code here and/or call default
    end
end;
-----[[

```

Par défaut le wizard crée deux variables d'audit par fonction automatiquement. Vous pouvez ajuster ce nombre en fonction de vos besoins.

La zone d'exécution du script est créée.

```
-----[[
-- Start of your program
-----[[
fp=lp.Fopen(lp_oid,'at');      --open the result file (to populate the IP Expert Tree)
if fp then    --the file is open
str=string.format('1 3 %cMy Audit%c\n',34,34); -- create on root for the tree
-- tree format
--
-- x y text
-- x = tree level
-- y = icon reference on the tree (IP Expert or IP Scanner Tree)
-- text = my text displayed on the Tree
--
lp.Print(str);
lp.Fprintf(fp,str);
if S100001_1==1 then
-----[[

```

```

lp.Print('Run function F100001_1\n');
str=string.format('2 3 %cMy Function information%c\n',34,34);
-- create on branch for the tree
lp.Fprintf(fp,str);
F100001_1();

end;

lp.Fclose(fp);
end

```

#### 9.4.4.7.1 zone d'exécution du script

Code	Commentaires
-- Start of your program --	Un commentaire pour délimiter la zone.
fp=lp.Fopen(lp_oid,'at'); --open the result file (to populate the IP Expert Tree) if fp then --the file is open ... ... end;	Le module IP Expert fournit au script le nom de fichier contenant les informations d'affichage pour les résultats ( <b>ipscannerconsole.txt</b> ) dans la variable système <b>lp_oid</b> . Le fichier est ouvert en mode append et le script s'arrête ; le fichier ne peut pas être ouvert.
str=string.format('1 3 %cMy Audit%c\n',34,34); -- create on root for the tree -- tree format -- -- x y text -- x = tree level -- y = icon reference on the tree (IP Expert or IP Scanner Tree) -- text = my text displayed on the Tree -- lp.Print(str); lp.Fprintf(fp,str);	On renseigne le fichier <b>ipscannerconsole.txt</b>
if S100001_1==1 then     lp.Print('Run function F100001_1\n');     str=string.format('2 3 %cMy Function information%c\n',34,34); -- create on branch for the tree     lp.Fprintf(fp,str);     F100001_1(); end;	Si la fonction 1 du fichier ipscanner 100001 a été cochée dans la stratégie associée à l'équipement, la variable s10001_1 est égale à 1, alors la fonction F10001_1() sera lancée.

#### 9.4.4.7.2 Zone des fonctions du script

code	commentaires
<pre>-- Start of functions</pre> <hr/> <pre>function F100001_1()     lp.Print('in function F100001_1()\n'); -- TODO: Add your message handler code here and/or call default     if A100001_10 == 1 then         -- TODO: Add your message handler code here and/or call default             end         if A100001_20 == 1 then             -- TODO: Add your message handler code here and/or call default                 end end;</pre>	<p>Un commentaire pour délimiter la zone.</p> <p>C'est la fonction F100001_1(). Si l'option 10 du script 100001 est cochée dans la stratégie, alors l'option est réalisée (variable A100001_10 = 1). Si l'option 20 du script 100001 est cochée dans la stratégie alors l'option est réalisée (variable A100001_20 = 1).</p>

Modification du code pour réaliser notre script :

Les modifications sont en rouge, les variables inutiles ont été supprimées.

```
-----  

-- LoriotPro V5 (c) Luteus SARL
-- IP Expert/Scanner Script File
-- this file is located to bin/config/script/ipsscanner
-- the name of the file was ipsscanner-xx.lua where xx is the scan ref number
-- V1.00
----- title area
--[][]
title "Change name of the host with Sysname "
ref 100001
type 1
comment "Sample code"
param "No Parameters"
scan 1 "scan the OID sysname"
audit 10 "Set the name of the host with snmp sysname information"
]]--
-- Generated by the LoriotPro Editor Wizard at : Fri Jan 11 18:31:49 2008
-----  

-- lp_host default ip address for this script
-- lp_oid on file name that content value concerning the host
lp_oid='c:/testipexpert.txt';
-- test the function and value
S100001_1=1;
A100001_10=1;  

----- Start of functions -----
-----  

function F100001_1()
    lp.Print('in function F100001_1()\n');

-- TODO: Add your message handler code here and/or call default
i=lp.Gets(lp_host,"sysname","a");
if i==1 then
str=string.format("1 3 %cSystem group%c\n",34,34);
lp.Print(str);
lp.Fprintf(fp,str);
str=string.format("2 3 %cName : %s %c\n",34,a["sysname.0"],34);
lp.Print(str);
lp.Fprintf(fp,str);
```

## LoriotPro V5.00 Extended Edition – Supervision et gestion SNMP par scripts LUA

### Introduction au System Expert et aux scripts

```
if A100001_10 == 1 then
    -- TODO: Add your message handler code here and/or call default
        lp.SetIPOption(lp_host,LP_LUA_NAME,a['sysname.0']);
        lp.Print("Host ",lp_host," name changed to : ",a['sysname.0'], "\n");
    end
end;
end;

-----
-- Start of your program -----
-----

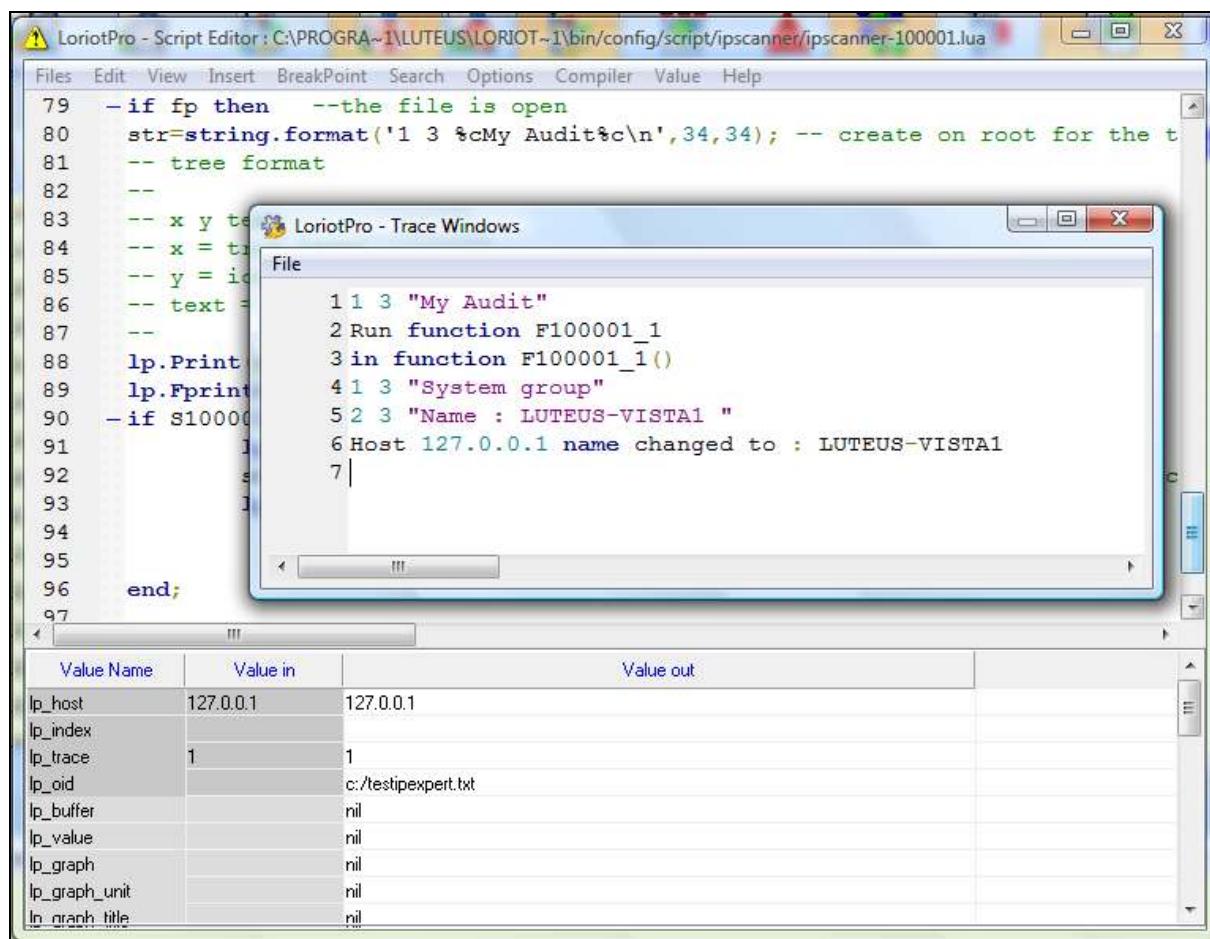
fp=lp.Fopen(lp_oid,'at');      --open the result file (to populate the IP Expert Tree)
if fp then  --the file is open
str=string.format('1 3 %cMy Audit%c\n',34,34); -- create on root for the tree
-- tree format
--
-- x y text
-- x = tree level
-- y = icon reference on the tree (IP Expert or IP Scanner Tree)
-- text = my text displayed on the Tree
--
lp.Print(str);
lp.Fprintf(fp,str);
if S100001_1==1 then
    lp.Print('Run function F100001_1\n');
    str=string.format('2 3 %cMy Function information%c\n',34,34); -- create on branch for
the tree
    lp.Fprintf(fp,str);
    F100001_1();
end;

lp.Fclose(fp);
end
```

Les variables suivantes ont été ajoutées pour simuler l'exécution du script.

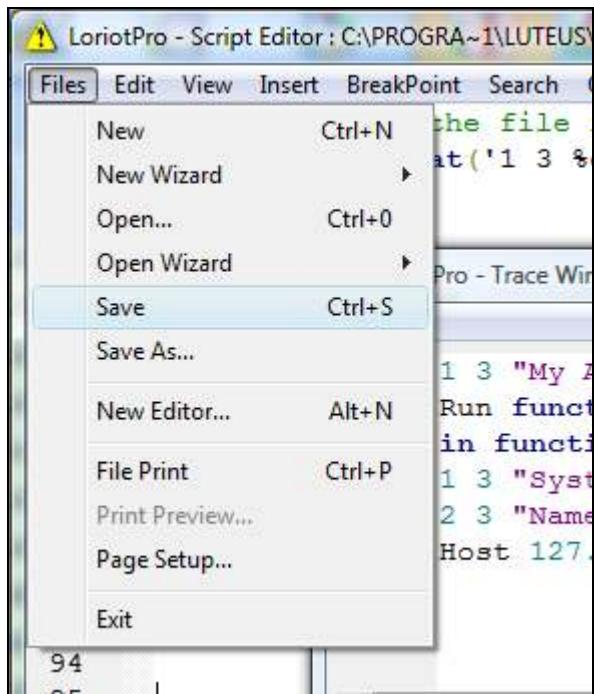
```
-- lp_host default ip address for this script
-- lp_oid on file name that content value concerning the host
lp_oid='c:/testipexpert.txt';
-- test the function and value
S100001_1=1;
A100001_10=1;
```

On lance le script avec la touche F5



#### 9.4.4.8 Sauvegarde du script IP Expert

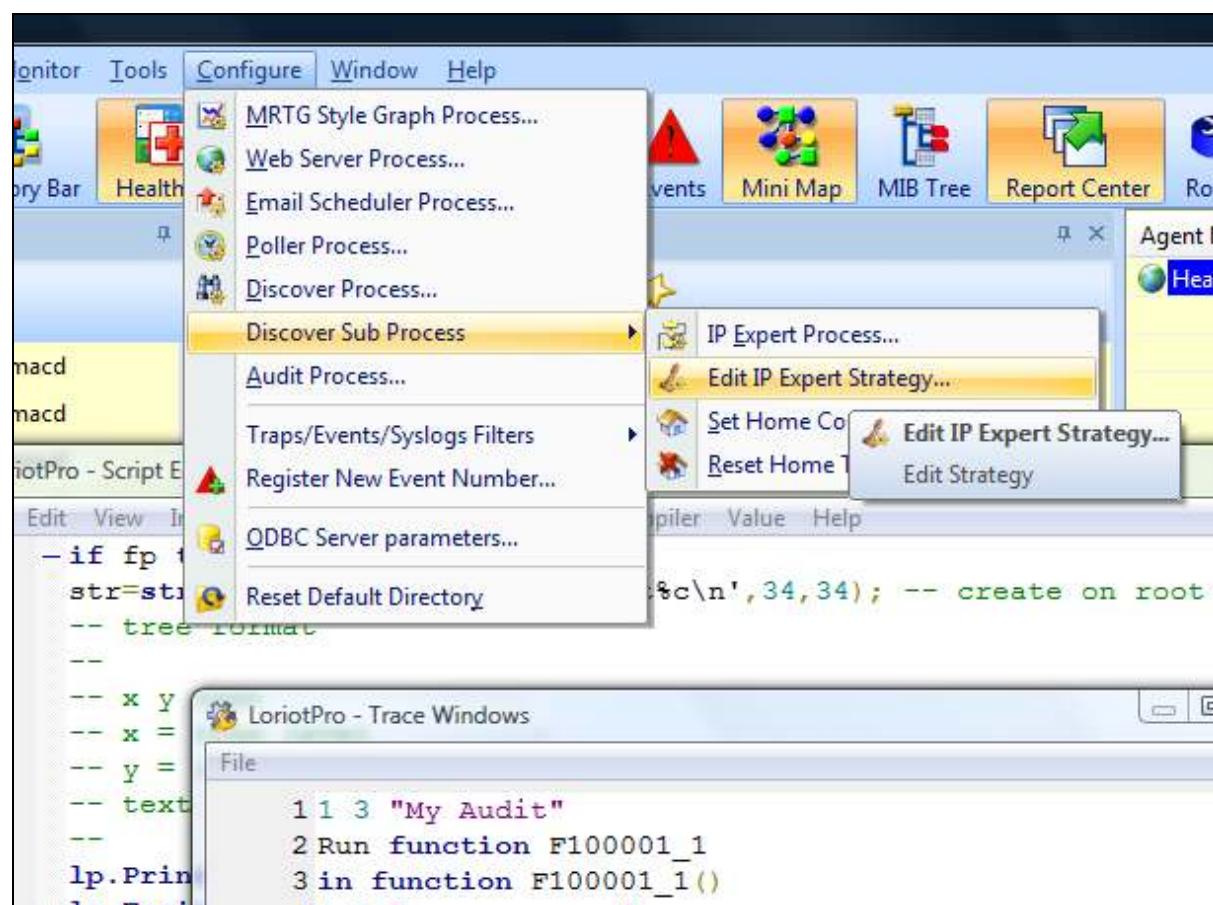
Le fichier de scripts doit être sauvegardé dans le répertoire **bin/config/script/ipsscanner** pour être analysé par le module de gestion des stratégies.



Utiliser Files>Save

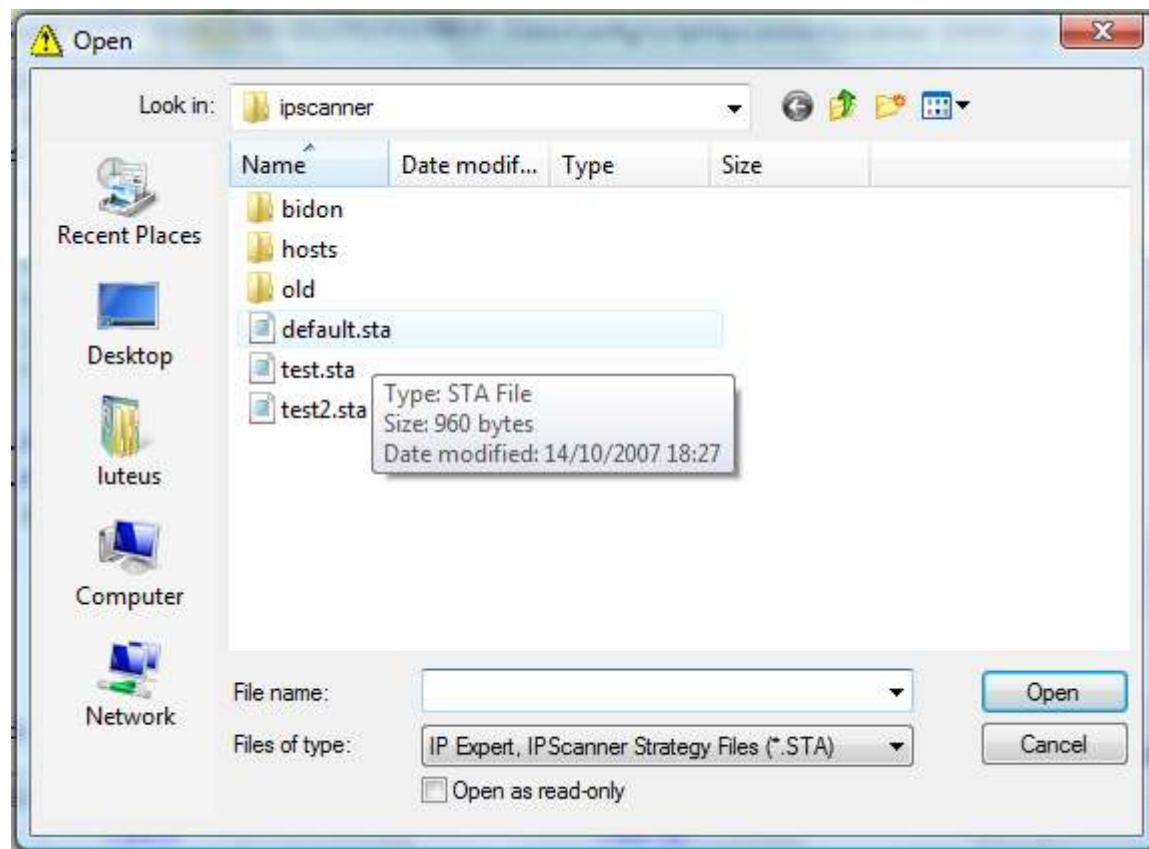
#### 9.4.4.9 Visualisation du nouveau script dans les stratégies

Le nouveau script est maintenant disponible dans le module de stratégie.

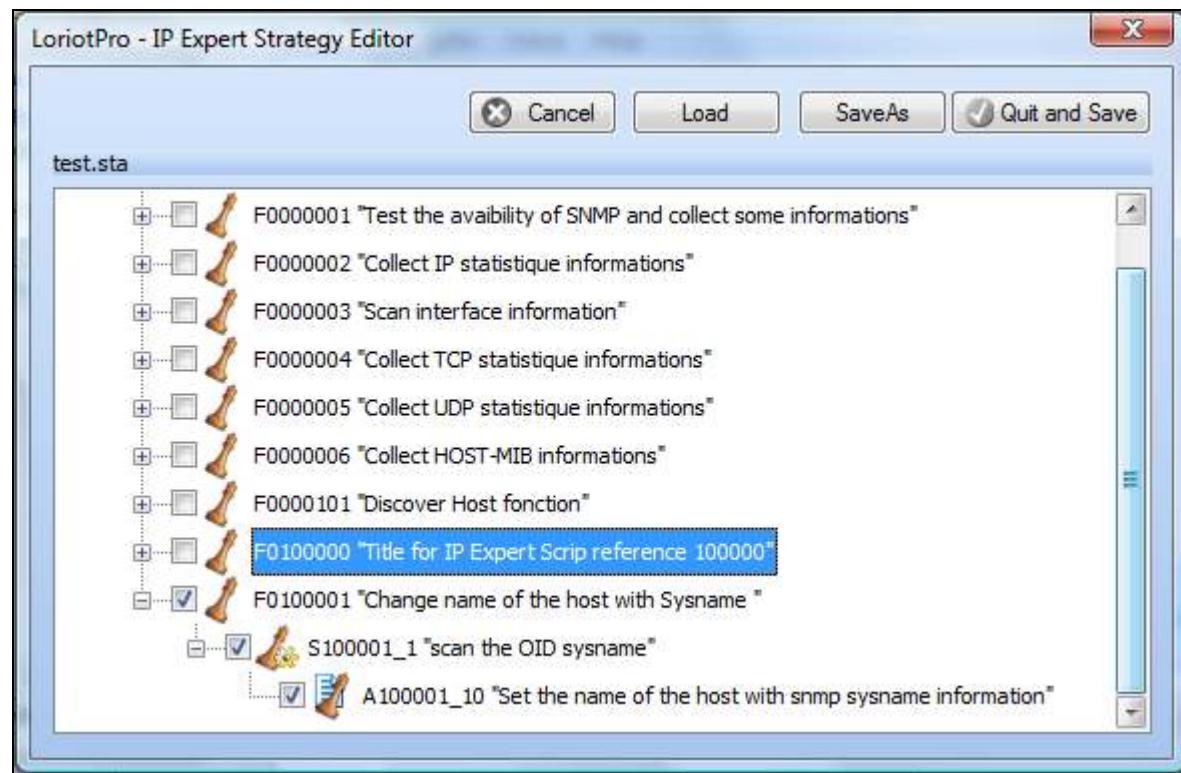


Charger la stratégie dans laquelle vous voulez activer ce nouveau script (test.sta par exemple).

Vous pouvez charger la stratégie default.sta et la sauvegarder ensuite sous un autre nom pour créer une nouvelle stratégie.

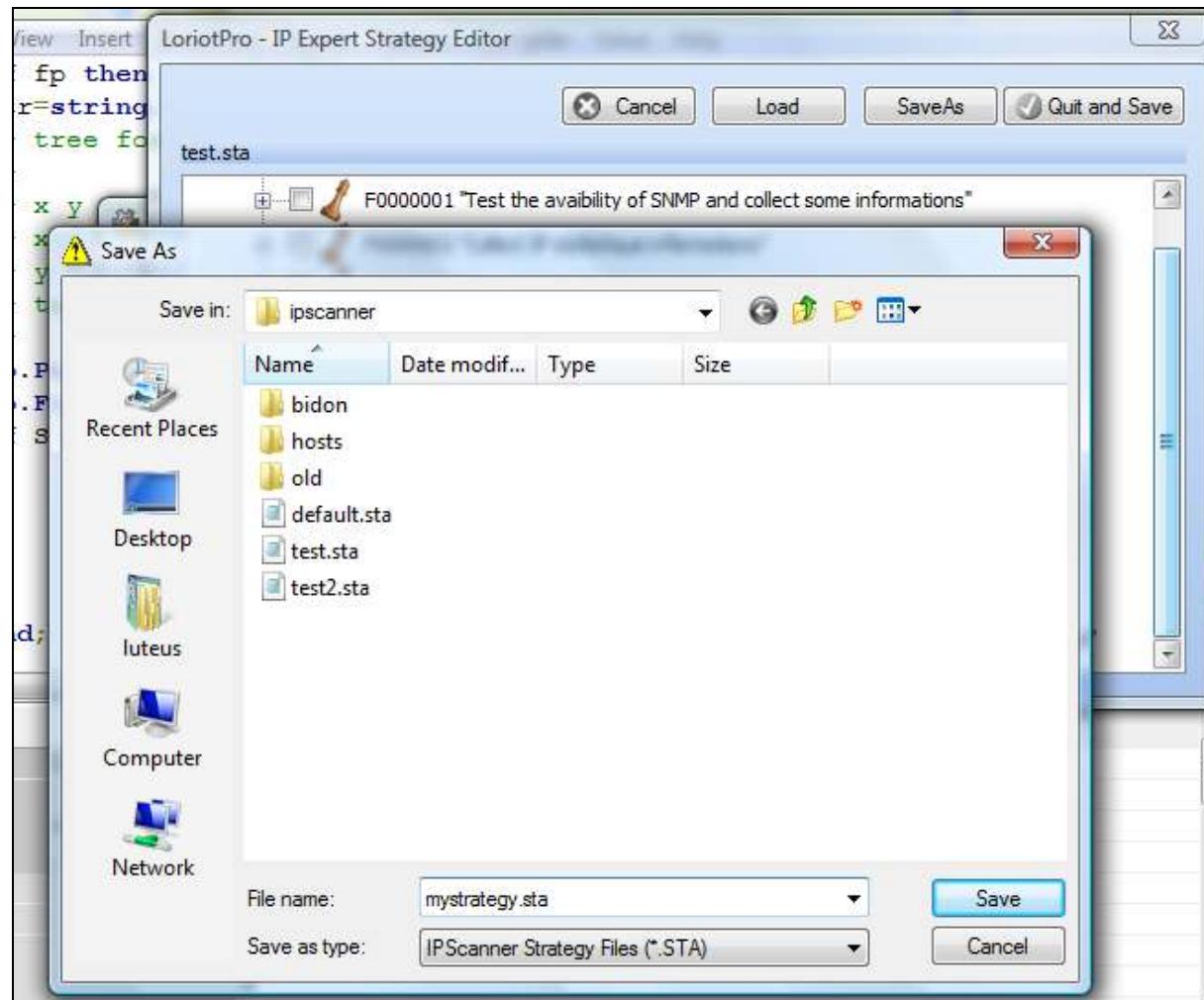


Votre nouveau script IP Expert de référence 100001 est visible en bas de l'arbre et vous retrouvez les paramètres de votre fonction 1. L'adresse IP de l'équipement étant une variable, ce script IP Expert peut être appliqué à tous les équipements de votre Directory.

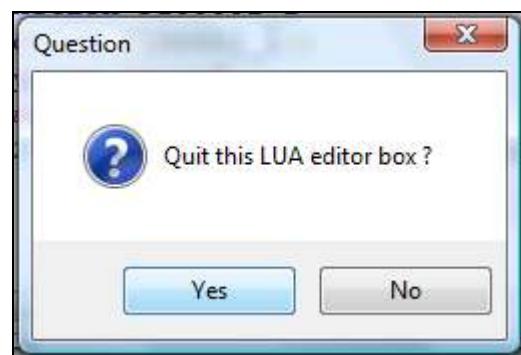


Pour notre test décochez l'ensemble des options et cochez uniquement les options de votre script.

Sauvegardez le fichier de stratégie sous le nom **mystrategy.sta**.



Quittez l'éditeur .



#### 9.4.4.9.1 Test du script avec IP Scanner

## LoriotPro V5.00 Extended Edition – Supervision et gestion SNMP par scripts LUA

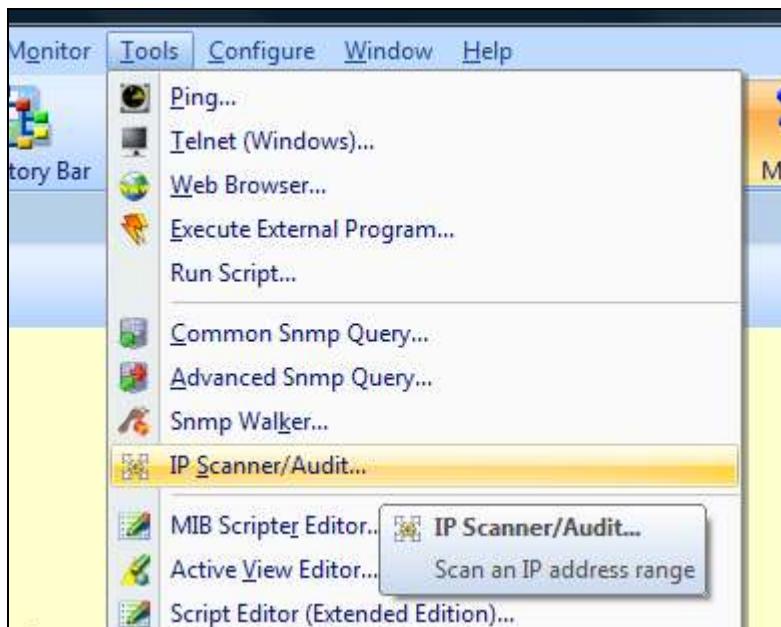
### Introduction au System Expert et aux scripts

Nous allons tester notre script et notre stratégie associée avec le module IP Scanner qui utilise les mêmes fonctionnalités que le module IP Expert.

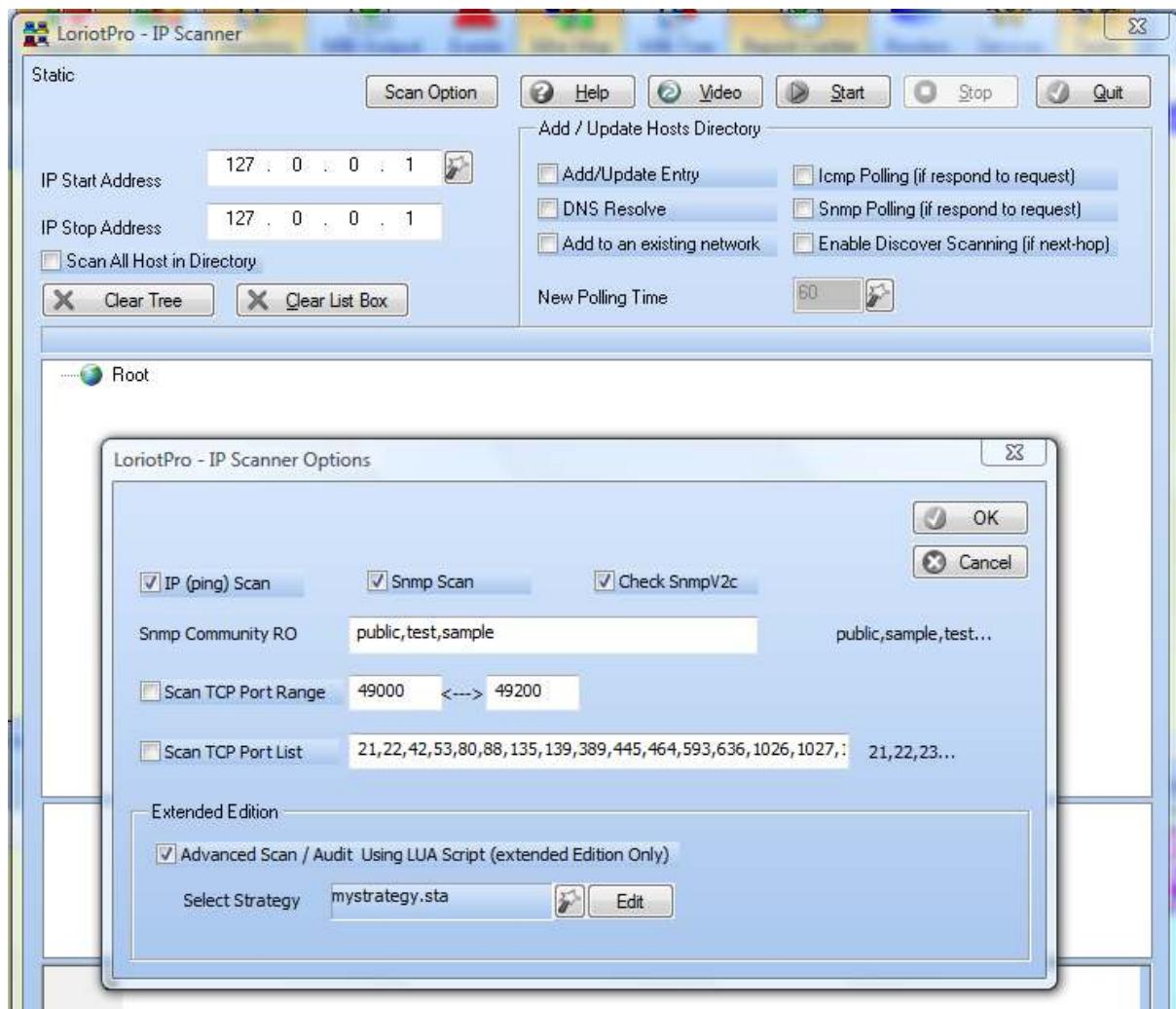
Selectionnez un équipement (clic de souris dans la Directory) qui supporte SNMP. (on ne peut pas changer le nom de l'objet LoriotPro)



Sélectionnez dans tools IP Scanner.



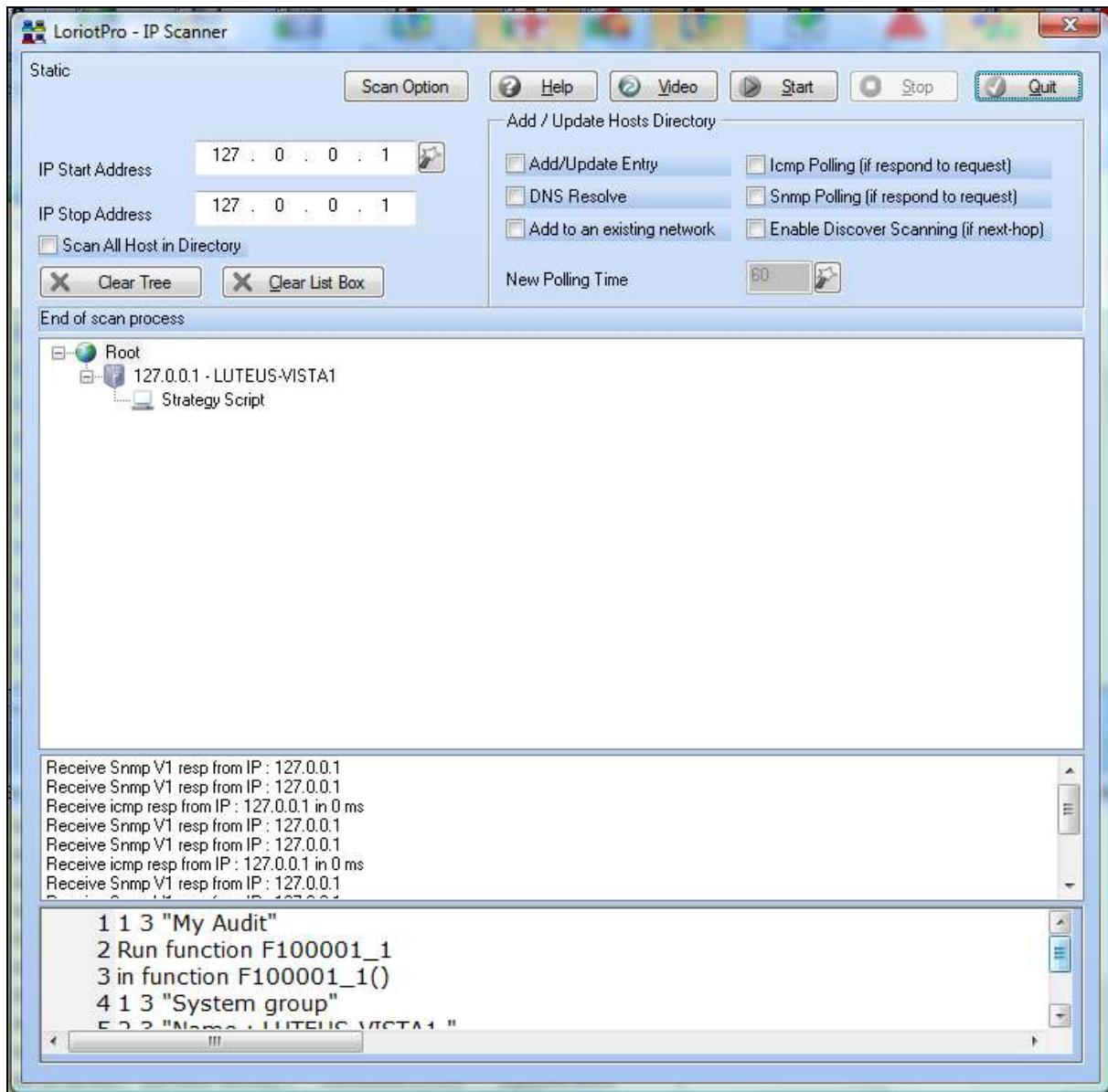
Le module IP Scanner est plus riche que IP Expert mais pour l'instant ouvrez la fenêtre d'option et sélectionnez la stratégie .



Faites « OK »

Le scan est lancé et le script est exécuté.

## LoriotPro V5.00 Extended Edition – Supervision et gestion SNMP par scripts LUA Introduction au System Expert et aux scripts



Le fichier **scan.lua** a été généré dynamiquement par IP Scanner pour lancer le fichier de script **ipscanner-100001.lua** avec les bons paramètres. Dans notre cas, il est très simple. Un seul script est lancé avec une seule fonction.

```
--LoriotPro V5.0 Lua file generate from Strategy script mystrategy.sta
--Generated : Fri Jan 11 19:35:52 2008
-----
--Load audit library
dofile(lp.GetPath()..'/config/script/loriotinit.lua')
dofile(lp.GetPath()..'/config/script/lib-audit/1-audit.lua')
--Init Global Parameter
ADV_IP_STATUS=2;
ADV_IP_DIRECTORY_UID=4941630697505817928;
ADV_MODELESS_MODE=0;
ADV_NOAUTOCONFIGURATION=0;
ADV_AUDIT_EVENT_NUMBER=600;
ADV_AUDIT_EVENT_LEVEL=2;
--Init TCP Scan Result
ADV_TCP_scanport={};
ADV_TCP_scanport_number=0;
-- scan the OID sysname
S100001_1=1;
-- Set the name of the host with snmp sysname information
```

## LoriotPro V5.00 Extended Edition – Supervision et gestion SNMP par scripts LUA

### Introduction au System Expert et aux scripts

```
A100001_10=1;
-- Set the name of the host with snmp sysname information
dofile(lp.GetPath() .. '/config/script/ipscanner/ipscanner-100001.lua');
```

## 10 Le module d’Audit

### 10.1 Introduction

Le module IP Expert peut être utilisé pour lancer à intervalles réguliers des analyses scriptées sur les équipements de la Directory. Cependant, malgré la puissance de ce système, il s'avère peu pratique car il ne permet pas la maîtrise de l'intervalle de réalisation de l'analyse. De plus, un « stop » du module Discover arrête l'ensemble des fonctionnalités du système IP Expert. Le module IP Expert est dédié à la découverte et à la configuration ou reconfiguration des équipements. Pour faire face à ces petits problèmes d'intervalles de polling et pour rendre le système d'analyse des équipements plus indépendant, LoriotPro V5 « Extended Edition » intègre un nouveau module géré par le noyau du logiciel et qui permet le positionnement de scripts d'audits sous les équipements de la Directory. Les scripts d'audits sont structurés et leur usage est banalisé par l'utilisation d'interfaces graphiques. Les scripts d'audits possèdent une structure spécifique qui permet leur réutilisation sur différents équipements de la Directory et éventuellement la constitution de librairies de ressources échangeables ou diffusables. Les scripts d'audits fournis avec l'« Extended Edition » sont OpenSource et peuvent être utilisés pour créer de nouveaux scripts. Les scripts d'audits peuvent être positionnés automatiquement dans la Directory par d'autres scripts. Le module IP Expert, lors de sa phase de configuration automatique des équipements, a la capacité, en fonction de votre sélection dans l'interface graphique de stratégie, d'insérer un script d'audit spécifique sous un équipement.

#### 10.1.1 Quelques exemples

Le module IP Expert scanne un équipement et découvre qu'il supporte la 'HOST-RESOURCE-MIB' et que cet équipement est un serveur de fichiers Microsoft. Il est alors possible pour le module IP Expert d'insérer un audit de contrôle de la capacité des disques du serveur avec un seuil de déclenchement d'alerte si la capacité d'utilisation dépasse 75%. IP Expert configure automatiquement, pour cet audit, un intervalle de réalisation toutes les 15 minutes et d'autres options.

Le module IP Expert scanne un équipement et découvre qu'il est un serveur d'impression. Il est alors possible pour le module IP Expert d'insérer un audit de contrôle du port TCP 515. IP Expert configure automatiquement, pour cet audit, un intervalle de réalisation toutes les 15 minutes et d'autres options.

Le module IP Expert scanne un équipement et découvre qu'il supporte la 'HOST-RESOURCES-MIB' et que cet équipement est un serveur Exchange Microsoft. Il est alors possible pour le module IP Expert d'insérer un audit de contrôle de la présence des services « Exchange » et de déclencher une alerte s'ils ne sont plus présents dans la table des process du serveur. IP Expert configure automatiquement, pour cet audit, un intervalle de réalisation toutes les 15 minutes et d'autres options.

Vous utilisez la fonction d'insertion de tâches sur un équipement et vous configurez manuellement un module d'audit que vous avez écrit en utilisant les Wizards de l'environnement de développement.

## 10.2 Console du module d'audit

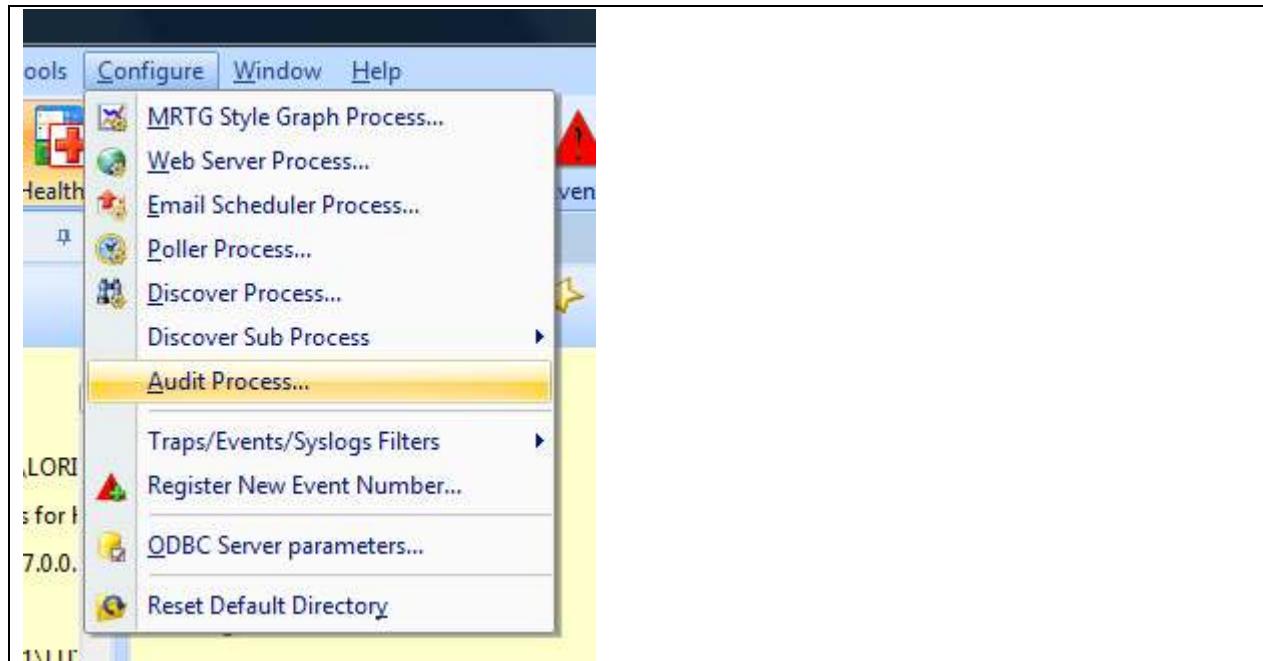
Le module de lancement des scripts d'audit est intégré au noyau du logiciel et dispose par défaut de 5 threads multi-tâches pour assurer au mieux les contraintes d'intervalles de lancement des scripts. Il est indispensable qu'au moins un thread soit disponible lorsque l'échéance de lancement d'un script intervient pour respecter les intervalles d'audits.

### Attention

Logiquement 5 Threads sont suffisants pour permettre une réalisation fluide des scripts d'audits ; cependant si la durée d'exécution de vos scripts est trop longue et vos intervalles de polling très courts, il est possible que le noyau ne puisse pas honorer les intervalles. Dans ce cas, il travaille en mode best effort. Si vous désirez disposer de plus de 5 threads prenez contact avec notre service support [support@loriotpro.com](mailto:support@loriotpro.com)

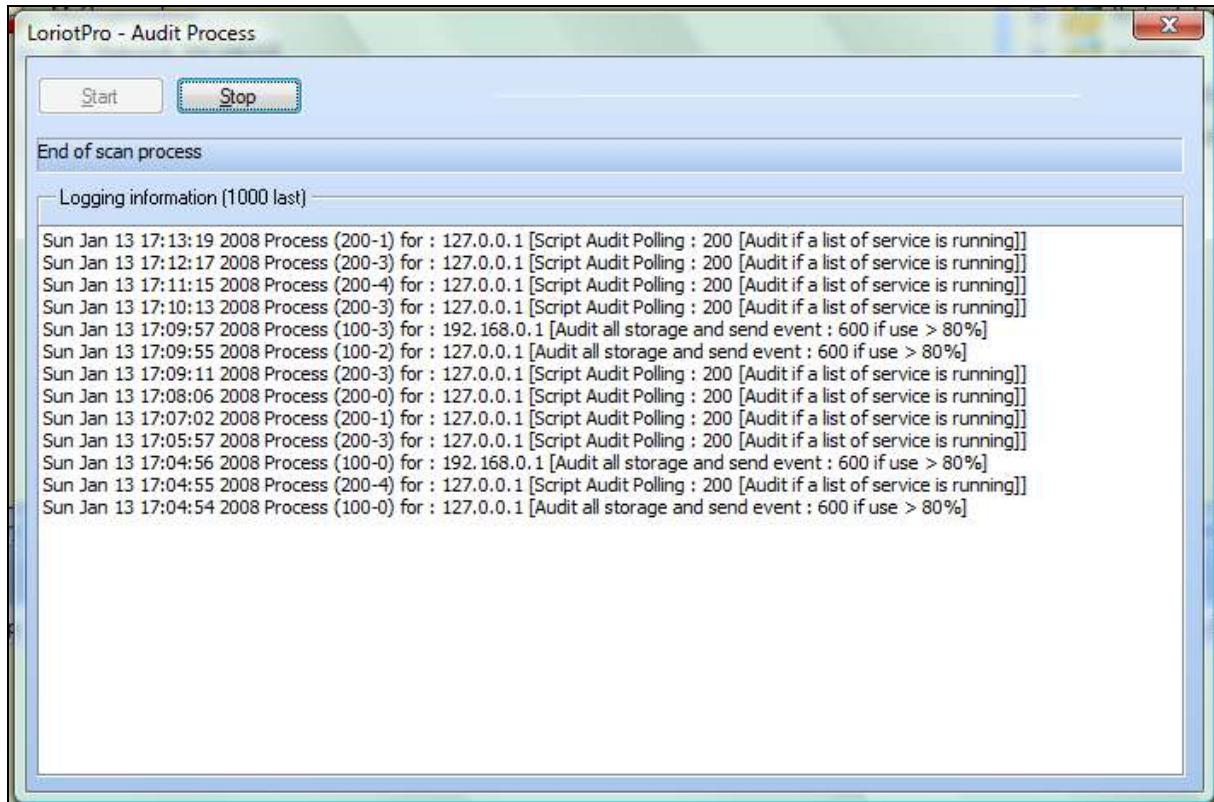
La console du module d'audit est visible à partir du menu :

**Configure>Audit process...**



La console du module d'audit fournit des informations sur le résultat des 1000 derniers audits réalisés. Il est possible de stopper le polling de l'ensemble des scripts avec le

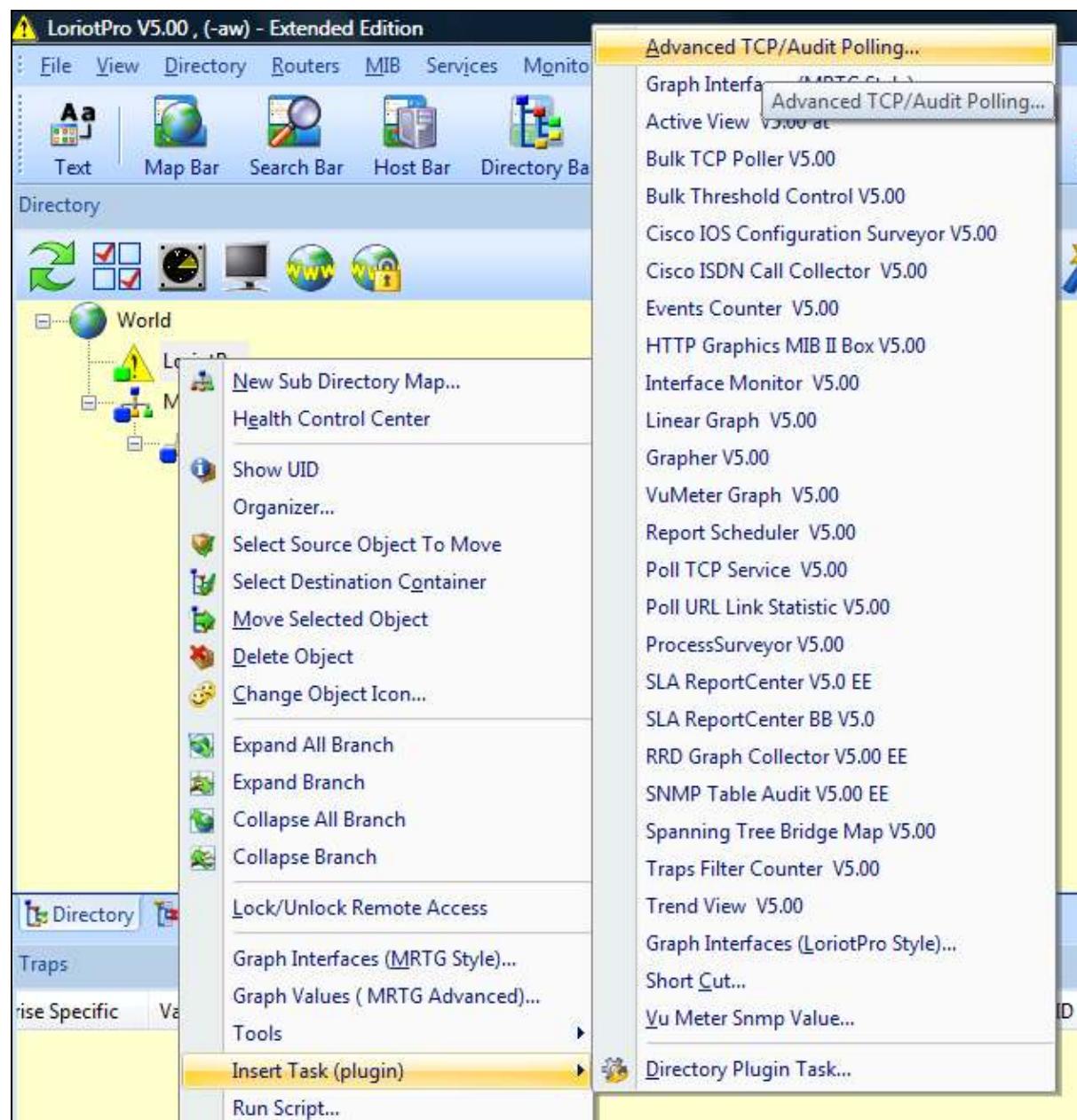
bouton « Stop » ; le bouton « Start » relance le polling. Le module d'audit possède 5 threads, ce qui est logiquement suffisant pour assurer une cohérence des intervalles de polling des audits. En cas d'utilisation de fonction Stop / Start, si une longue période s'installe entre le Stop et le Start, il risque d'y avoir un effet de bouchon au redémarrage, alors les intervalles de polling sont en mode « best effort ».



### 10.3 Insertion manuelle d'un script d'audit

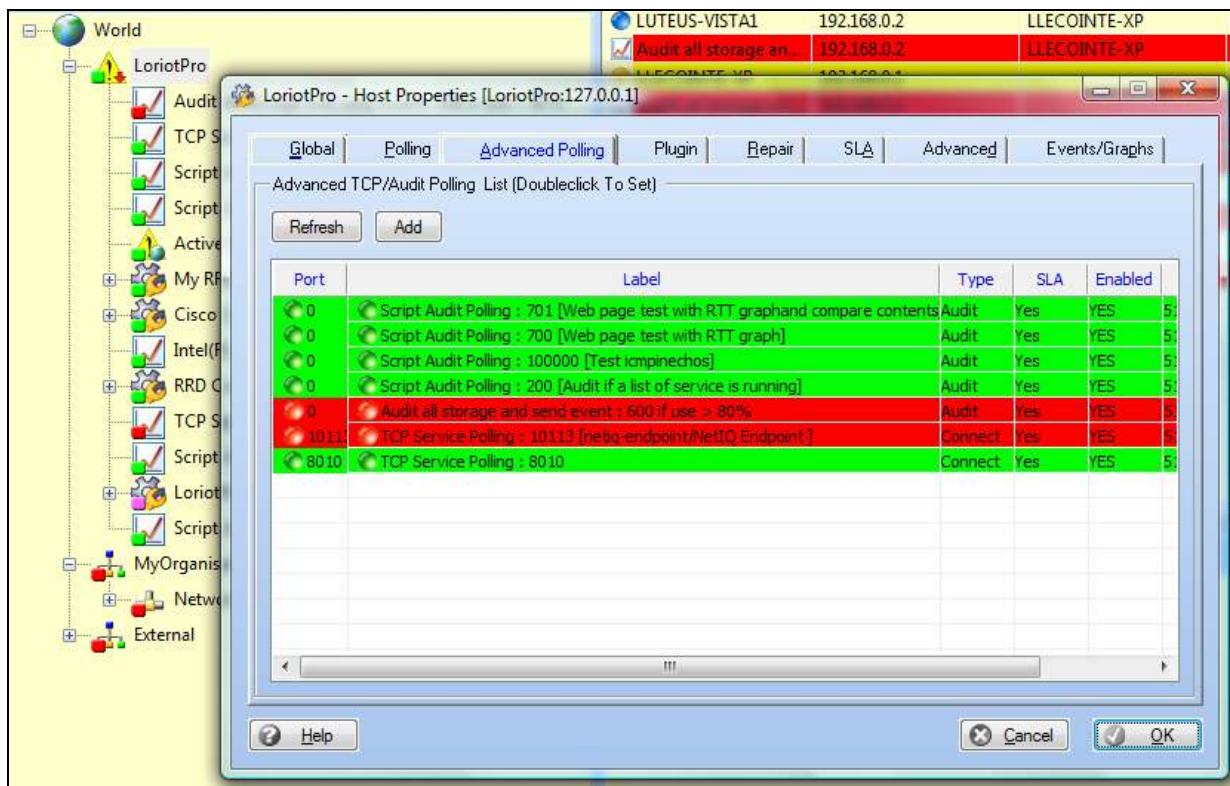
Pour insérer manuellement un script d'audit sur un équipement, il faut utiliser, à partir du menu contextuel d'un équipement, le menu

***Insert Task (plugin)>Advanced TCP/Audit Polling...***



ou bien lancer la boîte de dialogue des paramètres d'un équipement et choisir l'onglet :

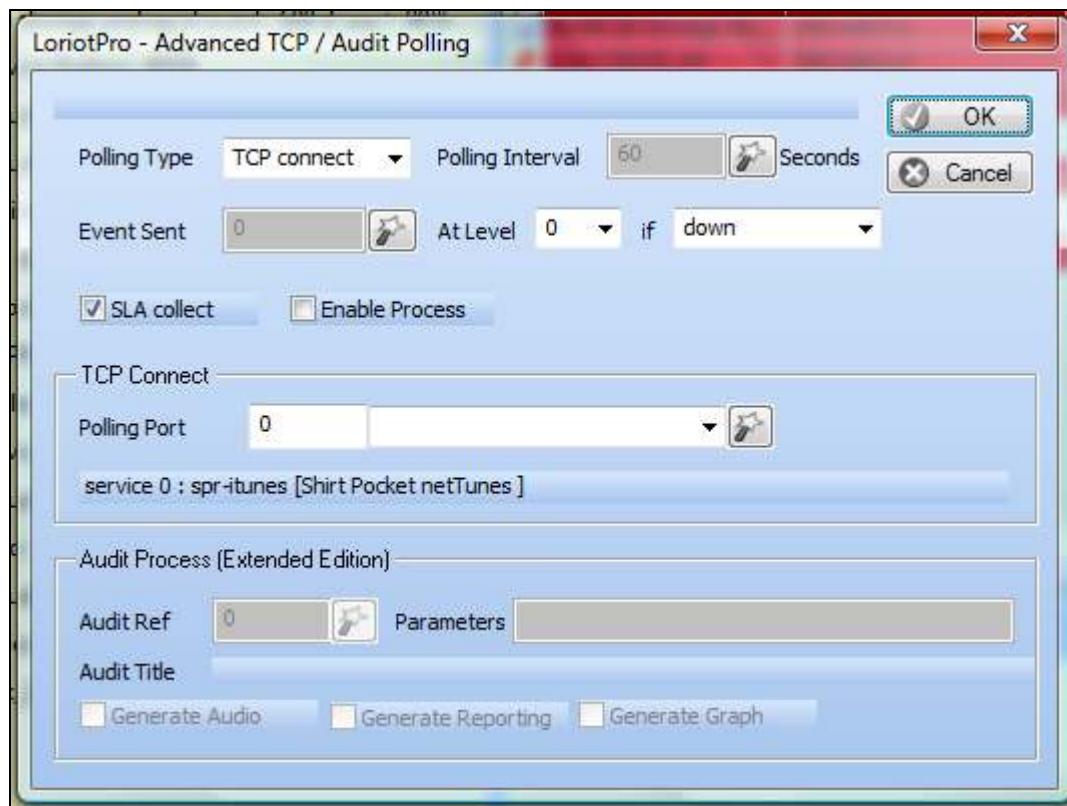
**Advanced Polling>Add**



Ces différents menus donnent accès à une dialog box générique qui autorise l'insertion de plusieurs nouveaux types de polling des équipements. Les scripts d'audits ne sont qu'un de ces nouveaux types de polling et sont ainsi considérés par le système. Cela nous permet de disposer de la gestion des fonctions de SLA que nous pouvons appliquer (avec réserve) au temps de déroulement d'un script d'audit.

### 10.3.1 Boîte de dialogue de sélection du type d'audit

La boîte de dialogue permet la configuration de plusieurs types d'audits. Par défaut, elle propose la configuration d'un audit (non scripté) de polling de port TCP.



Les paramètres généraux au process d'audit son.

#### 10.3.1.1 Options - Polling Interval

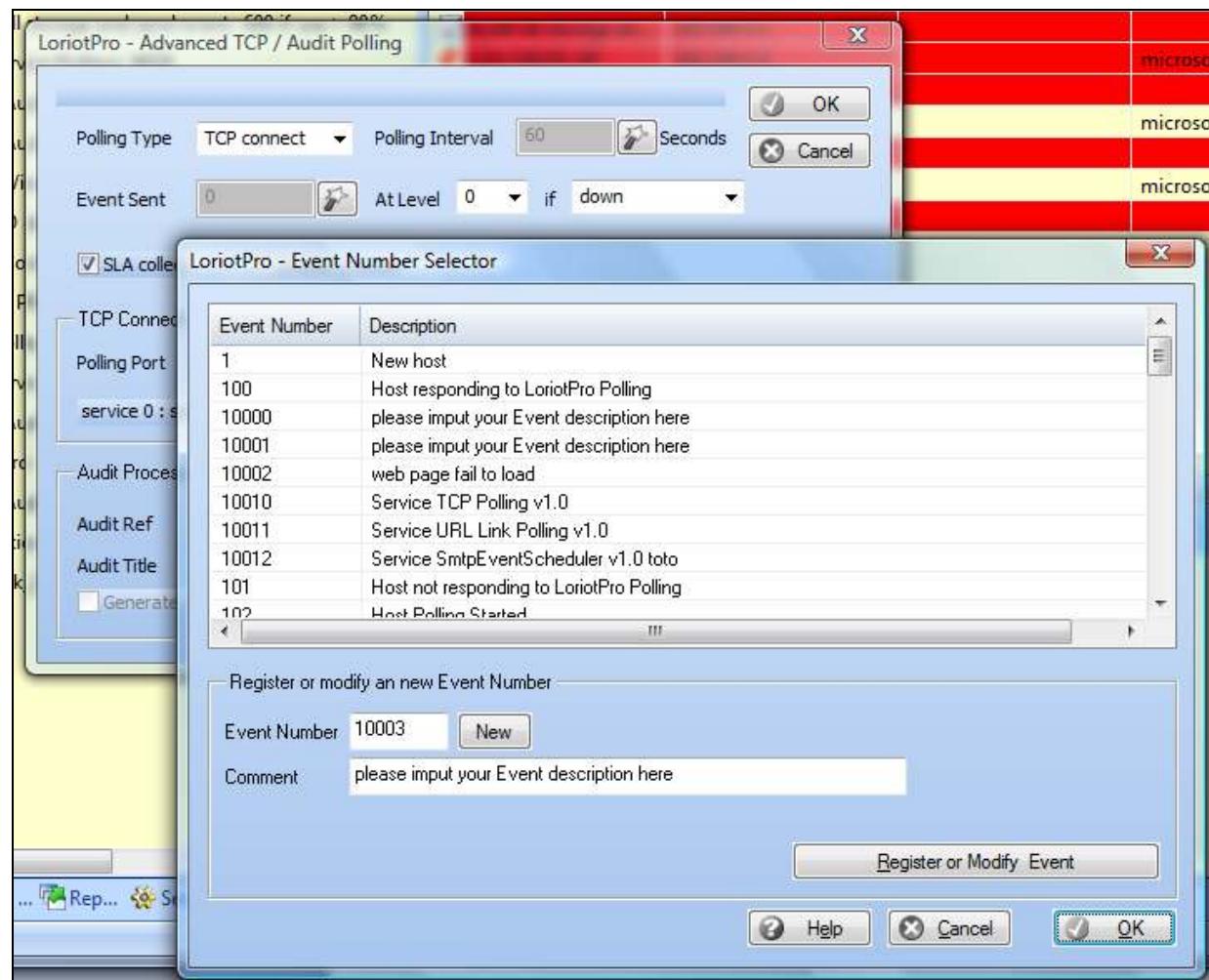
C'est le temps entre deux pollings (attention au mode best effort) ; cette option dispose d'un wizard pour vous aidez dans l'écriture de l'intervalle de temps.



### 10.3.1.2 Options - Event Sent

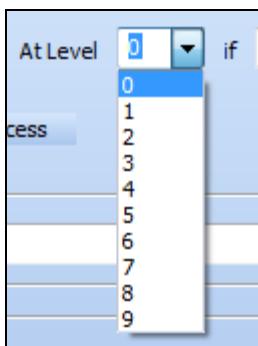
C'est le numéro d'événement utilisé si l'audit rencontre la nécessité d'envoyer un message au noyau de LoriotPro. Si le numéro d'Event est 0, il n'y aura pas de message envoyé, quel que soit le résultat de l'audit.

Cette option dispose d'un wizard.



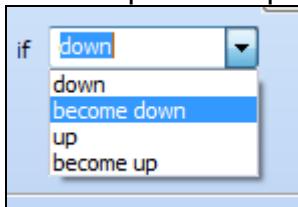
### 10.3.1.3 Options - At level

Le level du message Event envoyé :



#### 10.3.1.4 Options - If status

Vous disposez de plusieurs conditions de déclenchement de l'Event :



Status	Commentaire
down	L'audit ne répond pas.  Un message envoyé à chaque fois durant toute la période de status down.
Become down	L'audit vient de ne plus répondre mais il répondait au polling précédent  Un seul message envoyé, à chaque passage de up vers down
up	L'audit répond.  Un message envoyé à chaque fois durant toute la période de status up.
Become up	L'audit ne répondait pas et maintenant il répond.  Un seul message envoyé, à chaque passage de down vers up.
No sent	Le module de gestion des audits n'envoie pas d'évent quel que soit le résultat (up/down,). Par contre, le script d'audit, lui, peut envoyer des Events de type expertise en utilisant les paramètres de numéro d'Event et de Level s'il est programmé pour cette tâche.

#### 10.3.1.5 Options - SLA collect

Si cette case est cochée, le temps de réalisation de l'audit est collecté et sauvegardé dans un format compatible avec le module de gestion des SLA. Ces informations pourront être exploitées dans une logique de SLA avec les modules de gestion des SLA du logiciel. Il est possible bien sûr de gérer et exploiter les SLA par script.

---

**Attention** 

Si vous changez l'intervalle de polling, votre SLA sera moins pertinent, car vous visualiserez des périodes avec des pollings différents ce qui rendra les calculs incorrects.

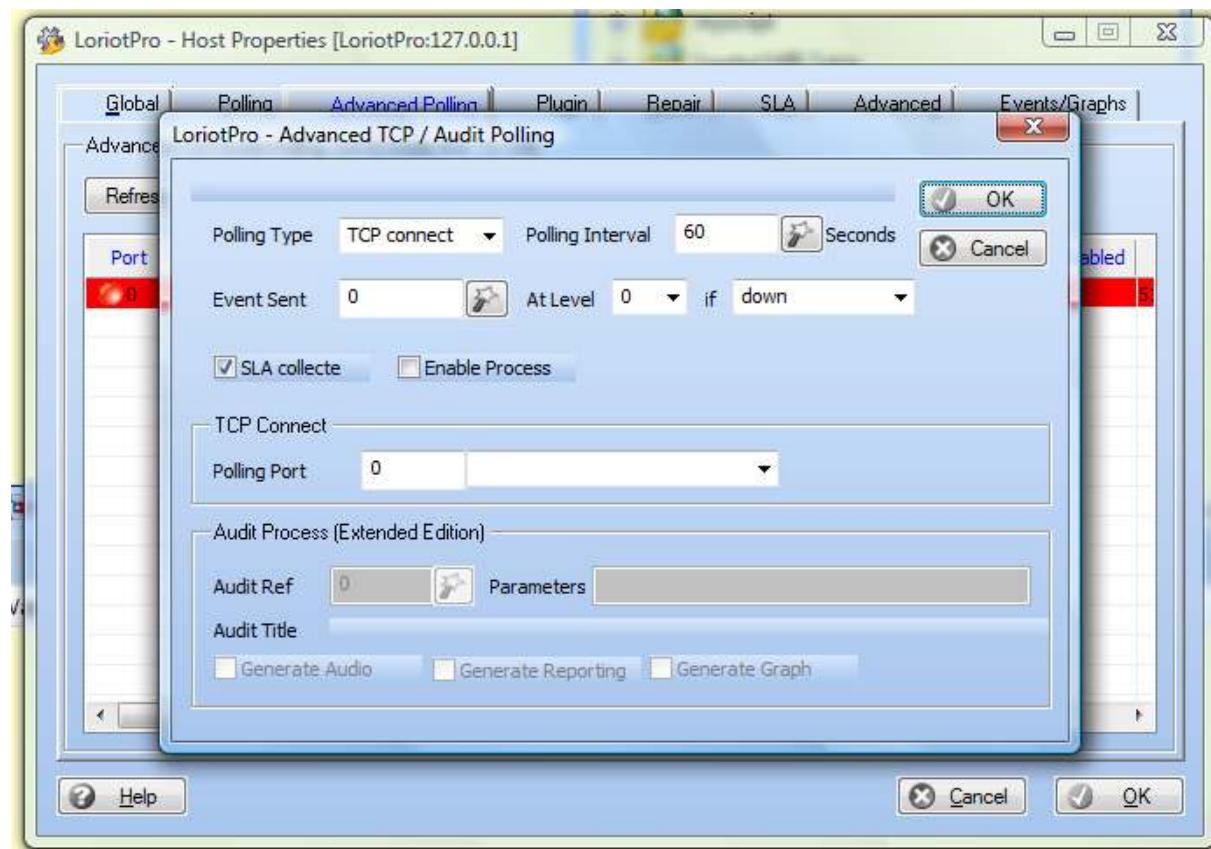
Les valeurs collectées correspondent uniquement au temps de traitement de l'audit, mais pas à son résultat. Dans le cas d'un audit scripté les éléments traités par le script peuvent être down mais le temps de traitement excellent.

#### 10.3.1.6 Options - Enable process

Cette case doit être cochée pour que l'audit soit réalisé. Il est possible de contrôler ces paramètres par script : contrôle du lancement ou de l'arrêt des audits sans intervention manuelle.

#### 10.3.1.7 Options - polling Type (TCP Connect)

La boîte de dialogue permet de positionner un audit non scripté géré par un autre noyau multitâche du logiciel.

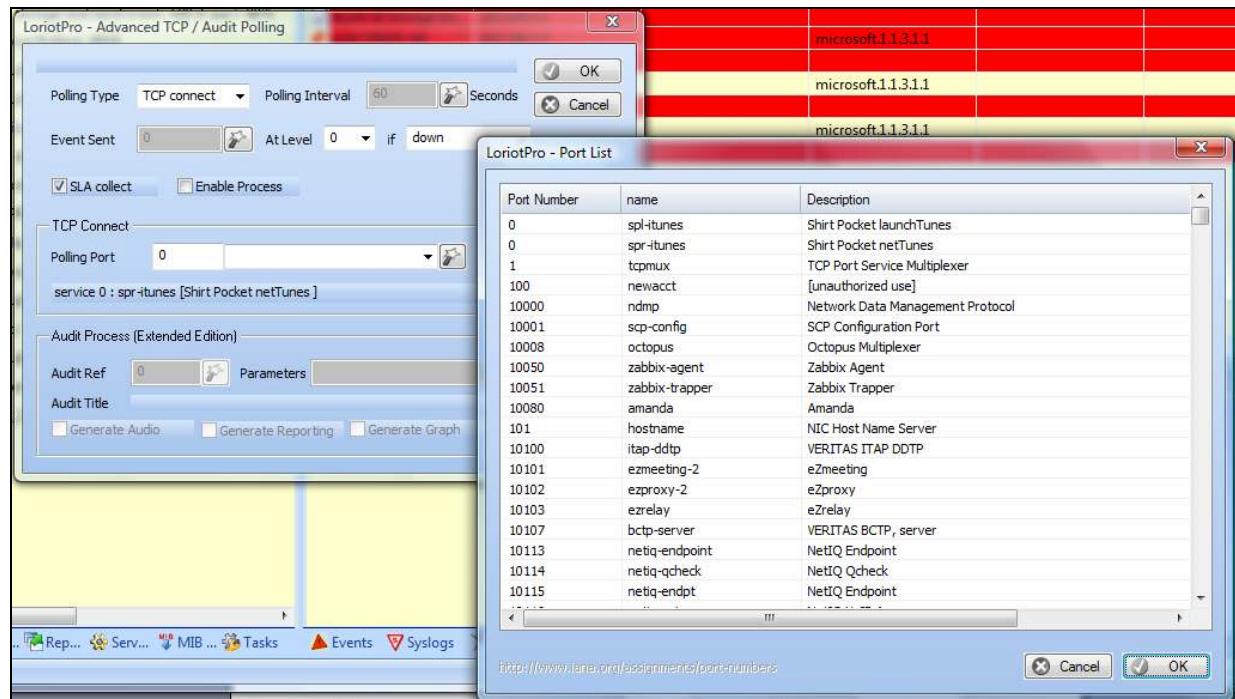


Cet audit dédié permet un polling de type asynchrone sur un port TCP donné de l'équipement sélectionné. Seuls les paramètres situés dans la zone TCP connectée sont valides.

### 10.3.1.8 Options - Polling Port

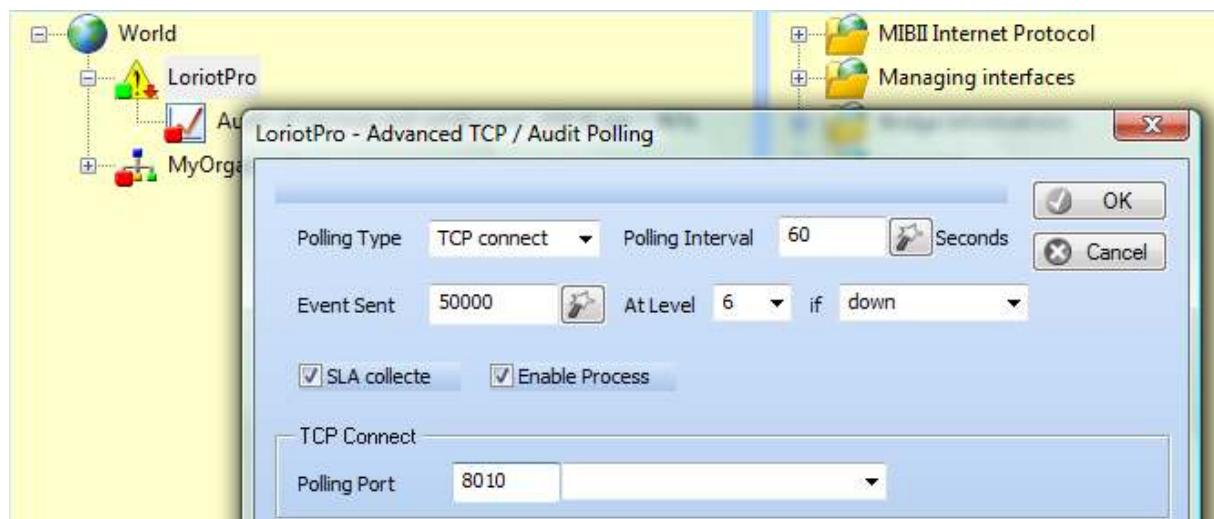
C'est le port TCP que vous désirez auditer.

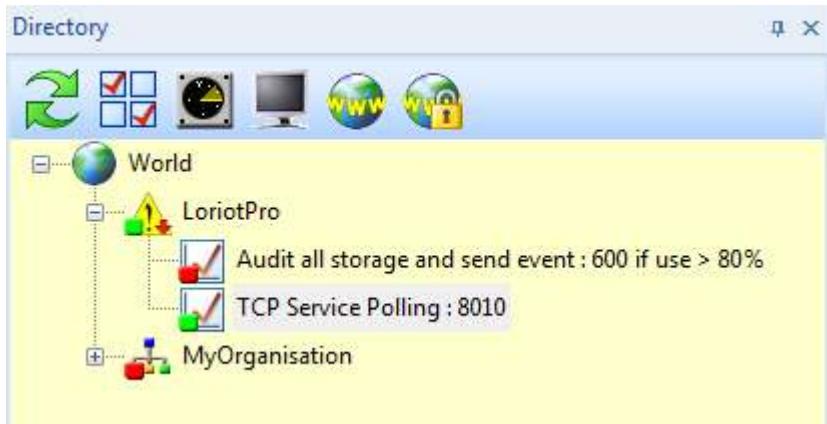
Le Wizard et les comboBox vous aident à sélectionner votre port TCP.



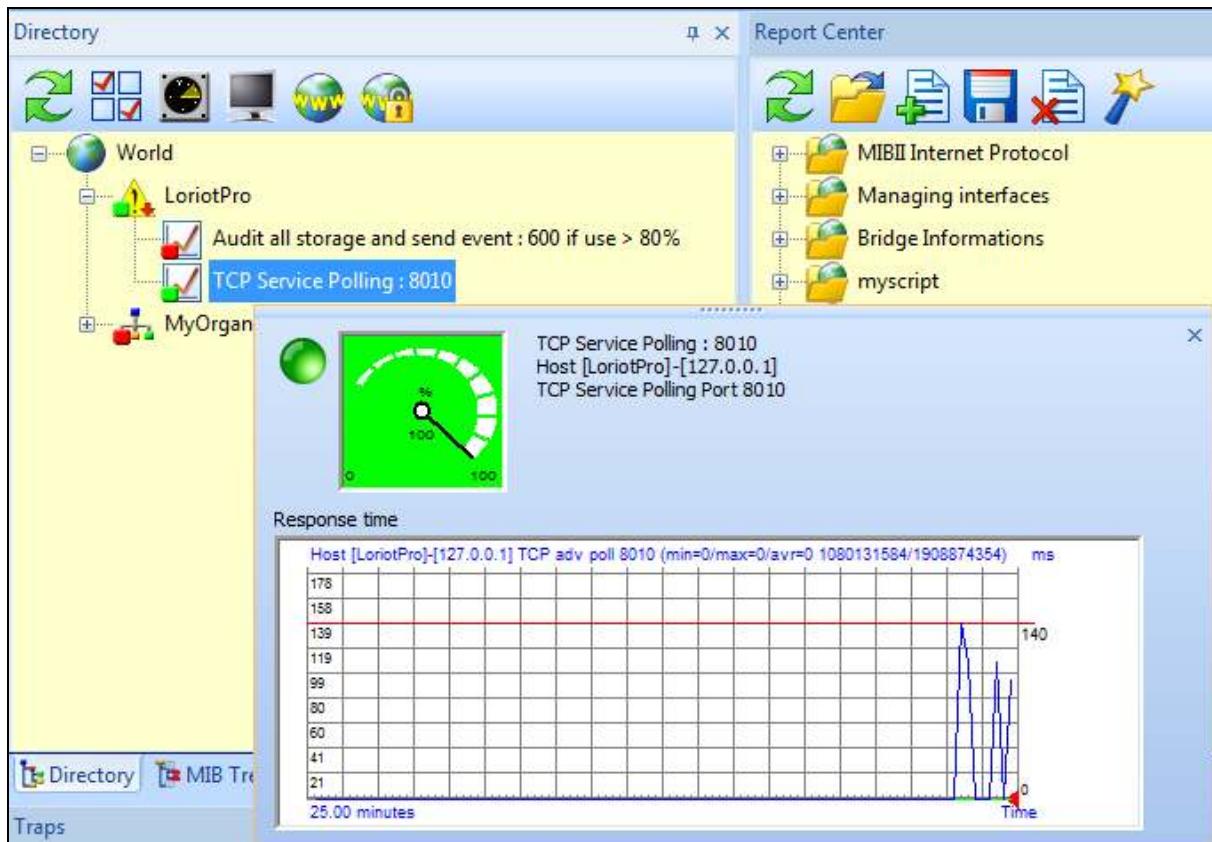
#### 10.3.1.8.1 Exemple

Le port 8010 du serveur Web intégré a LoriotPro est audité :



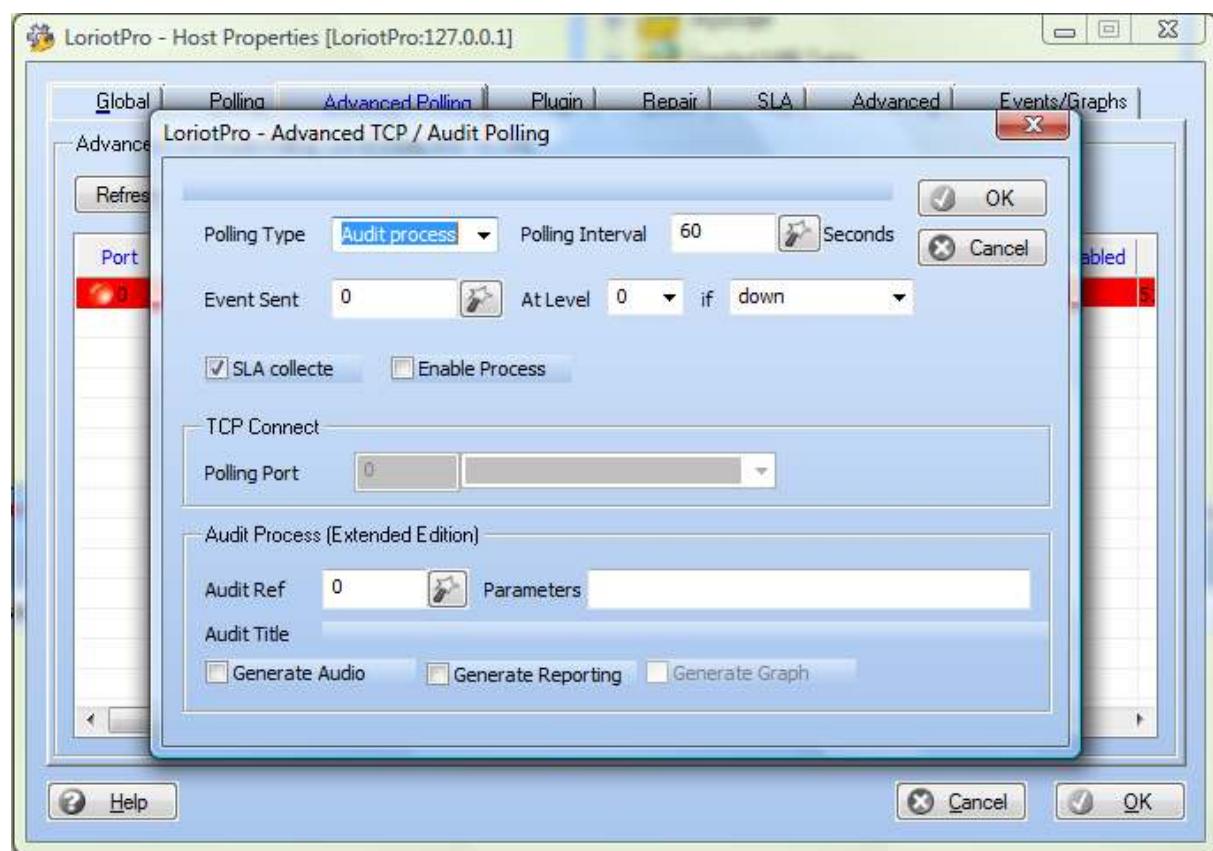


Un clic sur l'objet inséré dans la Directory nous fournit des informations liées à cet audit de port TCP à travers une fenêtre volante.



#### 10.3.1.9 Options - Polling Type (Audit process)

La boîte de dialogue permet de positionner un audit scripté géré par le module présenté dans le chapitre précédent.

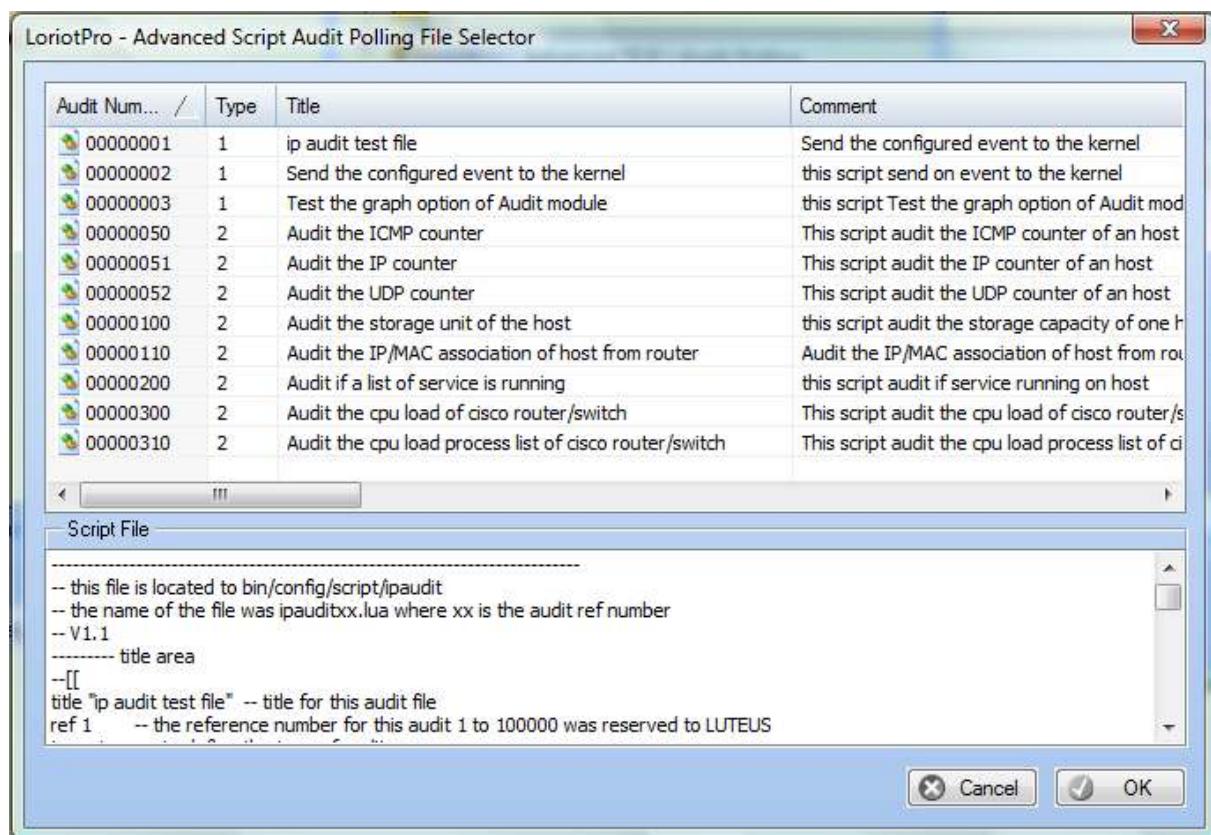


Seuls les paramètres situés dans la zone « Audit Process (Extended Edition) » sont valides.

#### 10.3.1.10 Options - Audit Ref

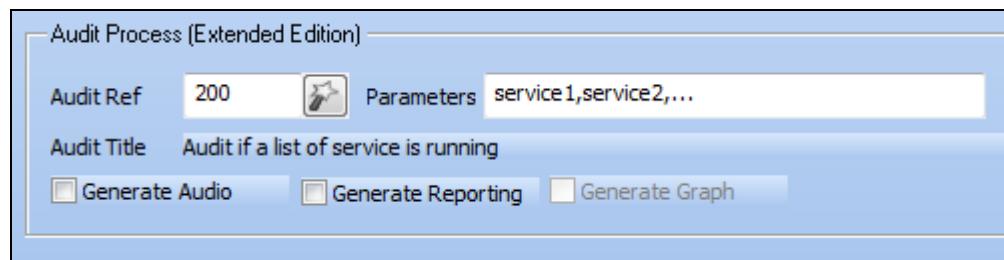
Les scripts d'audits sont structurés et possèdent chacun un numéro de référence unique. Ils sont construits sur le même modèle que les scripts IP Expert mais possèdent leur propre logique de variable.

Cette option contient un wizard qui sélectionne de façon graphique le script d'audit désiré.



### 10.3.1.11 Options - Parameters

Il est possible de fournir des paramètres au script d'audit comme par exemple des seuils d'alerte ou des noms de process à surveiller.



Par exemple l'audit 200 « Audit if a list of service is running »

Cet audit étant générique, les noms des services à auditer doivent être indiqués dans le champ paramètres.

#### Attention

Chaque script d'audit possède sa propre logique de paramètre, il faut consulter le texte associé au script pour renseigner ce champ correctement.

#### 10.3.1.12 Options - Generate Audio

Si vous cochez cette option, l'information est fournie au script d'audit sous la forme d'une variable. Si le script supporte la génération de message audio (service « text to speech »), il pourra générer des messages.

##### Attention

L'information envoyée au script est générique ; si le script ne supporte pas cette option alors il n'utilisera pas cette information.

Cette option étant générique, vous pouvez l'utiliser pour passer une validation d'option ou de fonction à votre script.

(voir les chapitres sur la création de scripts d'audits pour plus d'information)

#### 10.3.1.13 Options - Generate Reporting Options

Si vous cochez cette option, l'information est fournie au script d'audit sous la forme d'une variable. Si le script supporte la génération de rapport il pourra les générer.

##### Attention

L'information envoyée au script est générique ; si le script ne supporte pas cette option alors il n'utilisera pas cette information.

Cette option étant générique vous pouvez l'utiliser pour passer une validation d'option ou de fonction à votre script.

(voir les chapitres sur la création de scripts d'audits pour plus d'information)

#### 10.3.1.14 Options - Generate Graph Options

Cette option n'est pas paramétrable, c'est une information fournie par le script qui vous indique qu'il génère un graphe dynamique.

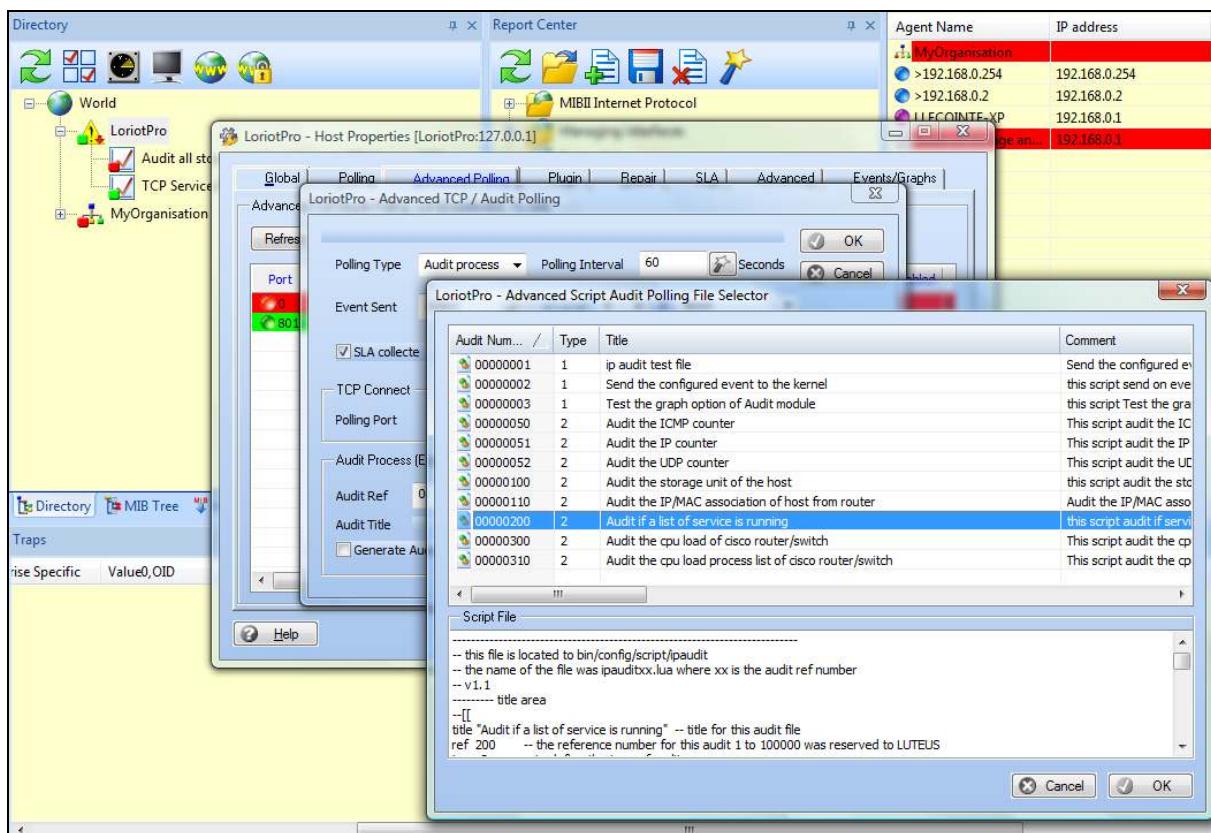
(Voir les chapitres sur la création de scripts d'audits pour plus d'information)

#### 10.3.2 Exemples

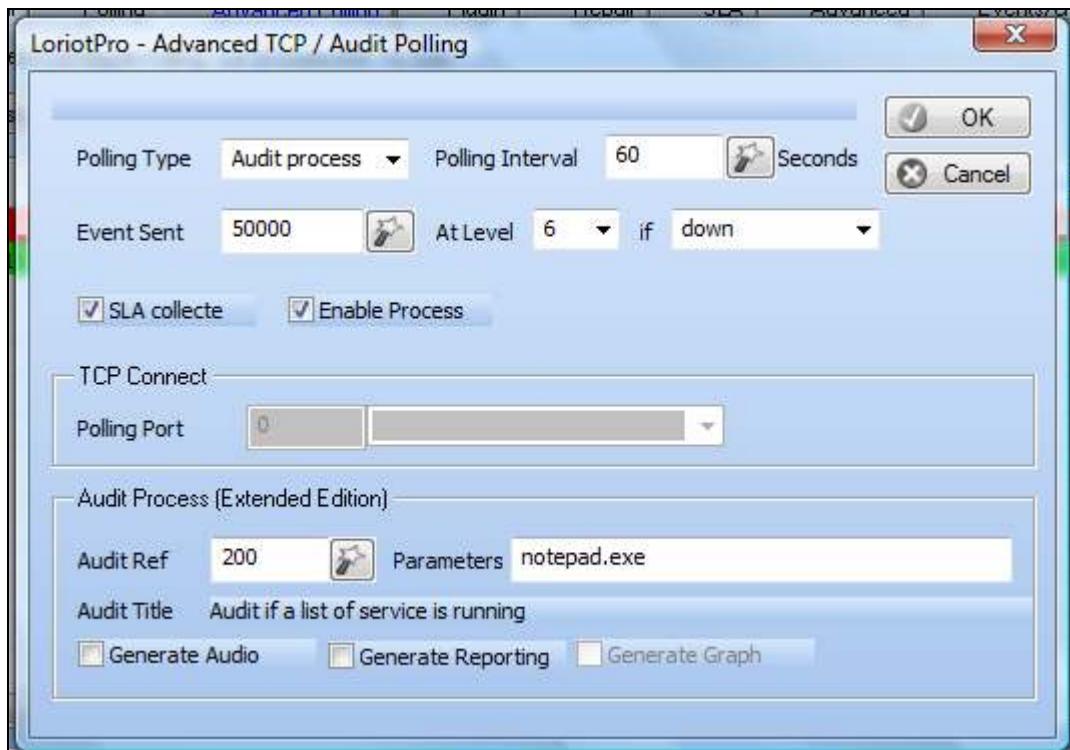
Insertion d'un script d'audit de service pour vérifier si le logiciel **notepad.exe** tourne sur un équipement.

##### Attention

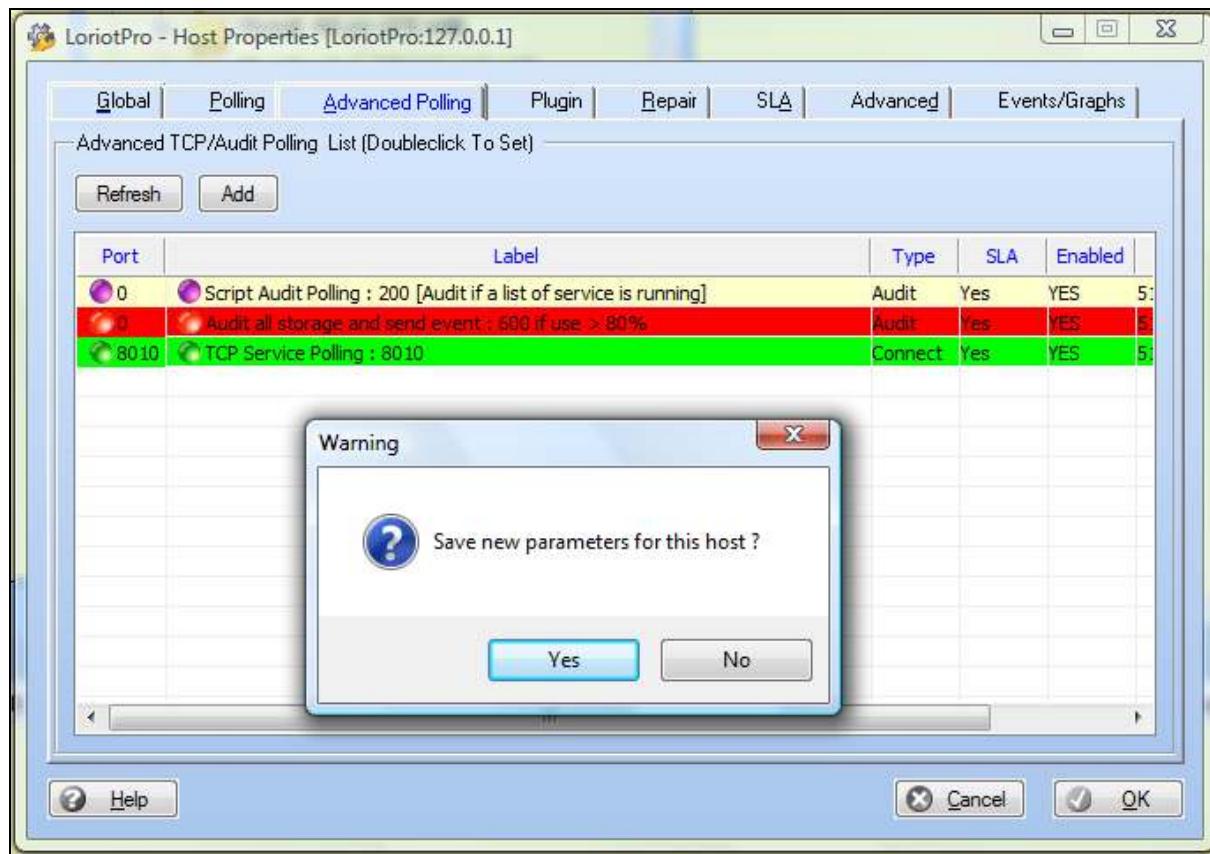
L'équipement doit répondre aux requêtes SNMP et supporter la HOST-RESOURCES-MIB.



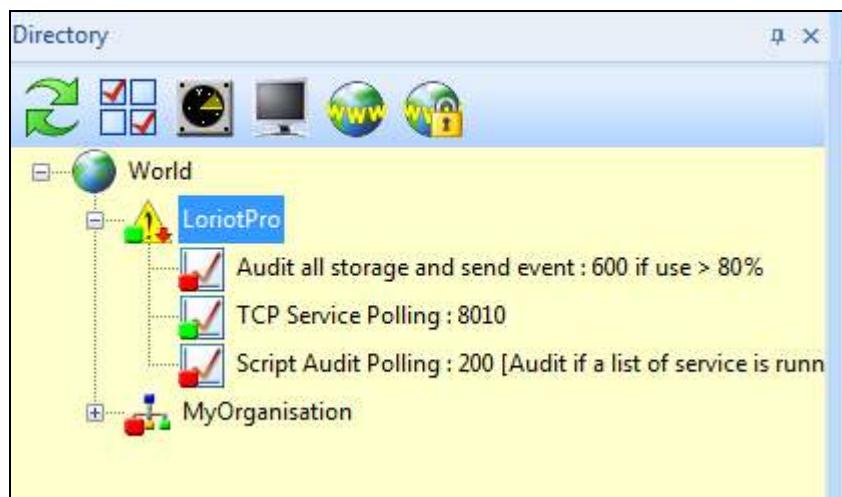
### Sélection du script d'audit



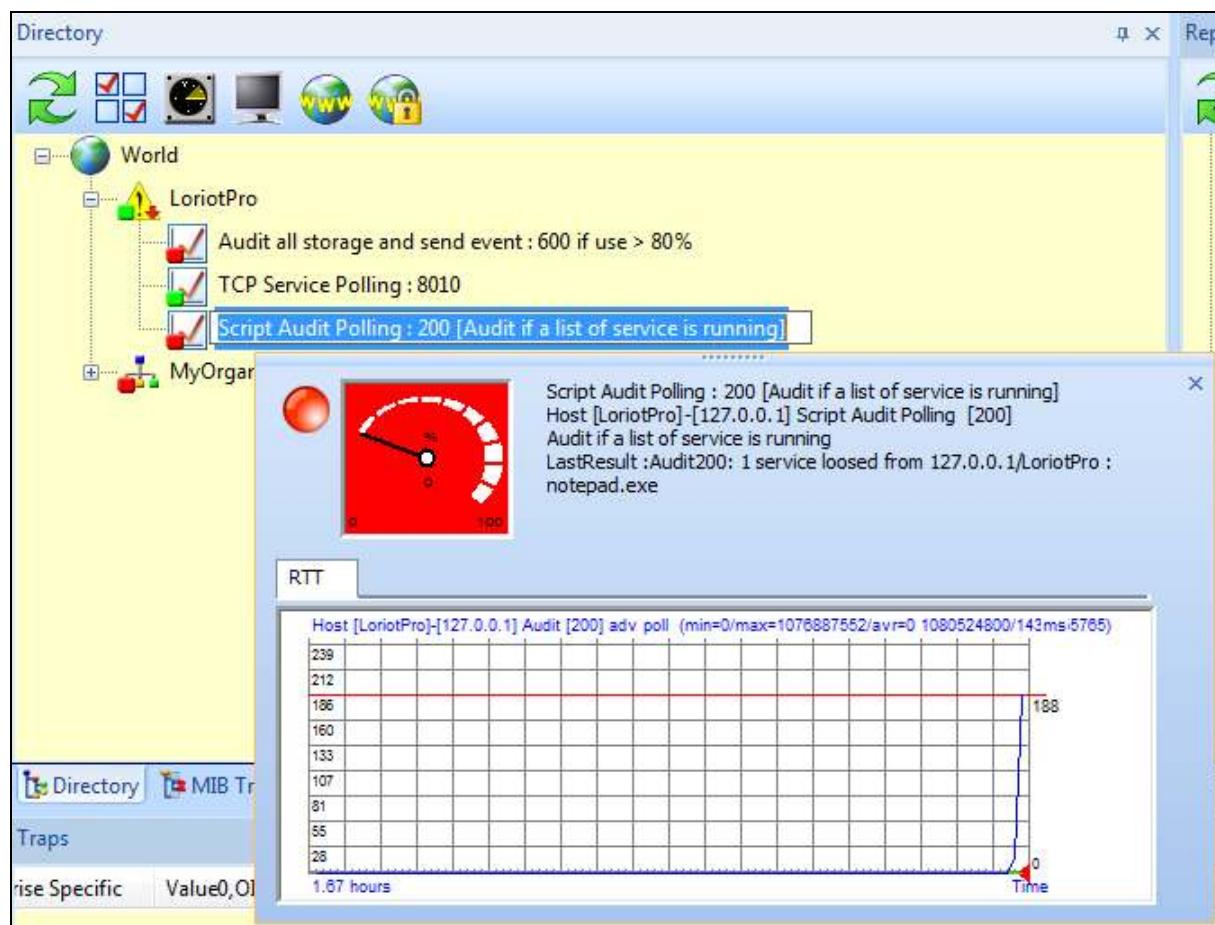
On précise le nom du logiciel qui doit être lancé sur l'équipement, vous validez notre audit : notepad.exe



Le script d'audit est ajouté aux tâches associées à l'équipement.



Le fenêtre volante associée à l'audit nous informe sur son statut (rouge pour statut down) et fournit un commentaire.



On peut remarquer que le RTT est implémenté et donne des résultats uniquement sur le temps de traitement de l'audit (pas sur le statut).

Le système reçoit des Events 50000 (configurés) au niveau 6 pour vous informez du problème soulevé par l'audit.

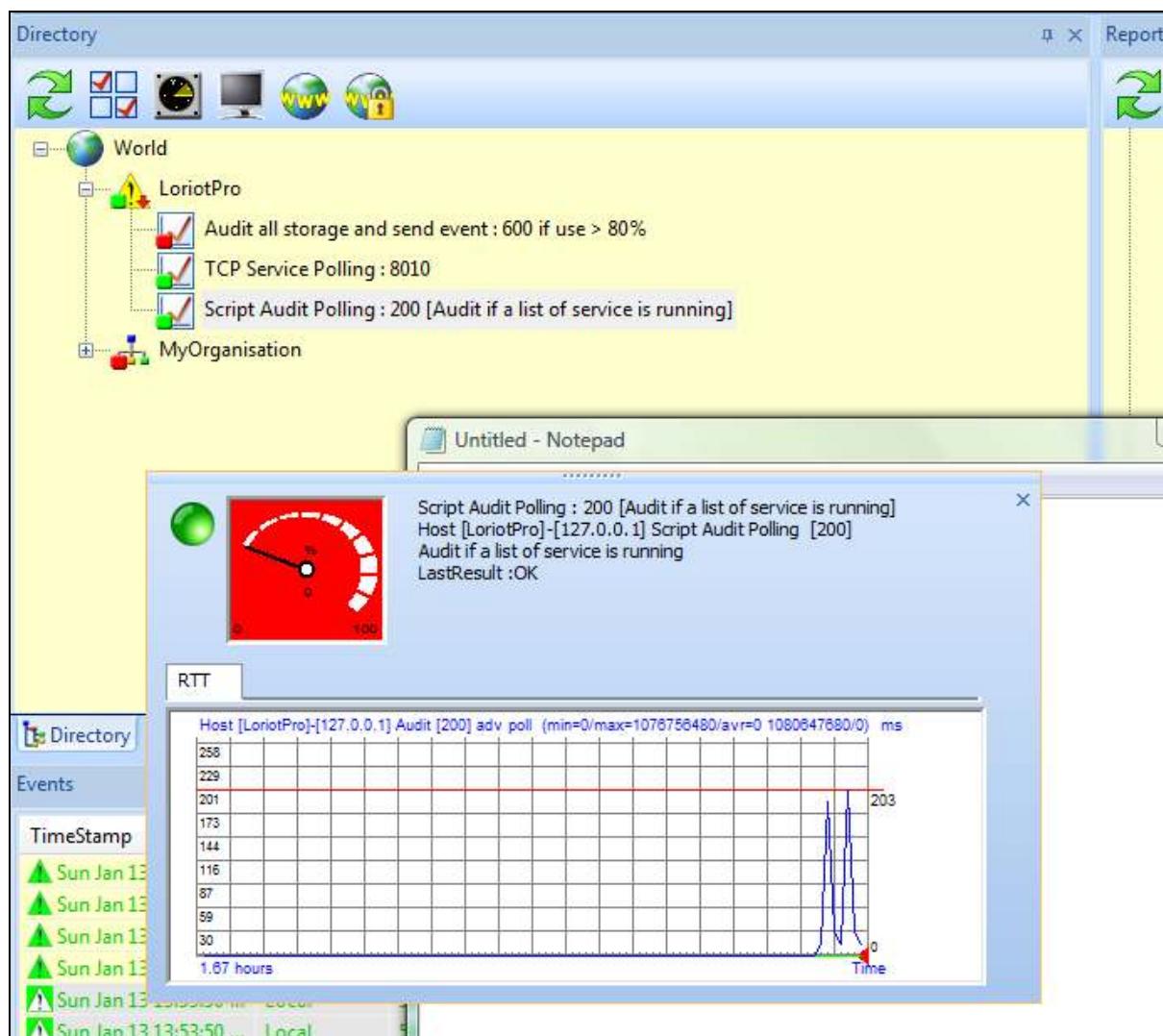
Events				
TimeStamp	LoriotPro ...	Ref N°	IP Ref	Alert
[Sun Jan 13 13:51:47 ...]	Local	50000	[LoriotPr...]	Adv_Audit: 127.0.0.1 200(0) Audit if a list of service is running is down
[Sun Jan 13 13:51:47 ...]	Local	50000	[LoriotPr...]	Audit200: 1 service loosed from 127.0.0.1/LoriotPro : notepad.exe
[Sun Jan 13 13:50:45 ...]	Local	50000	[LoriotPr...]	Adv_Audit: 127.0.0.1 200(4) Audit if a list of service is running is down
[Sun Jan 13 13:50:45 ...]	Local	50000	[LoriotPr...]	Audit200: 1 service loosed from 127.0.0.1/LoriotPro : notepad.exe

Vous pouvez bien sûr réaliser des filtres d'événements et lancer des actions à réception de ces Events.

Le Logiciel **notepad.exe** est maintenant lancé sur l'équipement.

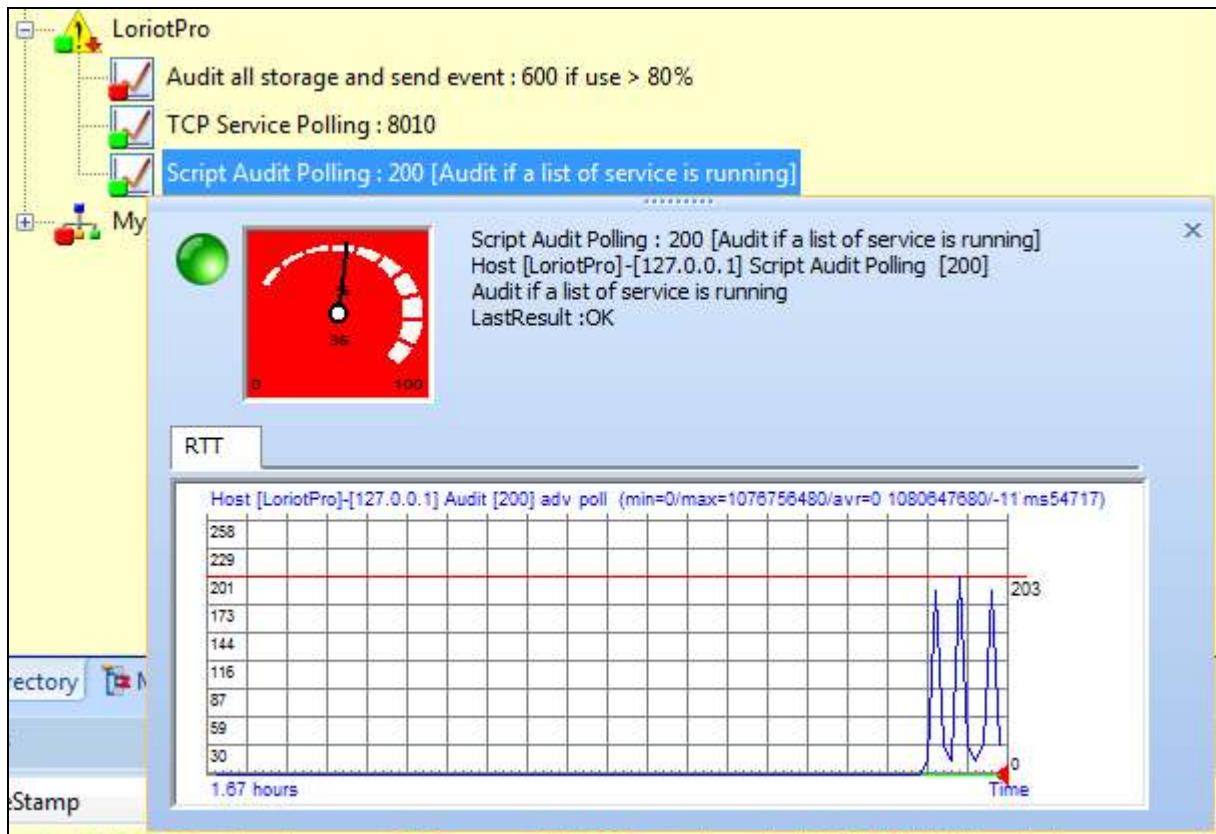
**Attention**

La mise à jour des tables de process n'est pas instantanée sur les équipements Microsoft et peut prendre plusieurs minutes.



Le statut de l'audit passe au vert, le vumètre de pourcentage de bonnes réponses est rouge car il y a plus de mauvaises réponses que de correctes pour l'instant.

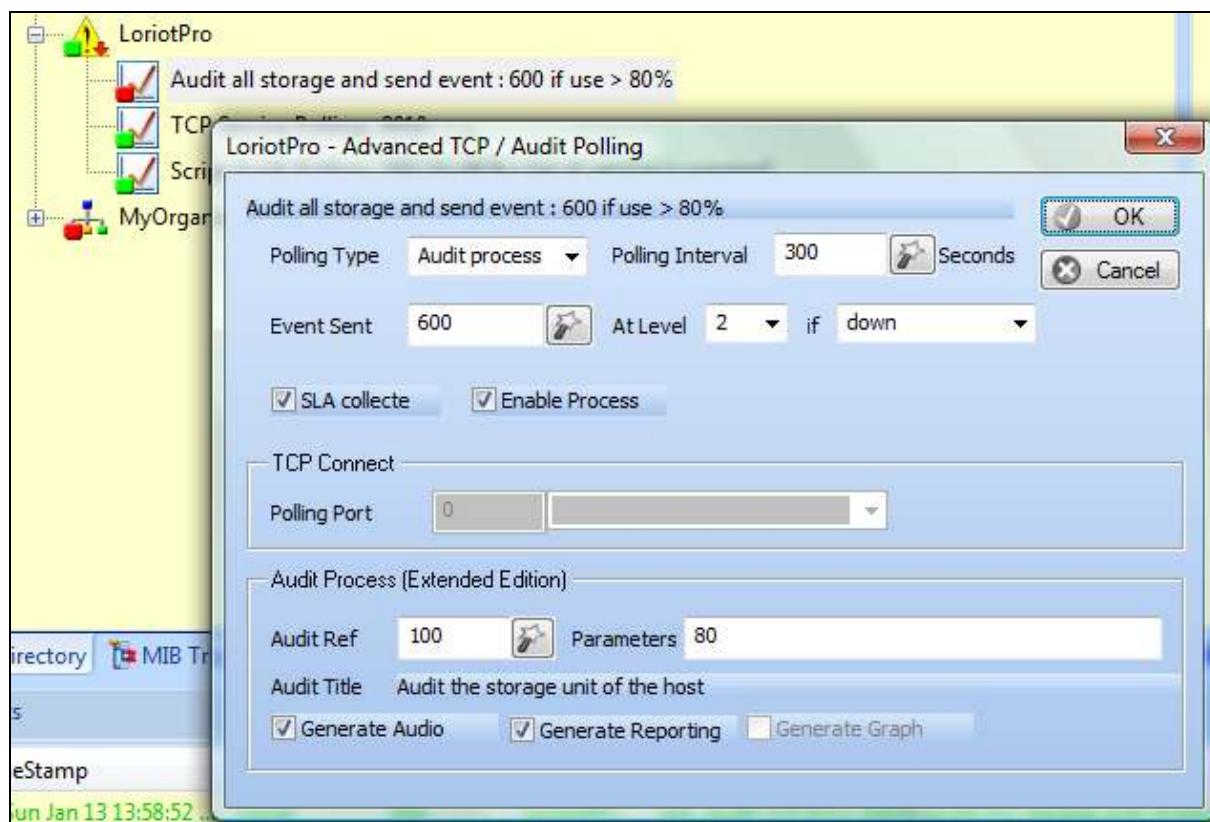
Après quelques pollings, le pourcentage change.



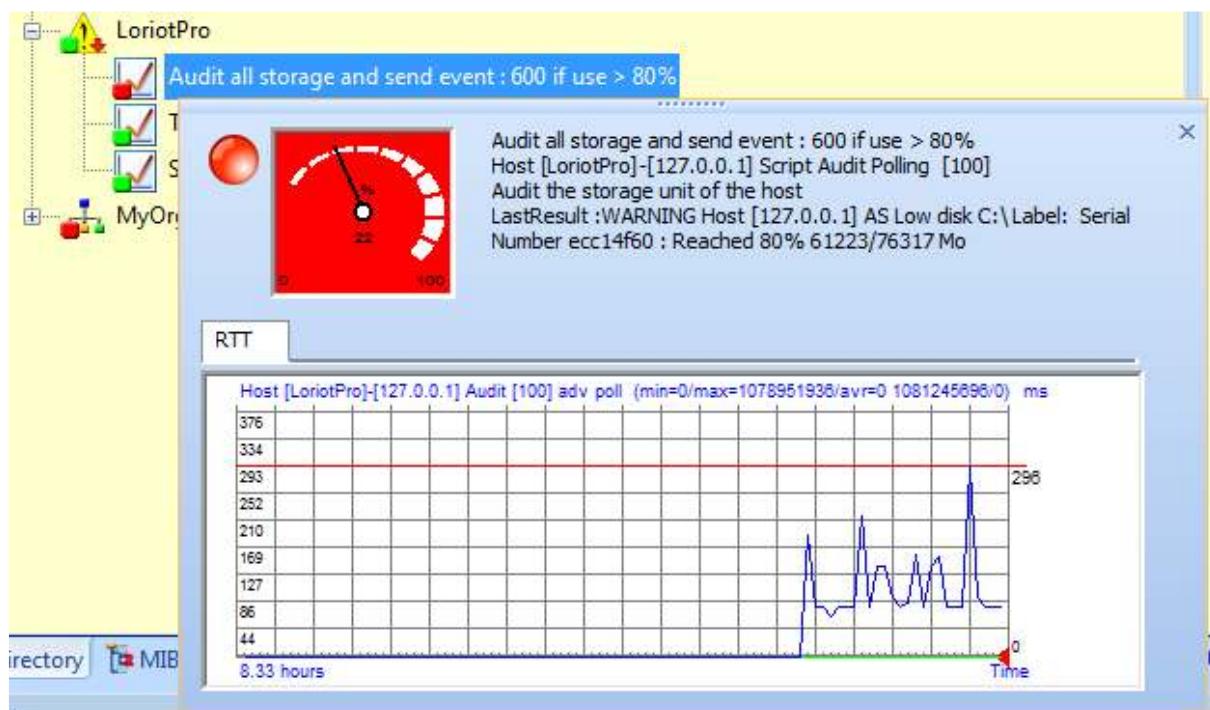
Exemple : pourcentage d'utilisation des disques durs d'un équipement.

**Attention**

L'équipement doit répondre aux requêtes SNMP et supporter la HOST-RESOURCES-MIB.



La fenêtre volante associée à l'audit nous fournit des informations complémentaires.



Des Events sont générés.

Events				
TimeStamp	LoriotPro ...	Ref N°	IP Ref	Alert
▲ Sun Jan 13 14:00:23 ...	Local	600	[LLECOINTE-XP] 192...	adv_audit: 192.168.0.1 100(0) audit the storage unit of the host is down
▲ Sun Jan 13 14:00:23 ...	Local	600	[LLECOINTE-XP] 192...	warning host [192.168.0.1] as low disk c:\ label:windows serial number 2e2241b : reached 95% 14690/15335 mo

## 10.4 Insertion automatique via un script

Il est possible d'utiliser un script pour insérer dynamiquement un audit sur un équipement. La fonction utilisée est **Ip.InsertADVAuditPolling**.

Voici un extrait de code tiré du script **ipscanner-101.lua** du module IP Expert qui réalise cette opération. Il s'agit de la fonction **Audit\_SetDefaultParam\_Server** qui insère un test d'audit de capacité de disque dur sur l'équipement si celui-ci supporte SNMP et la HOST-RESOURCES-MIB .

La partie qui nous concerne est coloré en bleu :

```
=====
function Audit_SetDefaultParam_server(ip)

if (ip==nil) then return 0; end

if (ADV_NOAUTOCONFIGURATION==1) then return 0; end
if (A101_1==nil) then return 0; end

    --verifie que l'ip existe dans la Directory
    if lp.GetIPInformation(ip,"array") then
        lp.Trace(array.name);
    else
        return 0;
    end

    lp.SetIPOption(ip,L_P_LUA_ISDEVICE,L_P_LUA_ISDEVICE_SERVER);
    lp.SetIPOption(ip,L_P_LUA_POLLING_PING,L_P_LUA_ON);
    lp.SetIPOption(ip,L_P_LUA_POLLING,30); --30 secondes
    lp.SetIPOption(ip,L_P_LUA_SLA,L_P_LUA_ON);
    lp.SetIPOption(ip,L_P_LUA_IPSCANNER_INTERVAL,300); --5 minutes

    -- insert un test des disks
    if (AUDIT_SNMP_AVAILABLE==1 and AUDIT_HOST_RESOURCES_MIB==1 and A101_21==1) then
        str=string.format("Audit all storage and send event : %i if use > 80%%", ADV_AUDIT_EVENT_NUMBER );
        uid1,ret=lp.InsertADVAuditPolling(ip,100,"80",300,1,1,1,ADV_AUDIT_EVENT_NUMBER,ADV_AUDIT_EVENT_LEVEL,1,1,str);
        if uid1 then lp.Print(uid1,"2 3 insert Storage Audit \n"); end
    end

    str=string.format("2 3 %cHost configured for : DEVICE_SERVER %c\n",34,34)
    lp.Print(str);
    lp.Fprintf(fp,str);
    ISCONFIGUREDTO=L_P_LUA_ISDEVICE_SERVER;
    ISCONFIGURED=1;
    return 1;

end
```

**Attention** 

Si vous essayez d'insérer plusieurs fois le même audit avec les mêmes paramètres, le logiciel se contente de faire un update de certains paramètres de l'audit déjà insérés. Consultez le référence manuel des librairies LUA de Loriotpro pour plus d'information.

Consultez le manuel de référence des librairies pour obtenir plus d'information sur les commandes d'insertion scriptées d'audits.

```
lp.InsertADVAuditPolling(  
    'ip',  
    audit_ref,  
    'param',  
    poll_interval,  
    enable,  
    gen_audio,  
    gen_report,  
    event,  
    level,  
    condition,  
    sla,  
    ['title']);  
  
lp.InsertADVTCPPolling(  
    'ip',  
    tcp_port,  
    poll_interval,  
    enable,  
    gen_audio,  
    gen_report,  
    event,  
    level,  
    condition,  
    sla,  
    ['title']);  
  
lp.UpdateADVAuditPolling(  
    'ip',  
    audit_ref,  
    'param',  
    poll_interval,  
    enable,  
    gen_audio,  
    gen_report,  
    event,  
    level,  
    condition,  
    sla);  
  
lp.UpdateADVTCPPolling(  
    'ip',  
    tcp_port,  
    poll_interval,  
    enable,  
    gen_audio,  
    gen_report,  
    event,  
    level,  
    condition,  
    sla);  
  
lp.UpdateADVTCPPollingUID(  
    'uid',  
    poll_interval,  
    enable,  
    gen_audio,  
    gen_report,
```

--l'adresse IP de l'équipement  
--Le numéro de reference de l'audit  
--les parameters associés à l'audit  
-- l'intervalle de polling  
-- 1 enable 0 disable  
-- 1 desired 0 not desired  
-- 1 desired 0 not desired  
-- le numero de l'event de reference 0 si pas d'Event  
-- le level des Events émis  
-- down, up, become down, become up  
-- 1 desired 0 not desired  
-- le titre dans la Directory

--l'adresse IP de l'équipement  
-- le port TCP pollé  
-- l'intervalle de polling  
-- 1 enable 0 disable  
-- 1 desired 0 not desired  
-- 1 desired 0 not desired  
-- le numero de l'event de reference 0 si pas d'Event  
-- le level des Events émis  
-- down, up, become down, become up  
-- 1 desired 0 not desired

--l'adresse IP de l'équipement  
--Le numéro de reference de l'audit  
--les parameters associés à l'audit  
-- l'intervalle de polling  
-- 1 enable 0 disable  
-- 1 desired 0 not desired  
-- 1 desired 0 not desired  
-- le numero de l'event de reference 0 si pas d'Event  
-- le level des Events émis  
-- down, up, become down, become up  
-- 1 desired 0 not desired

--l'adresse IP de l'équipement  
-- le port TCP pollé  
-- l'intervalle de polling  
-- 1 enable 0 disable  
-- 1 desired 0 not desired  
-- 1 desired 0 not desired  
-- le numero de l'event de reference 0 si pas d'Event  
-- le level des Events émis  
-- down, up, become down, become up  
-- 1 desired 0 not desired

-- l'UID dans la Directory de l'audit de type TCP connect  
-- l'intervalle de polling  
-- 1 enable 0 disable  
-- 1 desired 0 not desired  
-- 1 desired 0 not desired

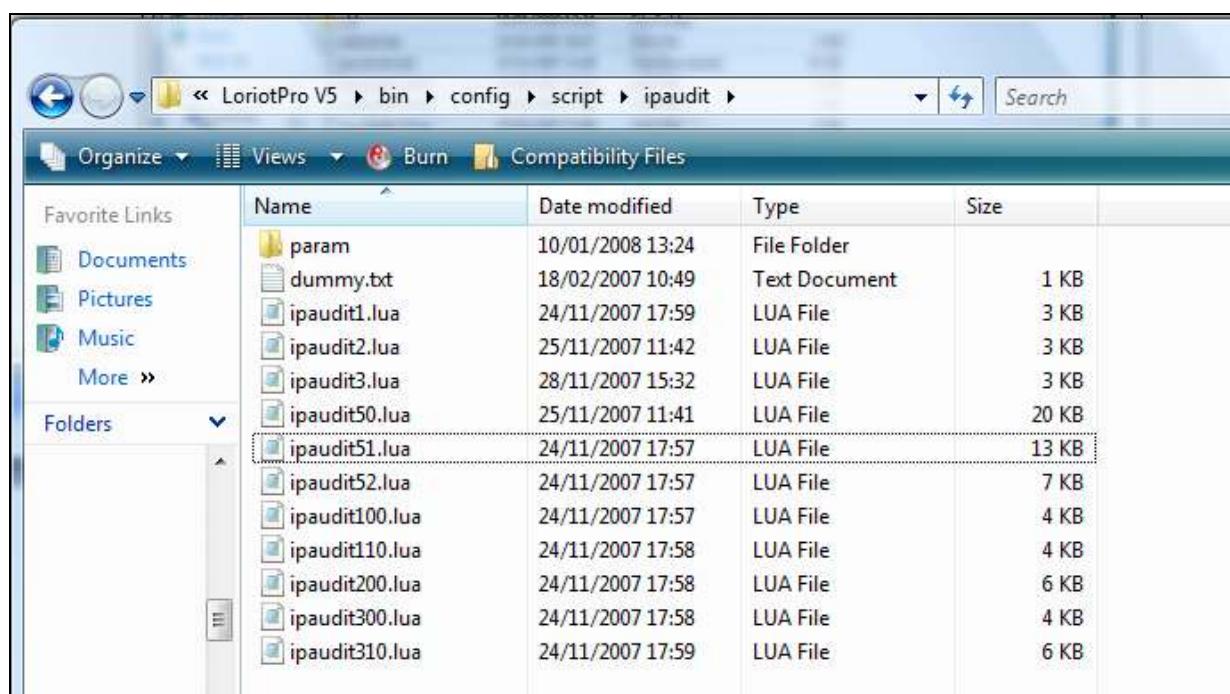
```
event,          -- le numero de l'event de reference 0 si pas d'Event
level,          -- le level des Events émis
condition,      -- down, up, become down, become up
sla);           -- 1 desired 0 not desired
```

## 10.5 Génération de Scripts pour le module d'Audit

Les scripts utilisés par le module d'Audit sont structurés afin de rendre opérationnel le module d'insertion et faciliter ainsi la tâche de l'administrateur.

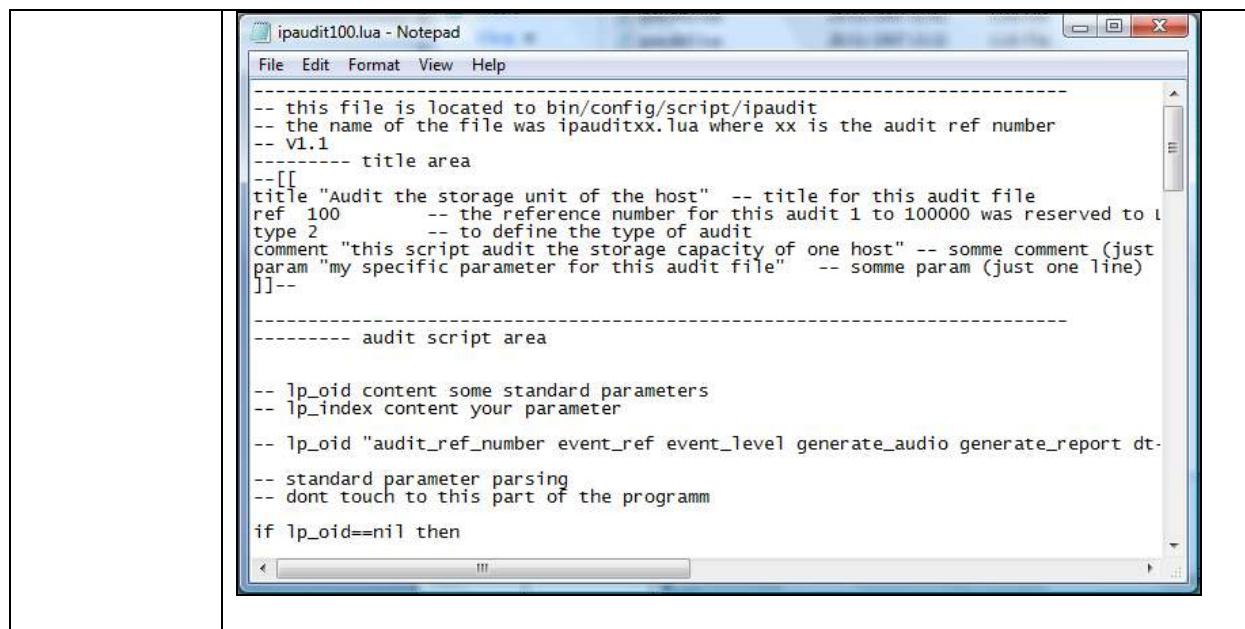
Tous les scripts du module d'audit sont situés dans le répertoire :

**Bin/config/script/ipaudit** du logiciel. Tous les fichiers ont un nom structuré afin que l'administrateur dispose de boîtes de sélection graphique permettant un choix aisément des options.



Ce répertoire contient les fichiers d'audits et un répertoire **param** utilisé par les fonctions des scripts d'audits pour faire une sauvegarde de paramètres.

Extention	Usage
ipauditx.lua	Ce sont les fichiers de scripts des modules d'audits. Le X représente le numéro unique de référence assigné au script.



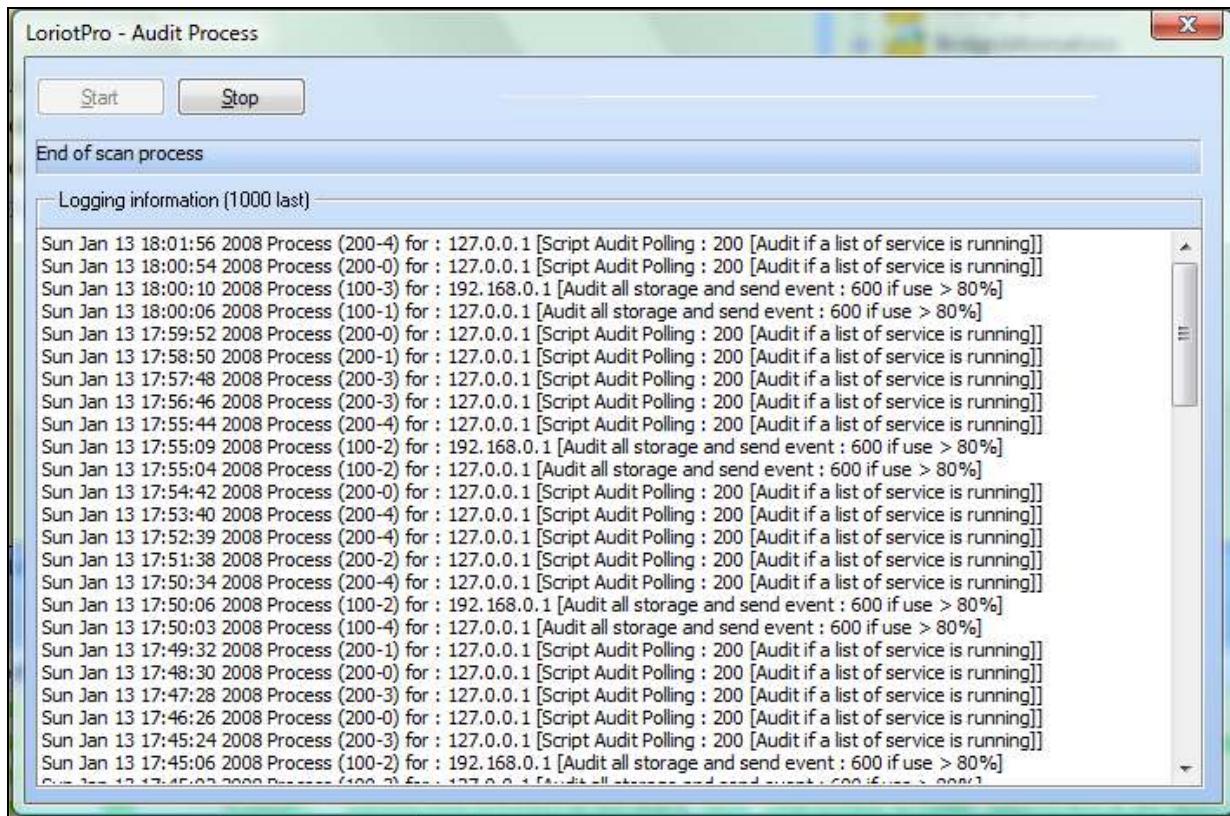
```
ipaudit100.lua - Notepad
File Edit Format View Help
-----
-- this file is located to bin/config/script/ipaudit
-- the name of the file was ipauditxx.lua where xx is the audit ref number
-- V1.1
----- title area
---[[[
title "Audit the storage unit of the host" -- title for this audit file
ref 100 -- the reference number for this audit 1 to 100000 was reserved to 1
type 2 -- to define the type of audit
comment "this script audit the storage capacity of one host" -- somme comment (just
param "my specific parameter for this audit file" -- somme param (just one line)
]]]

----- audit script area

-- lp_oid content some standard parameters
-- lp_index content your parameter
-- lp_oid "audit_ref_number event_ref event_level generate_audio generate_report dt-
-- standard parameter parsing
-- dont touch to this part of the programm
if lp_oid==nil then
```

### 10.5.1 Principe de lancement des scripts d'audit

Les scripts sont insérés manuellement ou automatiquement (par script) dans la Directory en association avec un équipement. Ils ont le paramètre « enable » et un intervalle de polling défini. Le module noyau de lancement des scripts d'audits parcourt en permanence la chaîne des objets de la Directory de type ADV\_AUDIT et, en fonction de la valeur de l'intervalle de polling, exécute le script ou passe au suivant. Pour que l'exécution d'un script ne bloque pas l'exécution des autres scripts, le module dispose de 5 threads lui permettant de distribuer les scripts et d'avoir au maximum 5 scripts concourants.

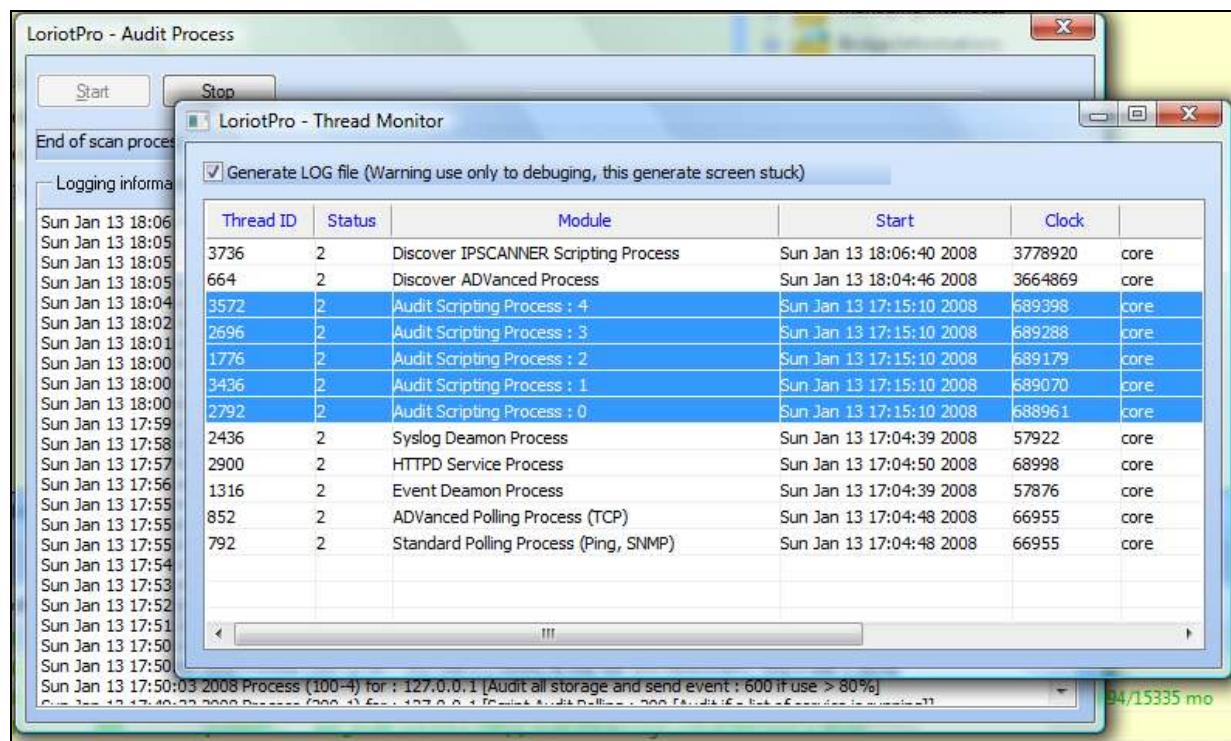
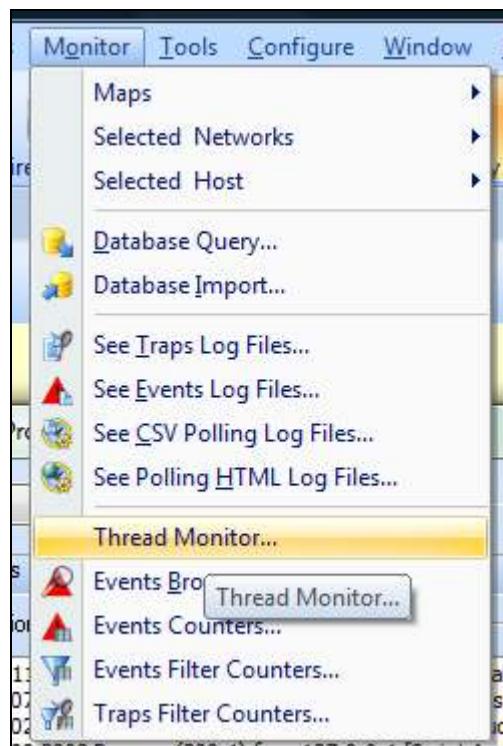


On peut voir dans la fenêtre de log une information concernant le numéro de référence et le numéro du thread utilisé pour son exécution (200-1).

#### ***Time Process (Numero\_d\_audit-Numero\_de\_thread) for : IP [titre + information]***

Les threads sont numérotés de 0 à 4.

Le thread monitor nous informe sur la présence de ces 5 threads.



En cas de blocage d'un script (boucle de programme), les informations **start** et **clock** seront bloquées sur des valeurs permettant une analyse du problème. Les fonctions Stop / Start du module d'audit résoudront peut-être un éventuel problème et libéreront les threads bloqués. Dans tous les cas, il faudra analyser la raison de ce blocage dans votre code.

## 10.5.2 Structure d'un fichier de scripts d'audits

Chaque fichier de script d'audit possède une référence unique qui permet au système d'utiliser le fichier dynamiquement. Les références 1 à 99999 sont réservées à la société Luteus et ne doivent pas être utilisées pour créer de nouveaux scripts. Le wizard de création de scripts d'audits intégré à l'environnement de développement gère automatiquement la numérotation des nouveaux scripts.

L'entête d'un fichier ipauditX.txt est structurée pour permettre au module de sélection de construire sa liste de choix.

### 10.5.2.1 Entête d'un fichier ipaudit.lua

Prenons en exemple le fichier ipaudit100.lua déjà utilisé :

ipaudit100.lua	Commentaires
----- ---- -- this file is located to bin/config/script/ipaudit -- the name of the file was ipauditxx.lua where xx is the audit ref number -- V1.1	Cette zone de commentaire est libre et contient les copyrights du script.
----- title area --[[	Ici commence la zone de définition des fonctions et options disponibles avec ce script.
title "Audit the storage unit of the host" -- title for this audit file	title "Un titre pour votre script entre guillemets »
ref 100 -- the reference number for this audit 1 to 100000 was reserved to LUTEUS	ref xx xx le numéro de référence du script. Le fichier porte aussi le numéro de référence dans son nom.
type 2 -- to define the type of audit	Information réservée
comment "this script audit the storage capacity of one host" -- somme comment (just one line)	comment "Un commentaire pour votre script entre guillemets »
param "my specific parameter for this audit file" -- somme param (just one line)	param "des paramètres de type texte ,ou un commentaire

	pour expliquer les paramètres attendus par le script »
]]--	La fin de l'entête

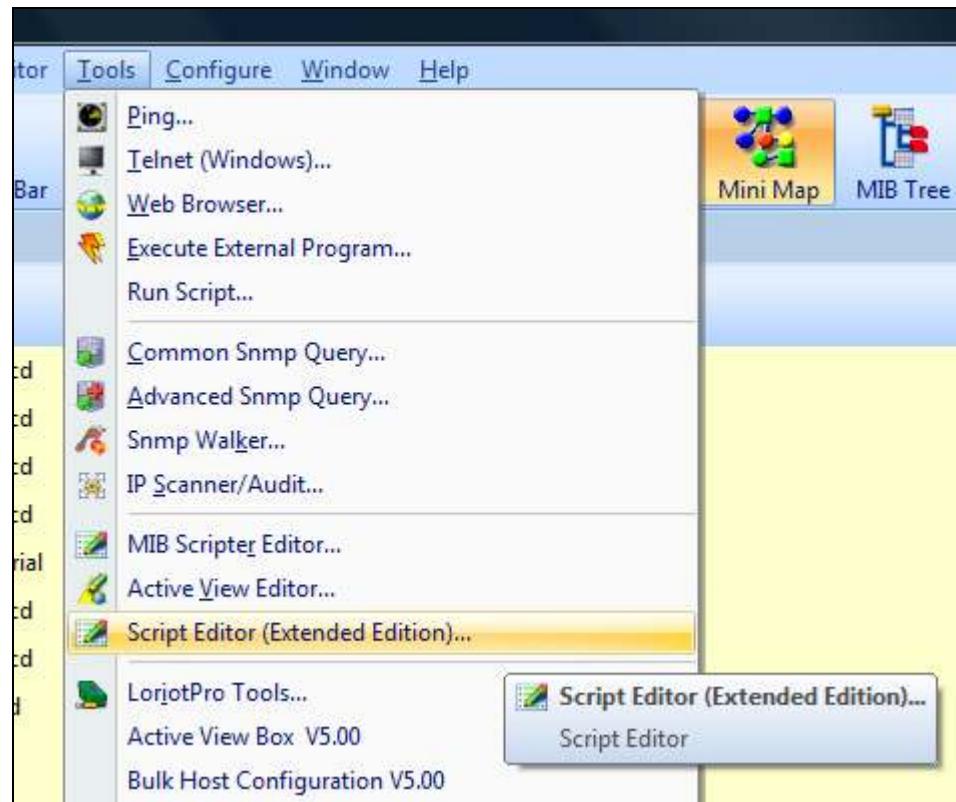
Soit ipaudit100.lua

```
-- this file is located to bin/config/script/audit
-- the name of the file was ipauditxx.lua where xx is the audit ref number
-- V1.1
----- title area
-- [[
title "Audit the storage unit of the host" -- title for this audit file
ref 100      -- the reference number for this audit 1 to 100000 was reserved to LUTEUS
type 2       -- to define the type of audit
comment "this script audit the storage capacity of one host" -- somme comment (just one line)
param "my specific parameter for this audit file"   -- somme param (just one line)
]]--
```

### 10.5.3 Wizard de création d'un fichier de scripts d'audits

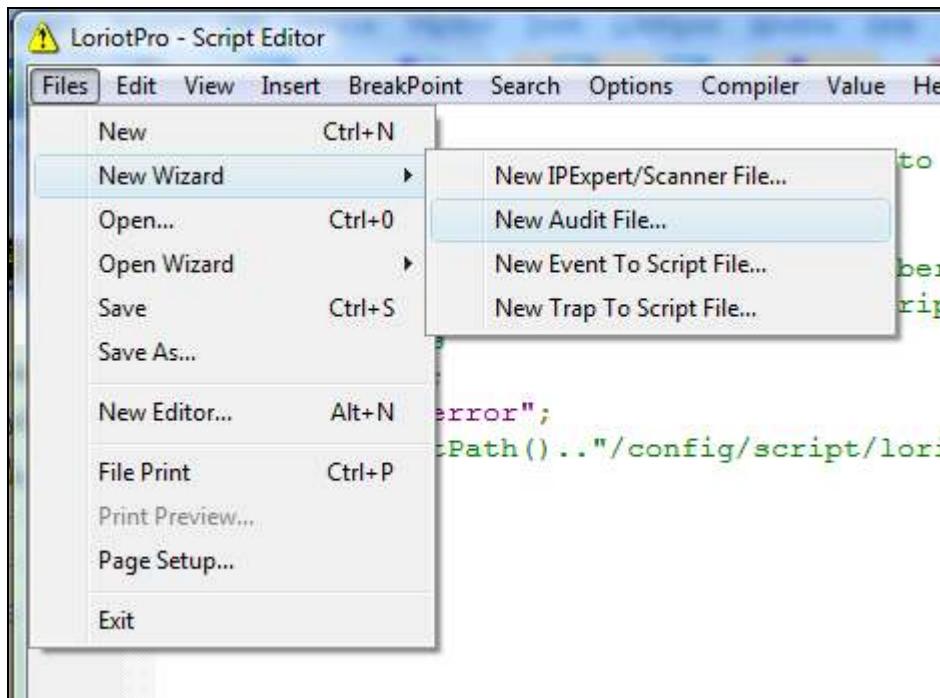
L'environnement de développement vous propose un Wizard de génération de scripts d'audits.

Le lancement de l'éditeur se fait à partir du menu principal du logiciel :

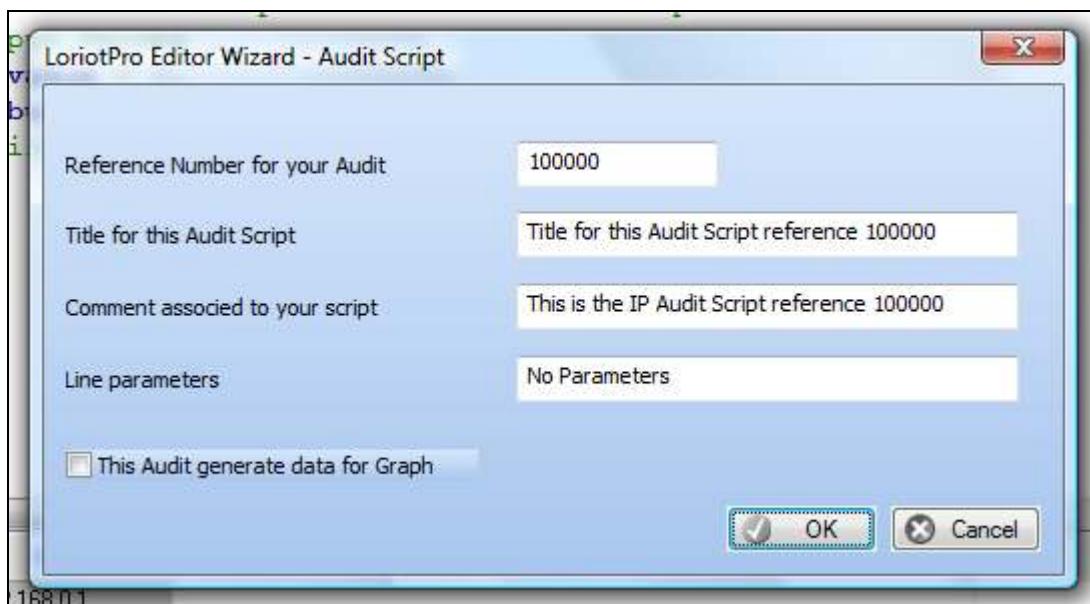


Il faut choisir l'option :

**Files/New Wizard/New Audit File...**



Une boîte de dialogue vous demande quelques paramètres.



#### **10.5.3.1 Options - Reference Number for your audit**

Un nombre de référence supérieur à 99999.

##### **Attention**

Il est préférable de conserver le numéro de référence proposé par le Wizard, il est logiquement libre.

#### **10.5.3.2 Options - Title for this Audit Script**

Un titre pour votre script.

#### **10.5.3.3 Options - Comment associed to your script**

Le commentaire pour votre script.

#### **10.5.3.4 Options - Line parameters**

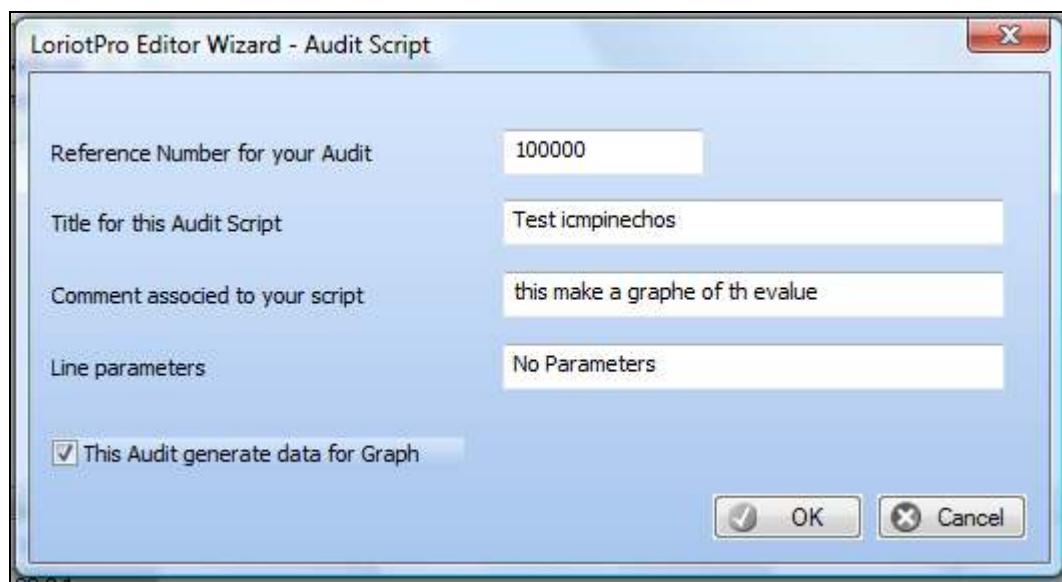
Un commentaire à destination de l'administrateur pour qu'il puisse renseigner ce champ.

#### **10.5.3.5 Options - This audit generate data for graph**

Le wizard va intégrer dans votre script les paramètres de gestion des datas de graphe collectés par le script.

#### **10.5.3.6 Exemple de script**

Dans cet exemple, nous allons faire un script d'audit simple qui collecte la valeur SNMP icmpinechos sur un équipement et l'utilise pour réaliser un graphe visible dans la Directory.



On clique « OK » pour valider la création du fichier de script.

The window title is 'LoriotPro - Script Editor : C:\PROGRA~1\LUCEUS\GORIOT~1\bin\config\script\ipaudit\ipaudit100000.lua'. The code editor displays the generated LUA script:

```
1 --- LoriotPro V5 (c) Luteus SARL
2 --- IP Expert/Scanner Script File
3 --- this file is located to bin/config/script/ipaudit
4 --- the name of the file was ipauditxx.lua where xx is the audit ref number
5 --- v1.00
6 ----- title area
7 ---[[
8   title "Test icmpinechos"
9   ref 100000
10  type 1
11  comment "this make a graphe of th evalue"
12  param "No Parameters"
13 ]][
14  -- Generated by the LoriotPro Editor Wizard at : Sun Jan 13 18:34:19 2008
15 -----
16 ----- audit script area
17
18
19
20  -- lp_oid content some standard parameters
21  -- lp_index content your parameter
22  -- lp_host default ip address for this script
23
24  -- lp_oid 'audit_ref_number event_ref event_level generate_audio generate_report dt-
25
26  -- (to test your script use this line)
```

Below the code editor is a table with two rows:

Value Name	Value in	Value out
lp_host	192.168.0.1	
lp_index		

Le script est généré avec les paramètres entrés dans le wizard.

Il n'est pas ici question d'expliquer le fonctionnement de la syntaxe du langage LUA mais uniquement la structure d'un fichier de script d'audit et son rapport avec la Directory.

-----  
-- LoriotPro V5 (c) Luteus SARL

## LoriotPro V5.00 Extended Edition – Supervision et gestion SNMP par scripts LUA

### Introduction au System Expert et aux scripts

```
-- IP Expert/Scanner Script File
-- this file is located to bin/config/script/ipaudit
-- the name of the file was ipauditxx.lua where xx is the audit ref number
-- V1.00
----- title area
-- [[
title "Test icmpinechos"
ref 100000
type 1
comment "this make a graphe of th evalue"
param "No Parameters"
]]--
-- Generated by the LoriotPro Editor Wizard at : Sun Jan 13 18:34:19 2008
-----
----- audit script area

-- lp_oid content some standard parameters
-- lp_index content your parameter
-- lp_host default ip address for this script

-- lp_oid 'audit_ref_number  event_ref  event_level  generate_audio  generate_report  dt->UID
graph_value'

-- (to test your script use this line)
-- lp_oid='100000 602 2 1 1 6666666666666666 1';
-- standard parameter parsing

if lp_oid==nil then
    lp_value=4;
    lp_buffer='lp_oid parameter error';
    return;
end

nb=lp.Parse(lp_oid,'array',' ');
if (nb~=7) then
    lp_value=4;
    lp_buffer='parsing of standard parametre lp_oid error';
    return;
end

audit_ref_number      =lp.atof(array[0]);
event_ref             =lp.atof(array[1]);
event_level           =lp.atof(array[2]);
generate_audio        =lp.atof(array[3]);
generate_report       =lp.atof(array[4]);
UID                  =array[5];
generate_graph        =lp.atof(array[6]);

-- standard Graph parameter parsing

lp.Print('Graph option for : ',lp_host,'\n');
-- lp_graph (1 = data collecte for graph 0 no collecte)
lp_graph=1; -- the audit script generate value for the graph
-- lp_graph_type -- data type : 0 counter 1 gauge
lp_graph_type=0; -- here it is counter
-- lp_graph_value -- the return value for the audit module to generate the graph
-- only one linear graph is possible
lp_graph_value=0; -- the script populate thi svalue with the result
-- the graph title
lp_graph_title=string.format('%s : graph audit',lp_host);
-- the graph unit
lp_graph_unit='octets'; -- here octet (it is a sample put your unit)
-----
-- Start of your program -----
-----
-- Use this line of code if you want use some defined LoriotPro LUA Define
-- dofile(lp.GetPath().."/config/script/loriotinit.lua");
-- Sample code
-- collecte data
-- [[
result=lp.Gets(lp_host,'icmpinechos','array');
if (result==1) then
lp.Print('Ok\n');

    if (lp.atof(array['icmpinechos.0'])<1000) then
```

```
-- initialise the data for the graph (if audit return data for graph)
-- lp_graph_value=lp.atof(array['icmpinechos.0']);
    lp_value = 2;
    lp_buffer ='ok'
else
    lp_value = 4;
    lp_buffer ='icmpinechos sup than 1000'
-- initialise the data for the graph (if audit return data for graph)
-- lp_graph_value=-1;
end
else
    lp_value = 4;
    lp_buffer ='host not respond';
    lp.Print('host not respond
');
end
]]-
```

Par défaut, le wizard vous propose une structure de script qui collecte la valeur **icmpinechos** dans la partie « Start of your program »

L'entête est générée et vous pouvez modifier les titres ou commentaires.

```
-----
-- LoriotPro V5 (c) Luteus SARL
-- IP Expert/Scanner Script File
-- this file is located to bin/config/script/ipaudit
-- the name of the file was ipauditxx.lua where xx is the audit ref number
-- V1.00
----- title area
--[[[title "Test icmpinechos" --
ref 100000 --
type 1 --
comment "this make a graphe of th evalue" --
param "No Parameters" --
]]--
-- Generated by the LoriotPro Editor Wizard at : Sun Jan 13 18:34:19 2008
-----
```

Une zone de variable vous permettant de simuler le script dans l'environnement de développement, est générée, il vous suffit de supprimer les deux – pour activer la ligne de code.

```
-- lp_oid='100000 602 2 1 1 6666666666666666 1';
```

```
----- audit script area

-- lp_oid content some standard parameters
-- lp_index content your parameter
-- lp_host default ip address for this script

-- lp_oid 'audit_ref_number event_ref event_level generate_audio generate_report dt->UID
graph_value'

-- (to test your script use this line)
-- lp_oid='100000 602 2 1 1 6666666666666666 1';
-- standard parameter parsing
```

Par défaut, cette zone est en commmentaires, car logiquement elle n'est pas dans le fichier de script mais concerne des variables fournies par le module d'audit au script. L'adresse ip du host concerné par le script est dans la variable **lp\_host** et les

paramètres configurés par l'administrateur dans la variable **Ip\_index**. La variable **lp\_oid** est gérée dynamique par le module d'audit.

```
-- standard parameter parsing

if lp_oid==nil then
    lp_value=4;
    lp_buffer='lp_oid parameter error';
    return;
end

nb=lp.Parse(lp_oid,'array',' ');
if (nb~=7) then
    lp_value=4;
    lp_buffer='parsing of standard parametre lp_oid error';
    return;
end

audit_ref_number      =lp.atof(array[0]);
event_ref              =lp.atof(array[1]);
event_level             =lp.atof(array[2]);
generate_audio         =lp.atof(array[3]);
generate_report        =lp.atof(array[4]);
UID                   =array[5];
generate_graph         =lp.atof(array[6]);
```

Cette zone correspond à l'analyse réalisée par le script pour récupérer les paramètres passés dans la chaîne de caractères **lp\_oid** par le module d'audit. Après l'analyse de la variable **lp\_oid** par le script, on retrouve les paramètres associés à votre audit.

Paramètres	commentaires
audit_ref_number	Le numéro de référence du fichier d'audit
event_ref	Le numéro d'événement à utiliser avec cet équipement pour l'audit
event_level	Le level à utiliser avec cet équipement pour l'audit
generate_audio	L'utilisateur désire la génération d'un message audio si l'audit le permet.
generate_report	L'utilisateur désire la génération d'un rapport si l'audit le permet.
UID	L'UID de référence dans la Directory de l'objet d'audit
generate_graph	réservé

```
-- standard Graph parameter parsing

lp.Print('Graph option for : ',lp_host,'\n');
-- lp_graph  (1 = data collecte for graph 0 no collecte)
lp_graph=1; -- the audit script generate value for the graph
-- lp_graph_type -- data type : 0 counter 1 gauge
lp_graph_type=0; -- here it is counter
-- lp_graph_value -- the return value for the audit module to generate the graph
-- only one linear graph is possible
lp_graph_value=0; -- the script populate this value with the result
-- the graph title
lp_graph_title=string.format('%s : graph audit',lp_host);
-- the graph unit
lp_graph_unit='octets'; -- here octet (it is a sample put your unit)
```

Cette zone a été créée afin de vous faciliter l'écriture des variables qui permettent de passer des valeurs au module d'audit dans le but de faire apparaître un graphe dans les fenêtres volantes. Il faut modifier les titres et paramètres à la main en fonction de vos besoins.

## Exemple

Voici le fichier de base pour réaliser notre exemple , en rouge les parties activées.

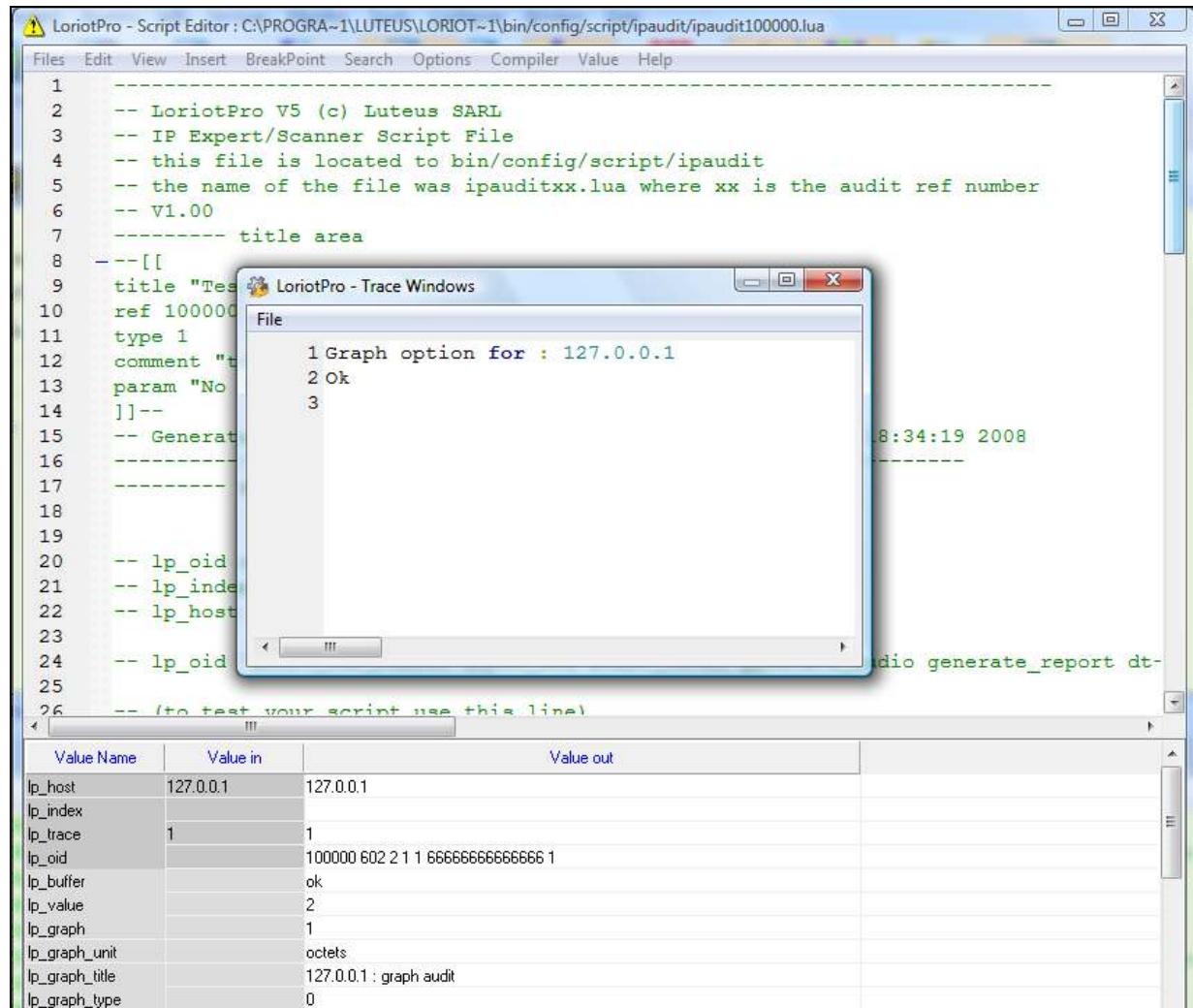
```
-----  
-- LoriotPro V5 (c) Luteus SARL  
-- IP Expert/Scanner Script File  
-- this file is located to bin/config/script/ipaudit  
-- the name of the file was ipauditxx.lua where xx is the audit ref number  
-- V1.00  
----- title area  
--[[  
title "Test icmpinechos"  
ref 100000  
type 1  
comment "this make a graphe of th evalue"  
param "No Parameters"  
]]--  
-- Generated by the LoriotPro Editor Wizard at : Sun Jan 13 18:34:19 2008  
-----  
----- audit script area  
  
-- lp_oid content some standard parameters  
-- lp_index content your parameter  
-- lp_host default ip address for this script  
  
-- lp_oid 'audit_ref_number' event_ref event_level generate_audio generate_report dt->UID  
graph_value,  
  
-- (to test your script use this line)  
lp_oid='100000 602 2 1 6666666666666666 1';  
-- standard parameter parsing  
  
if lp_oid==nil then  
    lp_value=4;  
    lp_buffer='lp_oid parameter error';  
    return;  
end  
  
nb=lp.Parse(lp_oid,'array',' ');  
if (nb~=7) then  
    lp_value=4;  
    lp_buffer='parsing of standard parametre lp_oid error';  
    return;  
end  
  
audit_ref_number      =lp.atof(array[0]);  
event_ref             =lp.atof(array[1]);  
event_level           =lp.atof(array[2]);  
generate_audio        =lp.atof(array[3]);  
generate_report       =lp.atof(array[4]);  
UID                  =array[5];  
generate_graph        =lp.atof(array[6]);  
  
-- standard Graph parameter parsing  
  
lp.Print('Graph option for : ',lp_host,'\n');  
-- lp_graph (1 = data collecte for graph 0 no collecte)  
lp_graph=1; -- the audit script generate value for the graph  
-- lp_graph_type -- data type : 0 counter 1 gauge  
lp_graph_type=0; -- here it is counter  
-- lp_graph_value -- the return value for the audit module to generate the graph  
-- only one linear graph is possible  
lp_graph_value=0; -- the script populate thi svalue with the result  
-- the graph title  
lp_graph_title=string.format('%s : graph audit',lp_host);  
-- the graph unit  
lp_graph_unit='octets'; -- here octet (it is a sample put your unit)  
-----  
-- Start of your program -----  
-----  
-- Use this line of code if you want use some defined LoriotPro LUA Define  
-- dofile(lp.GetPath().."/config/script/loriotinit.lua");
```

```
-- Sample code
-- collecte data

result=lp.Gets(lp_host,'icmpinechos','array');
if (result==1) then
lp.Print('Ok\n');

    if (lp.atof(array['icmpinechos.0'])<1000) then
-- initialise the data for the graph (if audit return data for graph)
    lp_graph_value=lp.atof(array['icmpinechos.0']);
    lp_value = 2;
    lp_buffer ='ok'
    else
    lp_value = 4;
    lp_buffer ='icmpinechos sup than 1000'
-- initialise the data for the graph (if audit return data for graph)
    lp_graph_value=-1;
end
else
    lp_value = 4;
    lp_buffer ='host not respond';
    lp.Print('host not respond');
end
```

On lance le script dans l'éditeur avec la touche « F5 ».



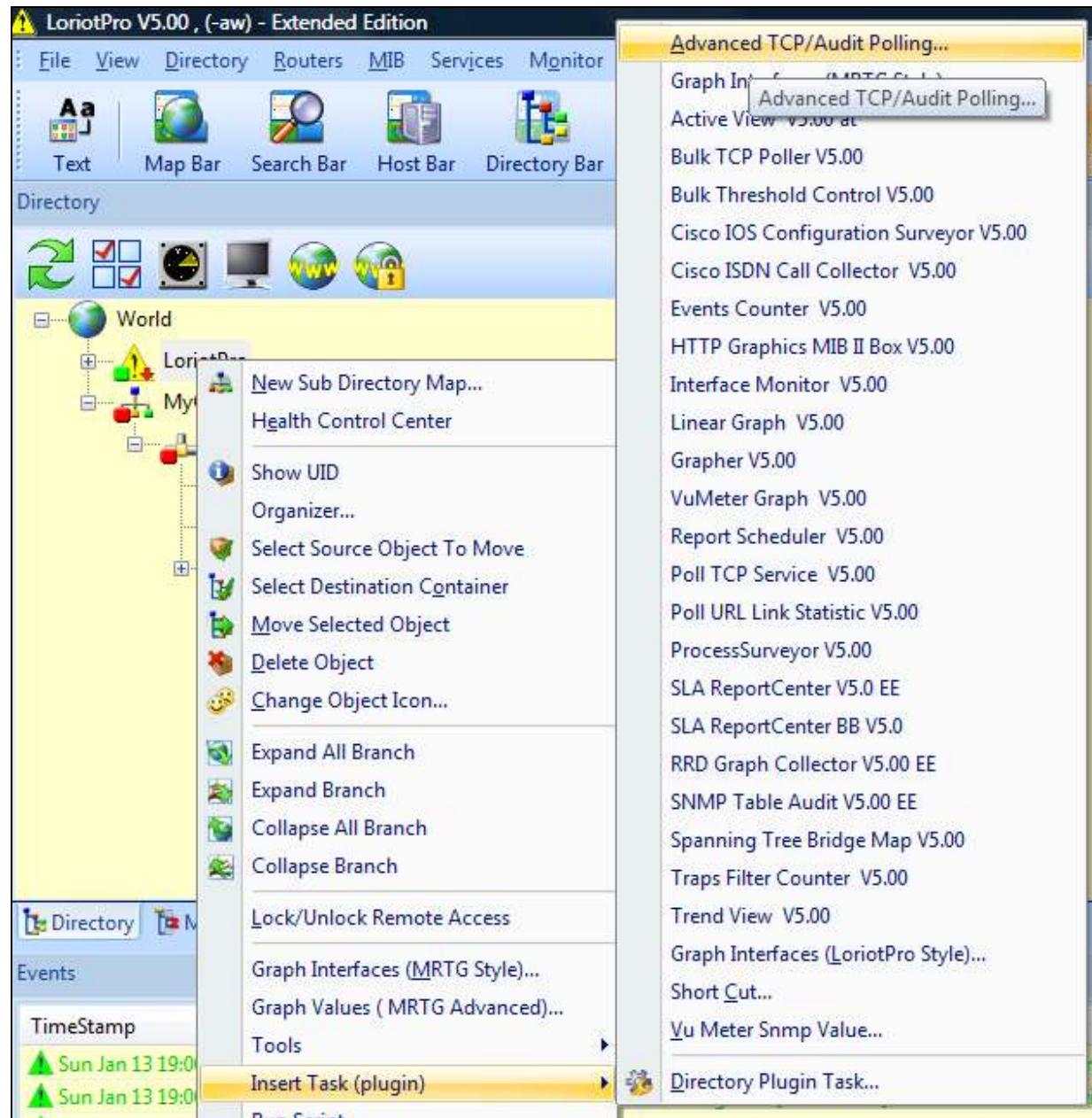
On supprime la ligne de test :

```
lp_oid='100000 602 2 1 1 6666666666666666 1';
```

```
-- lp_oid='100000 602 2 1 1 6666666666666666 1';
```

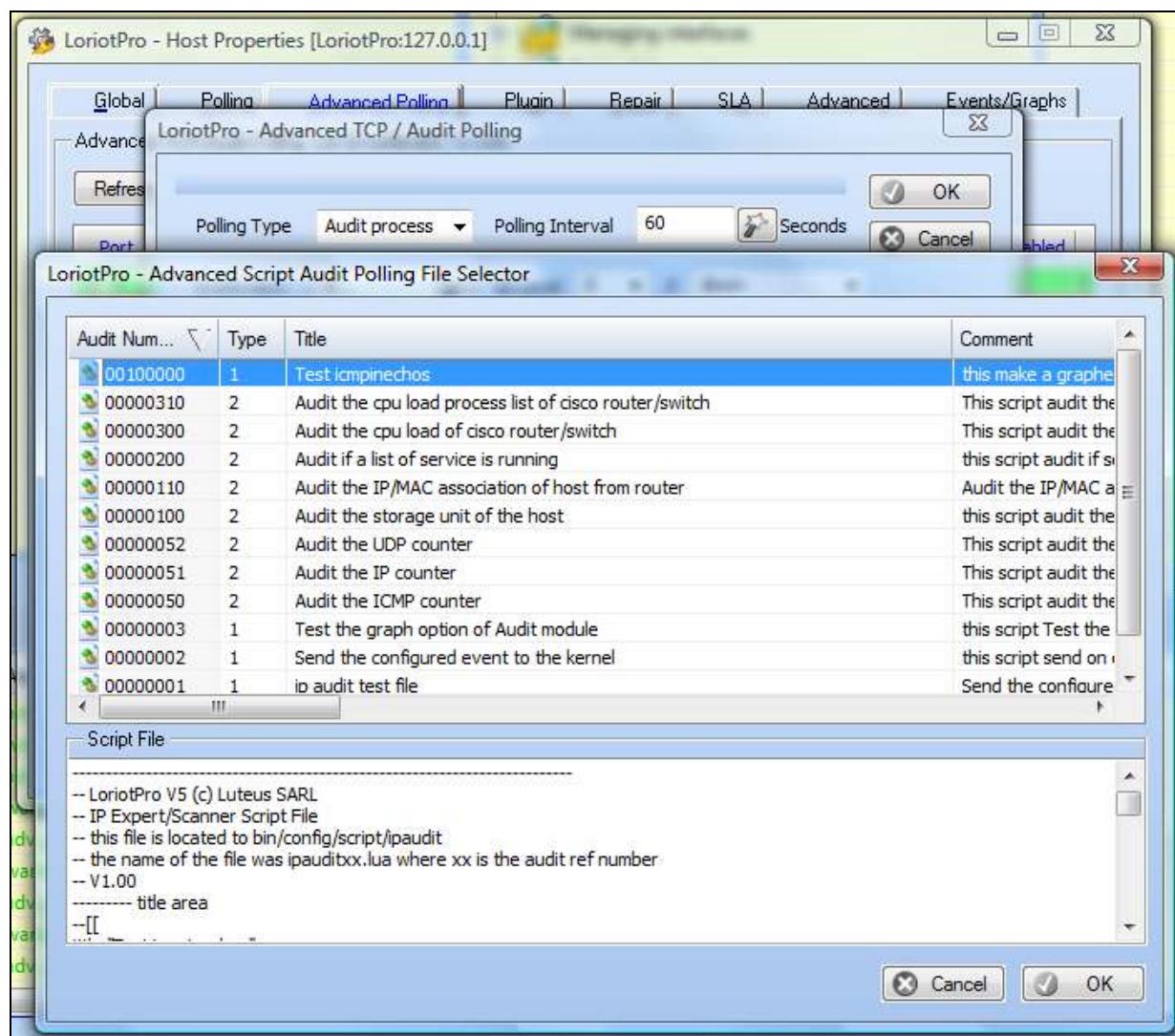
### On sauvegarde le script **CTRL^S**

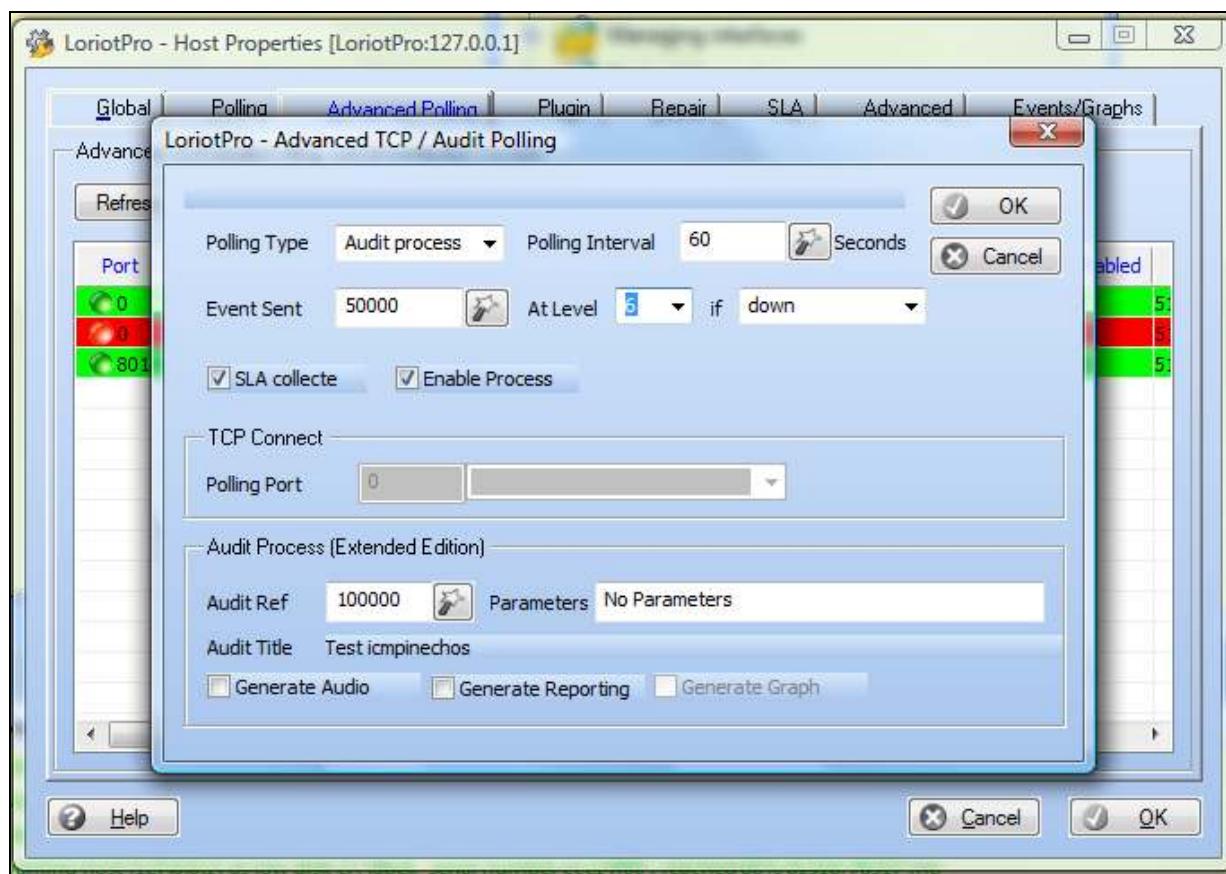
On se place sur un équipement dans la Directory et on insère notre script d'audit.



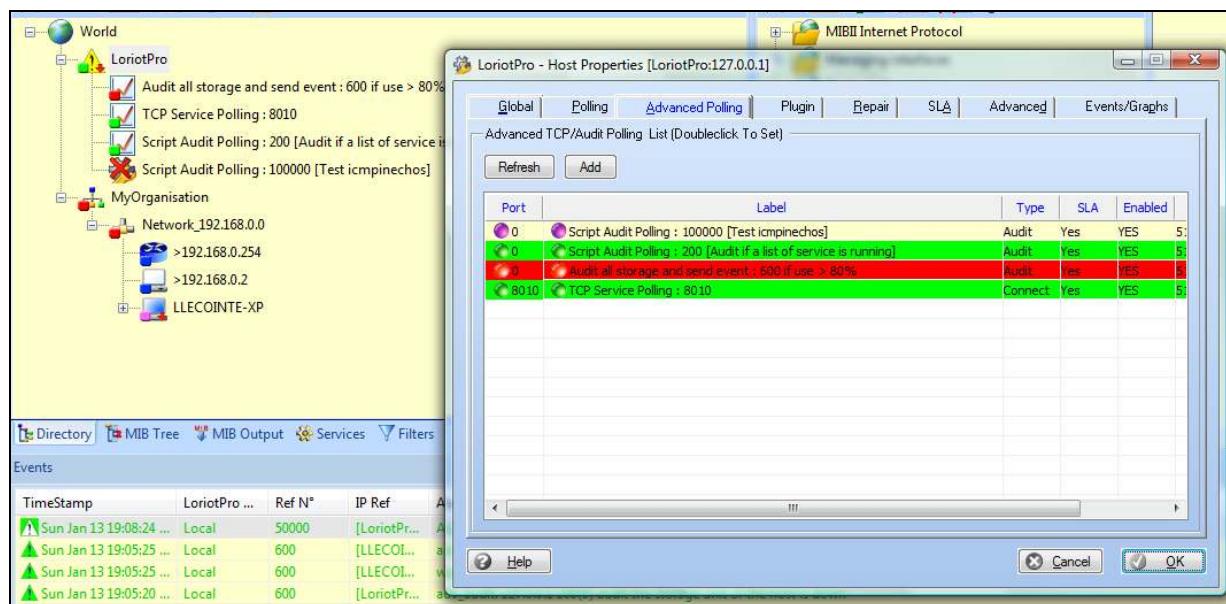
Notre script est visible dans la boîte de sélection.

LoriotPro V5.00 Extended Edition – Supervision et gestion SNMP par scripts LUA  
Introduction au System Expert et aux scripts

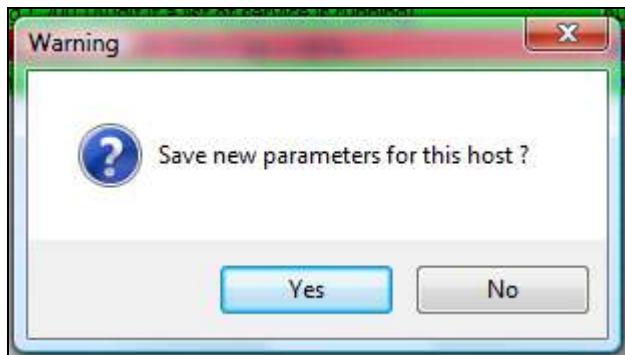




On n'oublie pas de cocher la case « Enable » pour activer l'exécution du script.



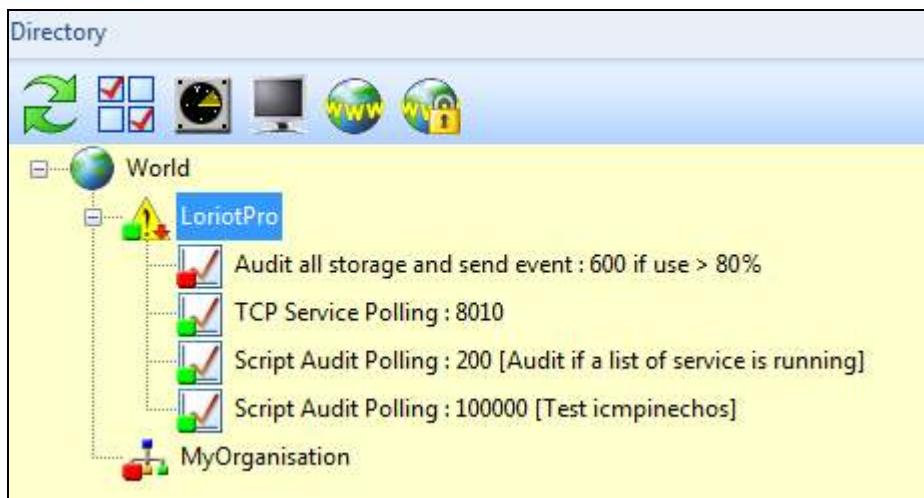
Le script d'audit est inséré dans la Directory sous l'équipement sélectionné. On peut remarquer que l'icône dans la Directory est une croix rouge car l'arbre n'est pas encore actualisé. L'icône sera correcte après un rafraîchissement de l'arbre de la Directory.



**Attention**

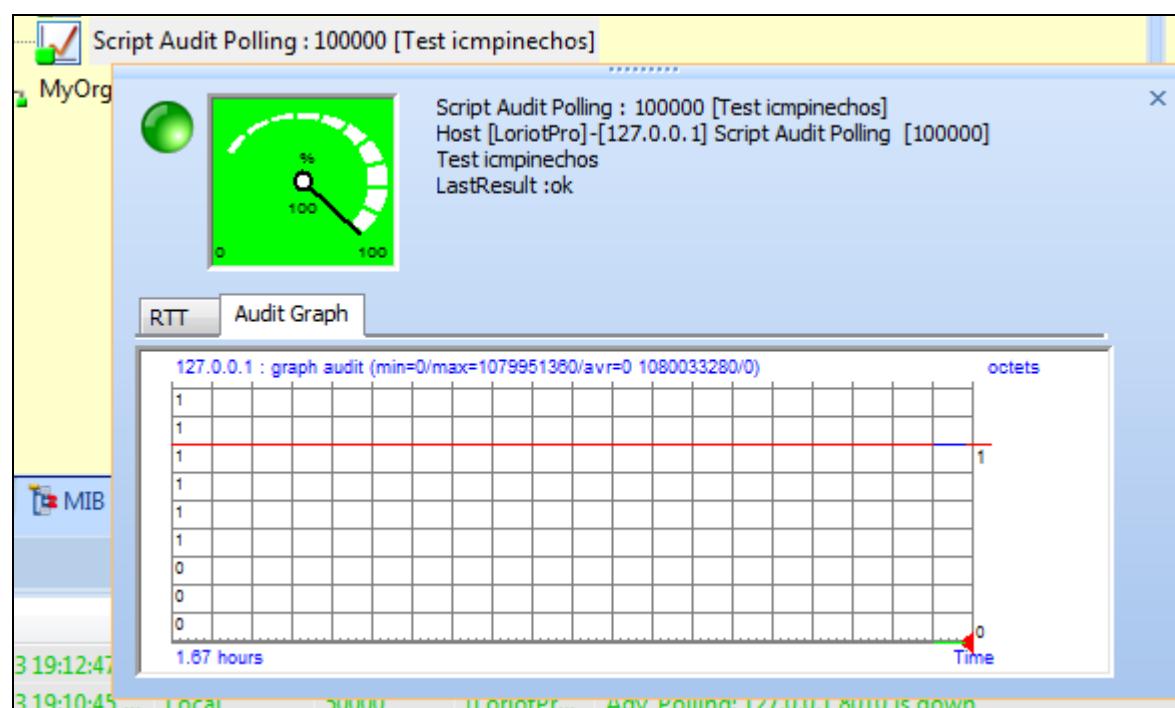
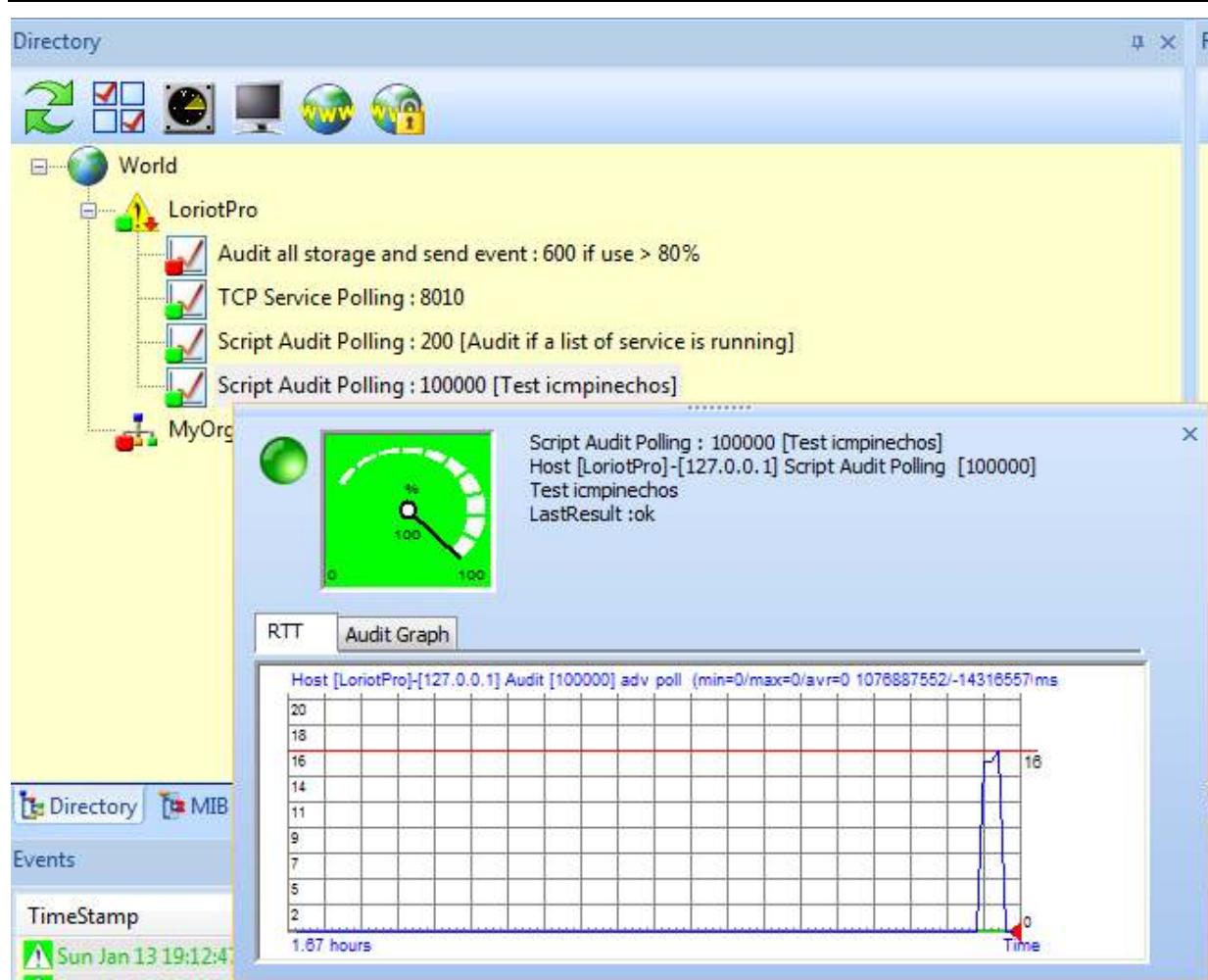
Une insertion d'audit sur un équipement via sa fenêtre de propriété, ne peut pas être annulée lorsque sa réalisation est terminée (il faut faire un « delete » de l'objet dans la Directory).

« Yes » ou « No » sort .



Avec la fenêtre volante, nous voyons apparaître l'onglet dédié au graphe.

LoriotPro V5.00 Extended Edition – Supervision et gestion SNMP par scripts LUA  
Introduction au System Expert et aux scripts



## 11 Objet SNMP virtuel

Pour simplifier l'usage des scripts dans l'ensemble des fonctions existantes de LoriotPro, il est possible d'associer un script LUA avec un objet SNMP. L'association entre un script LUA et un nom d'objet SNMP se réalise à l'aide d'un fichier de MIB standard comportant quelques balises propriétaires. Ce mécanisme permet de fournir de façon simple à l'utilisateur un ensemble de fonctions utilisables comme des objets SNMP standard.

De base, la MIB LUTEUS-SCRIPT-STANDARD-MIB incluse dans le package «Extended Edition» de LoriotPro (fichier bin/mibs/1-lp\_script-standard.mib) contient 40 objets virtuels à utiliser. La MIB LUTEUS-SCRIPT-MIB incluse dans le package «Extended Edition» de LoriotPro (fichier bin/mibs/1-lp\_script01.mib) contient des exemples d'objets virtuels créés à partir de scripts LUA.

Le principe d'utilisation de fichiers de MIB pour créer des objets virtuels SNMP est très simple. Le fichier est créé en utilisant la syntaxe ASN1 standard d'un fichier de MIB. Pour que l'objet soit considéré par LoriotPro comme un script LUA, il faut utiliser un « ACCESS » de type **lp\_access\_script**.

### 11.1.1 Passage de paramètres

Lorsque LoriotPro appelle un objet snmp virtuel, il lui passe plusieurs paramètres que le script va pouvoir utiliser pour s'initialiser. En retour, le script fournit deux valeurs que LoriotPro va utiliser pour construire la réponse.

LoriotPro fournit les paramètres suivants au script :

Paramètres	Descriptions
lp_host	L'adresse IP de l'équipement sous la forme d'une chaîne de caractères
lp_oid	Le nom de l'objet snmp virtuel sous la forme d'une chaîne de caractères
lp_index	L'index associé à l'objet snmp virtuel sous la forme d'une chaîne de caractères incluant le point. « .2 »

Le script fournit les paramètres suivants à LoriotPro :

Paramètres	Descriptions
lp_value	Une valeur (format double pour supporter le 64 bits) (si l'objet est de type « integer »)
lp_buffer	Une chaîne de caractères (si l'objet est de type string)

Exemple:

Le fichier **your-script.mib** contient la définition de l'objet **lp\_your\_diskused** qui va nous permettre de calculer le pourcentage d'utilisation de notre disque C :.

```
YOUR-SCRIPT-MIB DEFINITIONS ::= BEGIN

IMPORTS
    luteus                FROM LUTEUS-TC-MIB;
    scripts                FROM LUTEUS-TC-MIB;

your_script      OBJECT IDENTIFIER ::= { scripts 1000 }

-- Vous devez commencer à 1000 (1 to 999 is reserved to Luteus)
-- ou bien attacher vos objets à votre propre OID
-- si vous désirez diffuser vos MIB en utilisant notre OID
-- enterprises.luteus
-- consulter nous pour obtenir un range de valeur

-- le nom des objets est limité à une vingtaine de caractères

lp_your_diskused OBJECT-TYPE
    SYNTAX      Integer32
    ACCESS      lp_access_script
    STATUS      current
    DESCRIPTION
        "Test un calcul sur
lp_value      =          (lp.Get(lp_host,'hrstorageused.index') /
lp.Get(lp_host,'hrstoragesize.index'))*100;"
        "
-- le script peut être "embedded" directement dans le fichier
-- il est placé entre les balises <LP_SCRIPT>
-- il est aussi possible de ne pas utiliser ces balises
-- à la compilation un fichier lp_your_diskused.lua sera généré dans
-- le répertoire bin/config/script
-- il est possible de copier directement le fichier lp_your_diskused.lua
-- dans le répertoire bin/config/script.
-- une fois compilé il est possible de modifier le fichier
-- lp_your_diskused.lua se trouvant dans bin/config/script.
-- attention si vous recompiler cette MIB vos modifications seront perdu.

-- lp_host et lp_index est passé au script par LoriotPro
-- lp_iid n'est pas utilisé par ce script
-- lp_value est retourné par le script car l'objet est de type Integer32

<LP_SCRIPT>

lp_buffer ="BAD"

    get1=("hrstorageused"..lp_index); --concatenation
    get2=("hrstoragesize"..lp_index); --concatenation

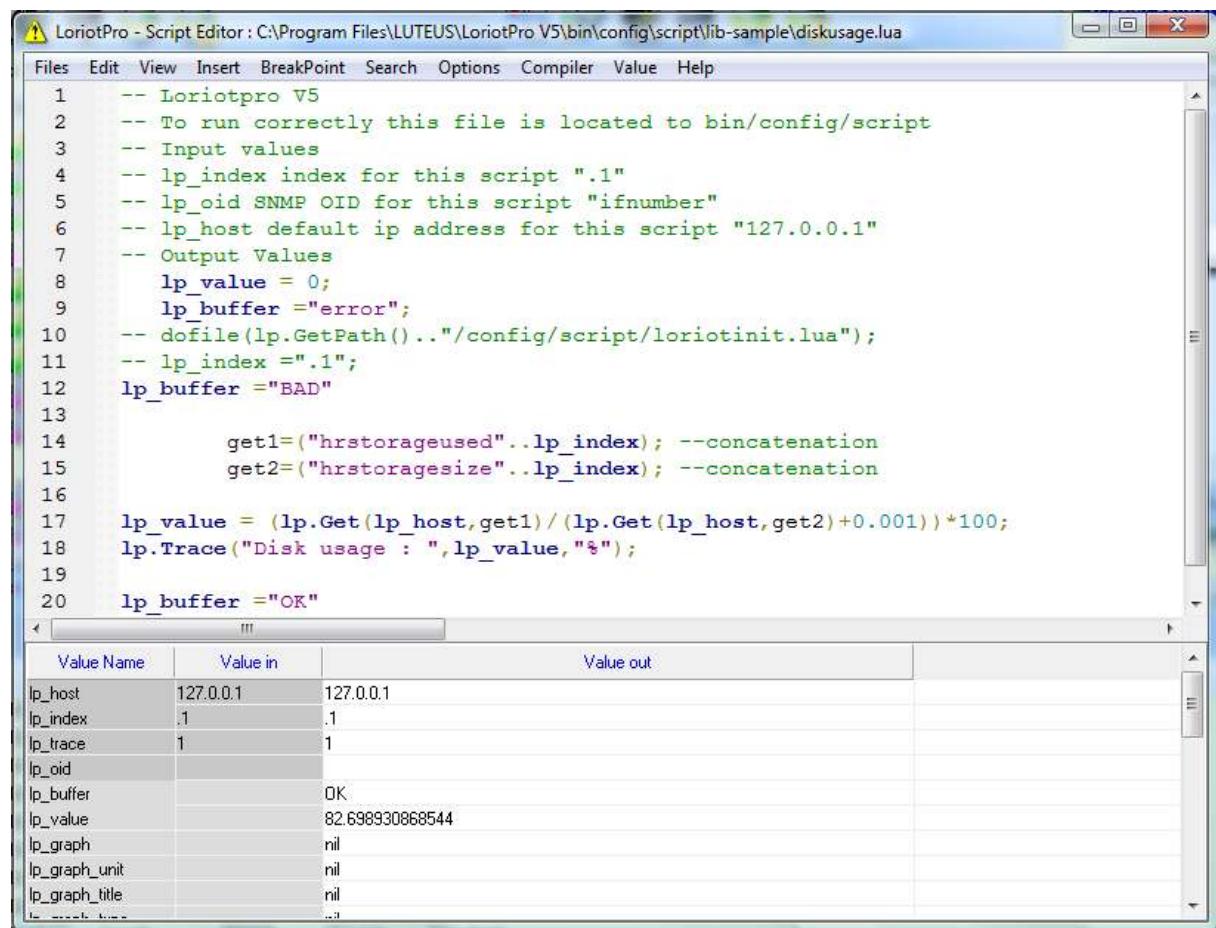
lp_value = (lp.Get(lp_host,get1)/(lp.Get(lp_host,get2)+0.001))*100;
lp_buffer ="OK"

<LP_SCRIPT>

::= { your_script 1 }

END
```

On peut tester le script au préalable avec l'éditeur de script. Reportez-vous au chapitre concernant l'éditeur pour plus d'information sur son utilisation.



The screenshot shows the LoriotPro Script Editor window. The title bar reads "LoriotPro - Script Editor : C:\Program Files\LUTEUS\LoriotPro V5\bin\config\script\lib-sample\diskusage.lua". The menu bar includes Files, Edit, View, Insert, BreakPoint, Search, Options, Compiler, Value, and Help. The main area contains the following Lua code:

```
1 -- Loriotpro v5
2 -- To run correctly this file is located to bin/config/script
3 -- Input values
4 -- lp_index index for this script ".1"
5 -- lp_oid SNMP OID for this script "ifnumber"
6 -- lp_host default ip address for this script "127.0.0.1"
7 -- Output Values
8     lp_value = 0;
9     lp_buffer ="error";
10    -- dofile(lp.GetPath() .. "/config/script/loriotinit.lua");
11    lp_index =".1";
12    lp_buffer ="BAD"
13
14        get1= ("hrstorageused"..lp_index); --concatenation
15        get2= ("hrstoragesize"..lp_index); --concatenation
16
17    lp_value = (lp.Get(lp_host,get1)/(lp.Get(lp_host,get2)+0.001))*100;
18    lp.Trace("Disk usage : ",lp_value,"%");
19
20    lp_buffer ="OK"
```

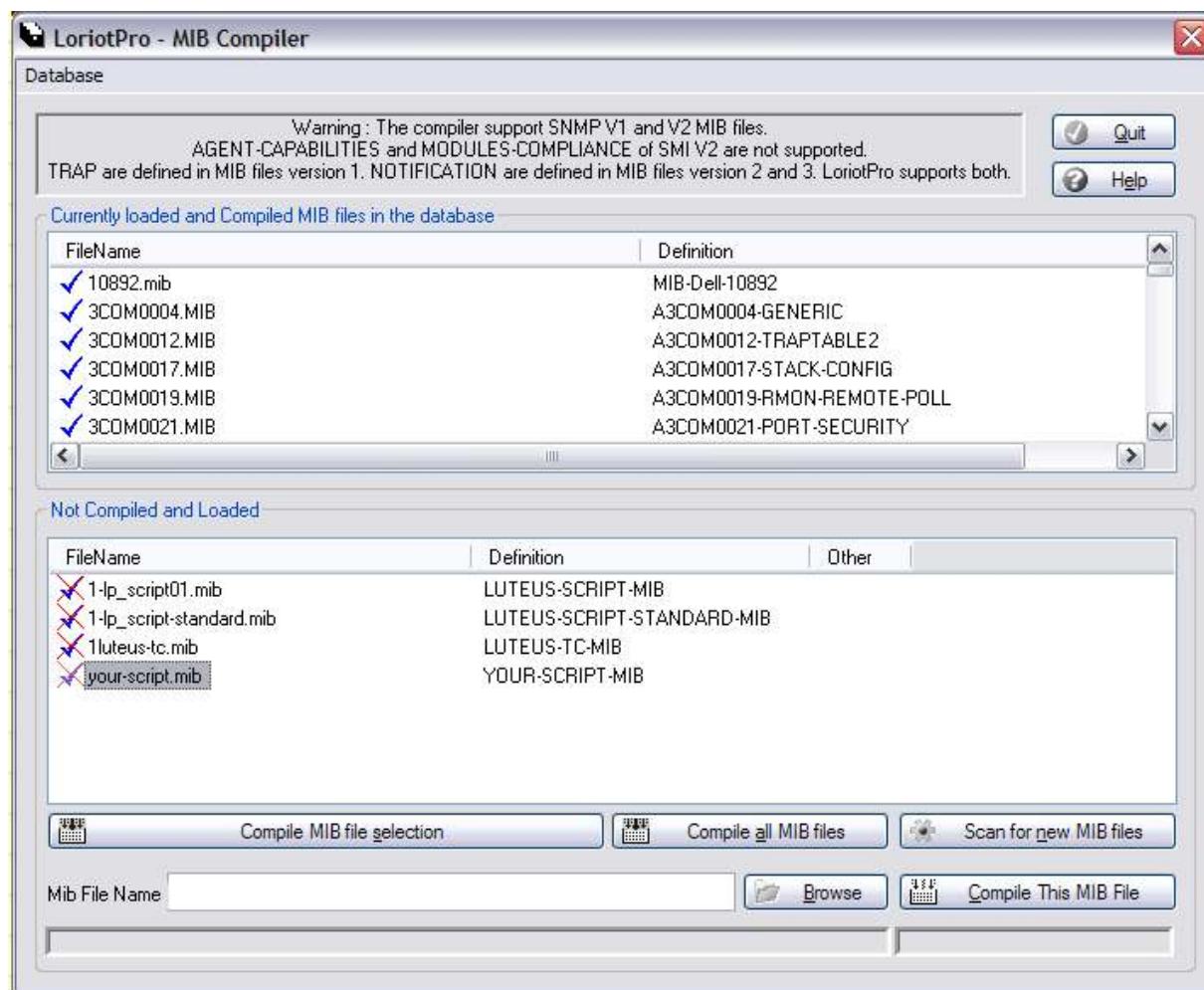
Below the code is a table titled "Value" with columns "Value Name", "Value in", and "Value out". The table contains the following data:

Value Name	Value in	Value out
lp_host	127.0.0.1	127.0.0.1
lp_index	.1	.1
lp_trace	1	1
lp_oid		
lp_buffer		OK
lp_value		82.698930868544
lp_graph		nil
lp_graph_unit		nil
lp_graph_title		nil
lp_detailed_usage		..1

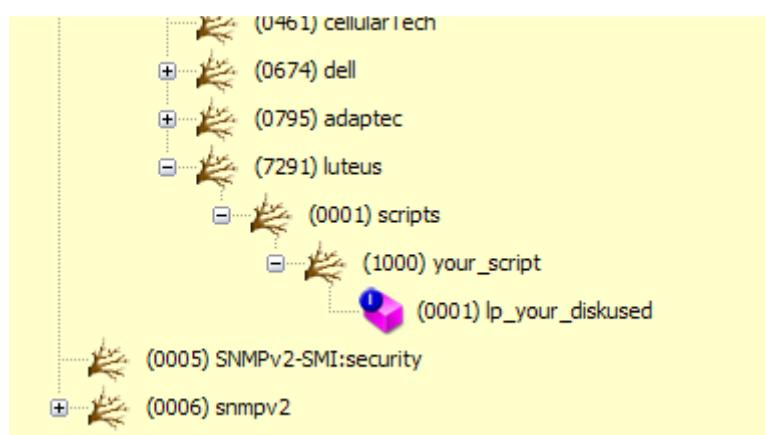
La touche F5 permet de lancer le script

On place le fichier **your-script.mib** dans le répertoire *bin/mibs* et on lance le compilateur de MIB.

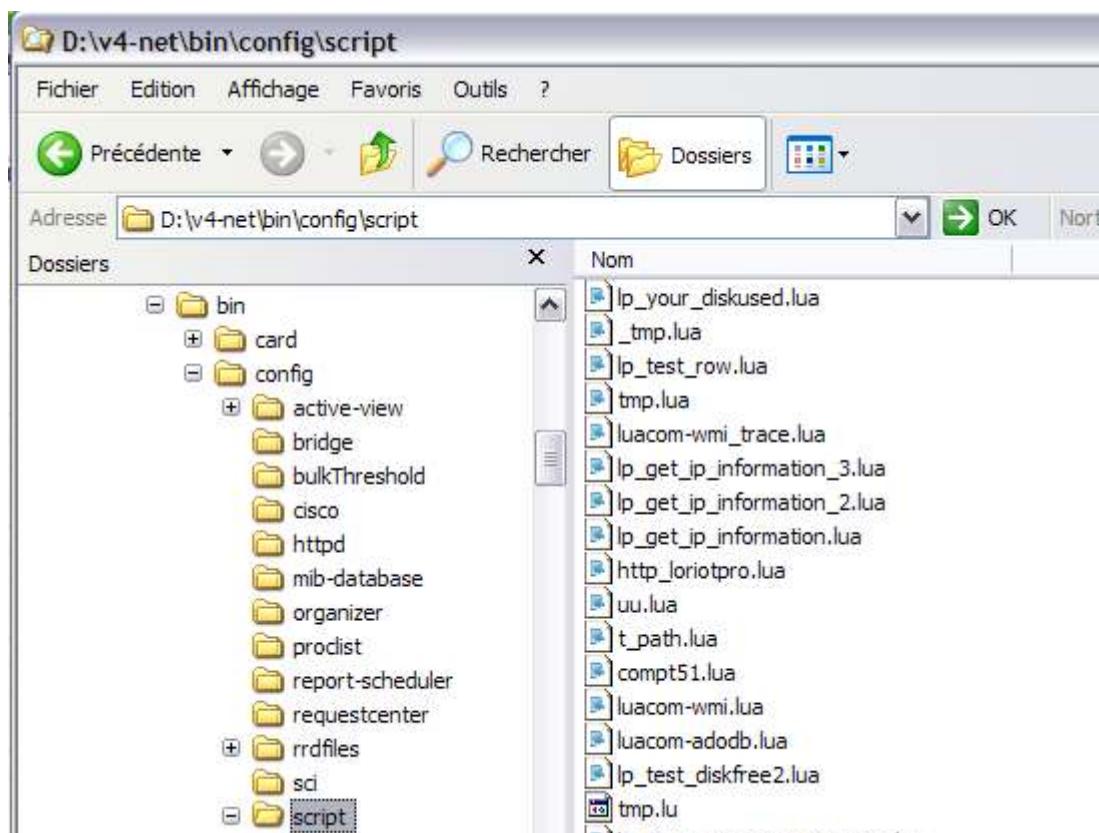
## LoriotPro V5.00 Extended Edition – Supervision et gestion SNMP par scripts LUA Introduction au System Expert et aux scripts



Après la compilation, votre script est vu comme un objet SNMP.



Le fichier « *lp\_your\_diskused.lua* » a été créé dans le répertoire *bin/config/script*.



The screenshot shows a 'Bloc-notes' window titled 'lp\_your\_diskused.lua'. The window contains the following Lua script:

```
Ip_buffer ="BAD"

get1= ("hrstorageused"..Ip_index); --concatenation
get2= ("hrstoragesize"..Ip_index); --concatenation

Ip_value = (Ip.Get(Ip_host,get1)/(Ip.Get(Ip_host,get2)+0.001))*100;
Ip_buffer ="OK"
```

L'objet peut maintenant être utilisé comme un objet SNMP standard à travers tous les modules de LoriotPro.

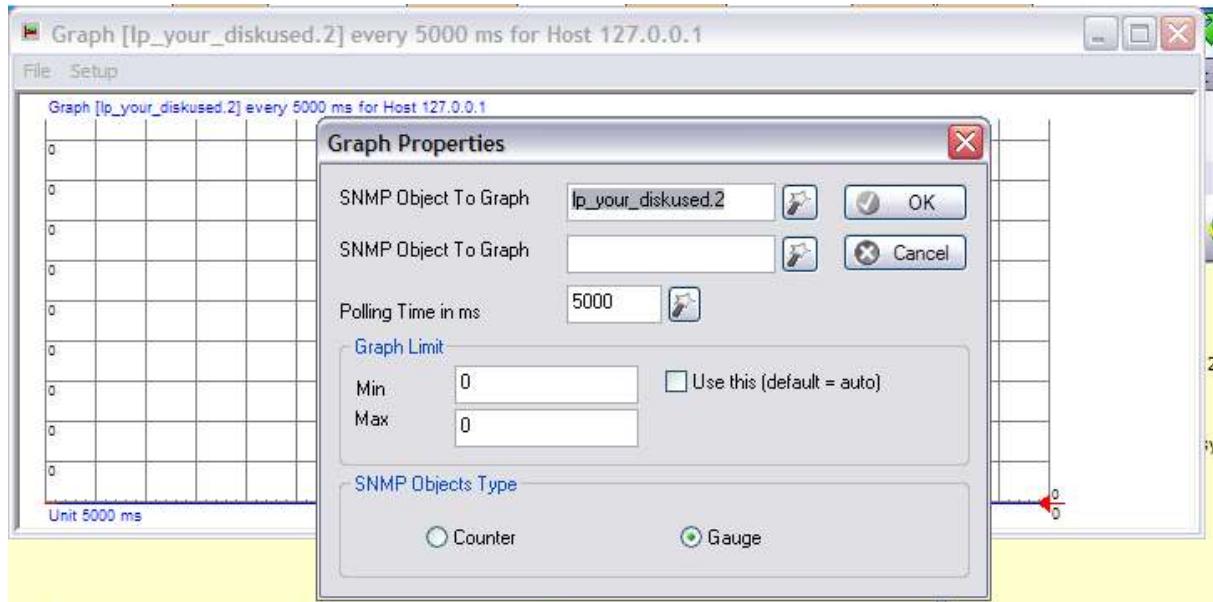
**Attention**

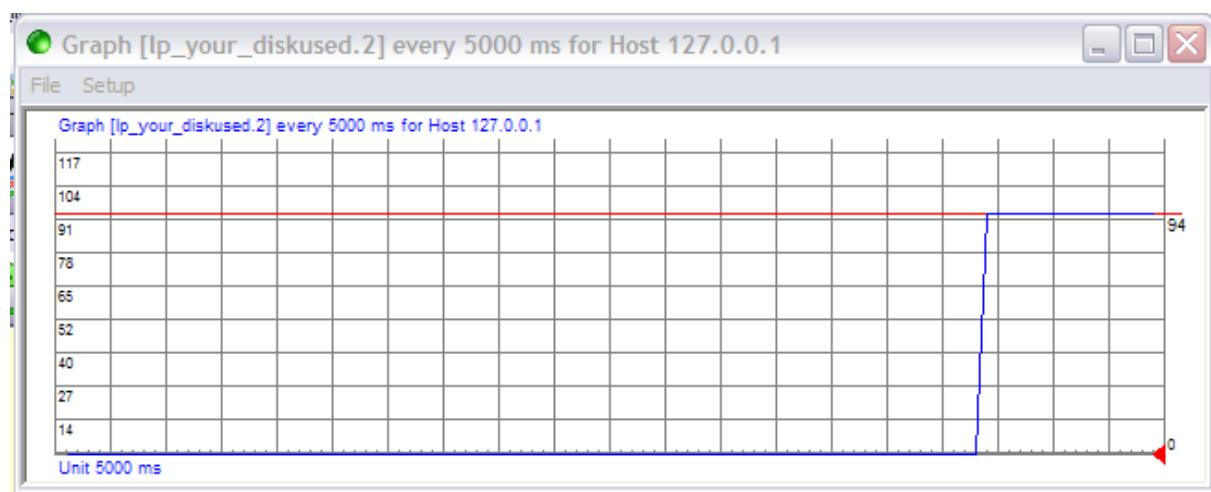
Cet objet est indexé et il faudra donc indiquer manuellement l'index du disque que nous désirons tester.



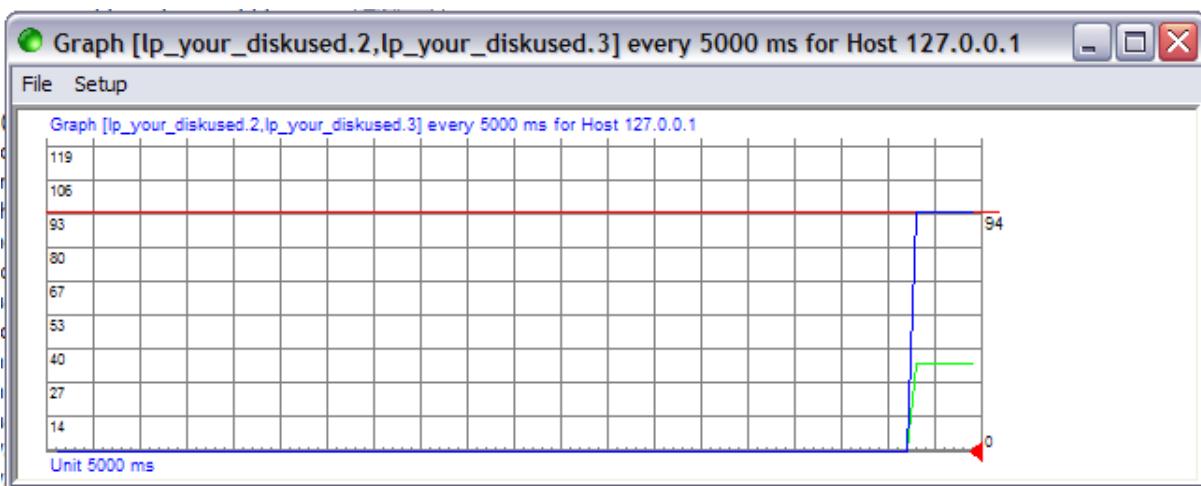
Dans ce module qui affiche le nom de l'objet SNMP reçu, nous avons **lp\_your\_diskused.2** qui n'est en réalité qu'un artefact ; par contre, nous avons bien les 94,02 % d'utilisation de notre disque C :.

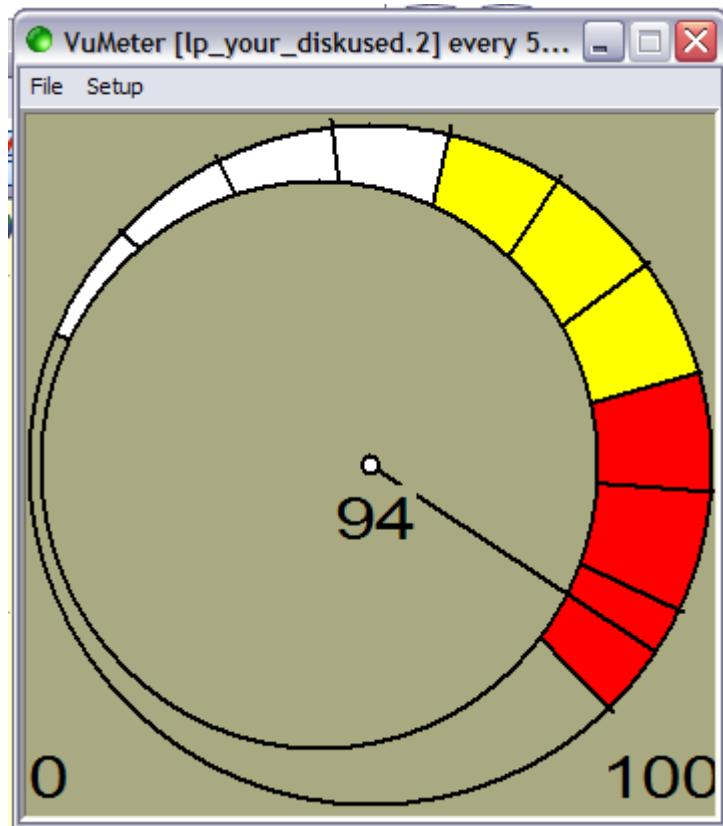
Nous pouvons l'utiliser dans un plugin de graphe linéaire.



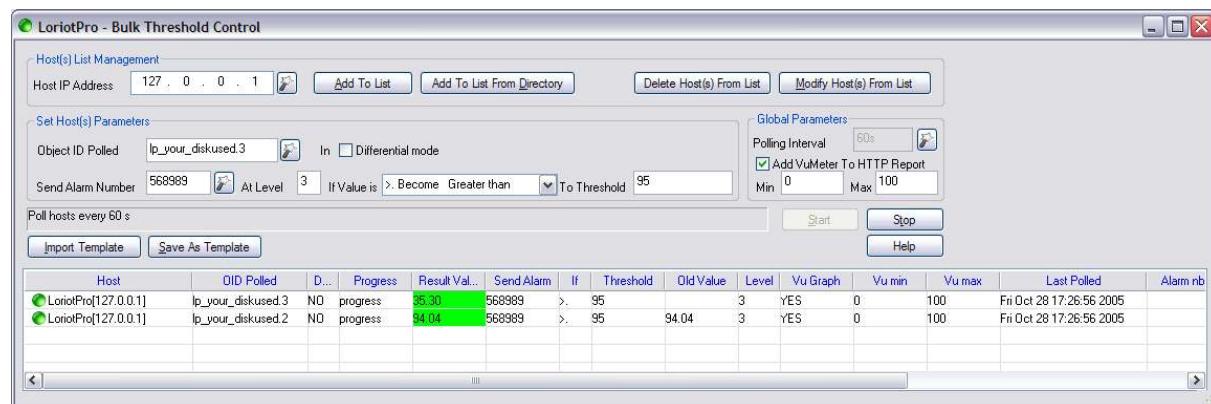


Des requêtes multiples peuvent être réalisées en changeant l'index.



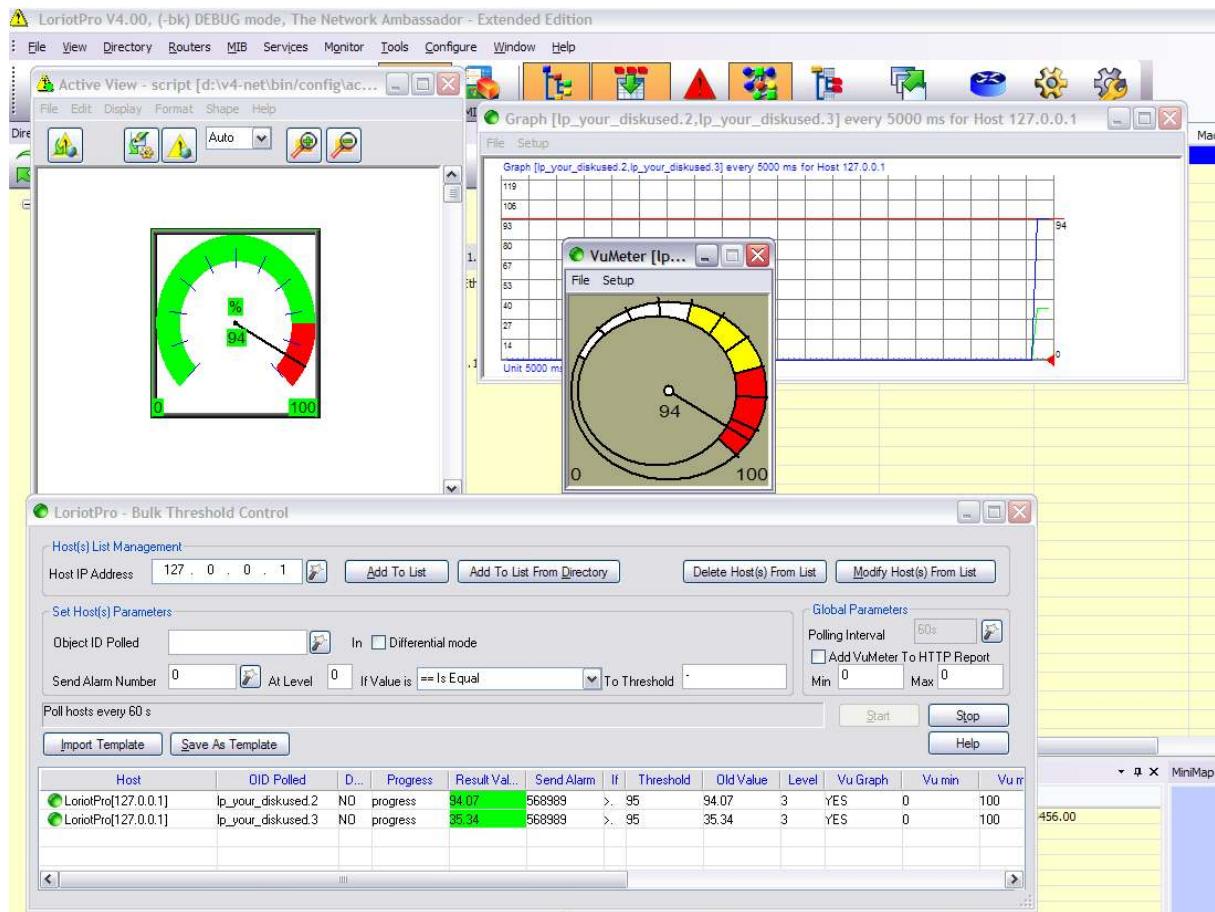


Utilisation avec le plugin « VuMeter ».



Utilisation avec le plugin « Bulk Threshold Control ».

## LoriotPro V5.00 Extended Edition – Supervision et gestion SNMP par scripts LUA Introduction au System Expert et aux scripts

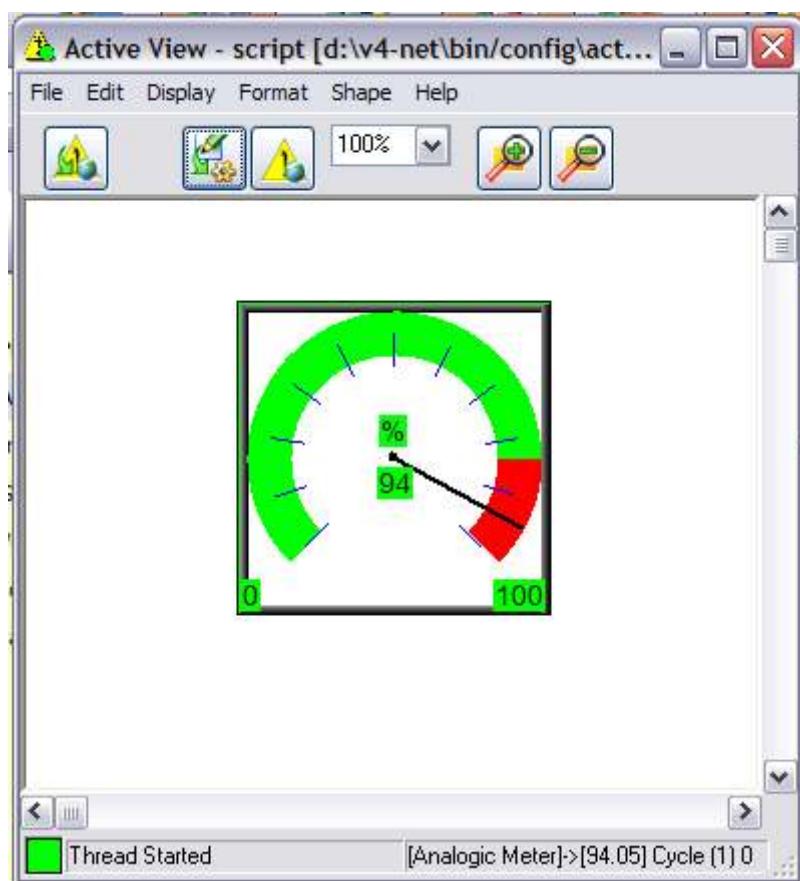
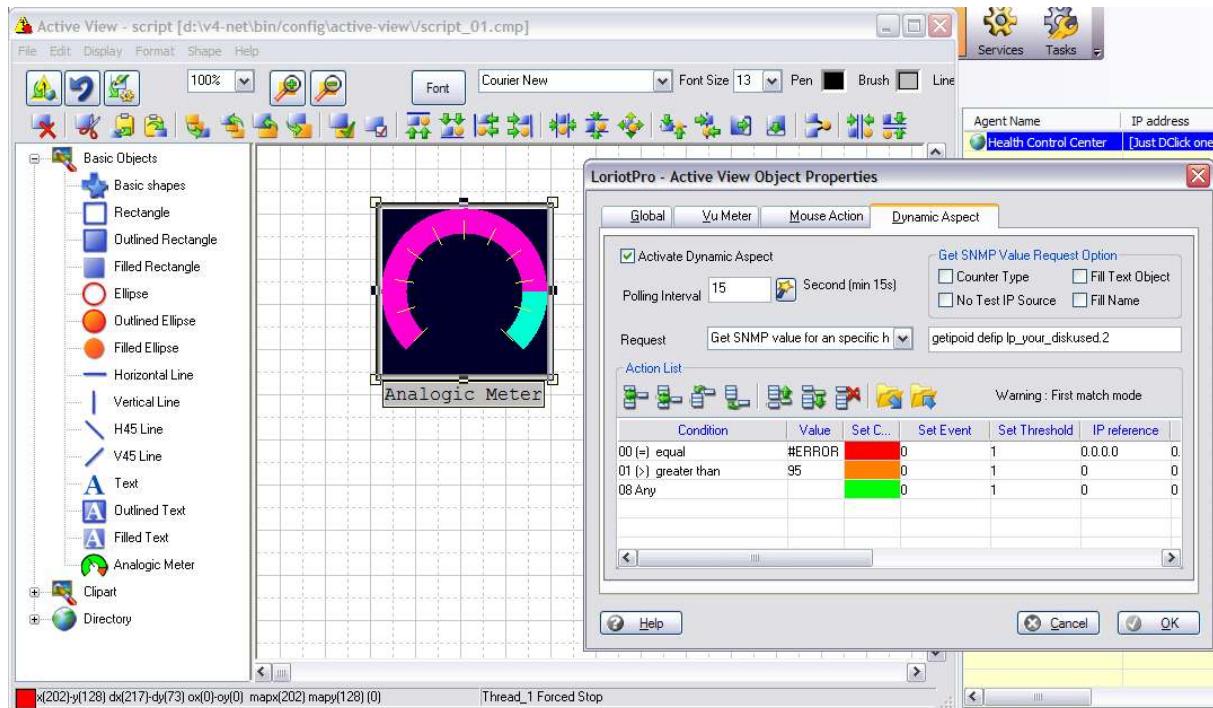


Le principe des objets SNMP virtuels est simple, une requête avec l'OID du ou des objets virtuels est envoyée vers l'équipement. Au retour du paquet (mode synchrone uniquement), le noyau de LoriotPro regarde les objets de type **lp\_access\_script** (logiquement à NUL) retournés par l'agent SNMP de l'équipement et les remplace par le résultat des scripts. Les scripts peuvent eux aussi envoyer des requêtes vers l'agent SNMP ou un ensemble d'agents SNMP.

**Attention de ne pas réaliser des boucles d'appelles d'objets SNMP virtuel.**

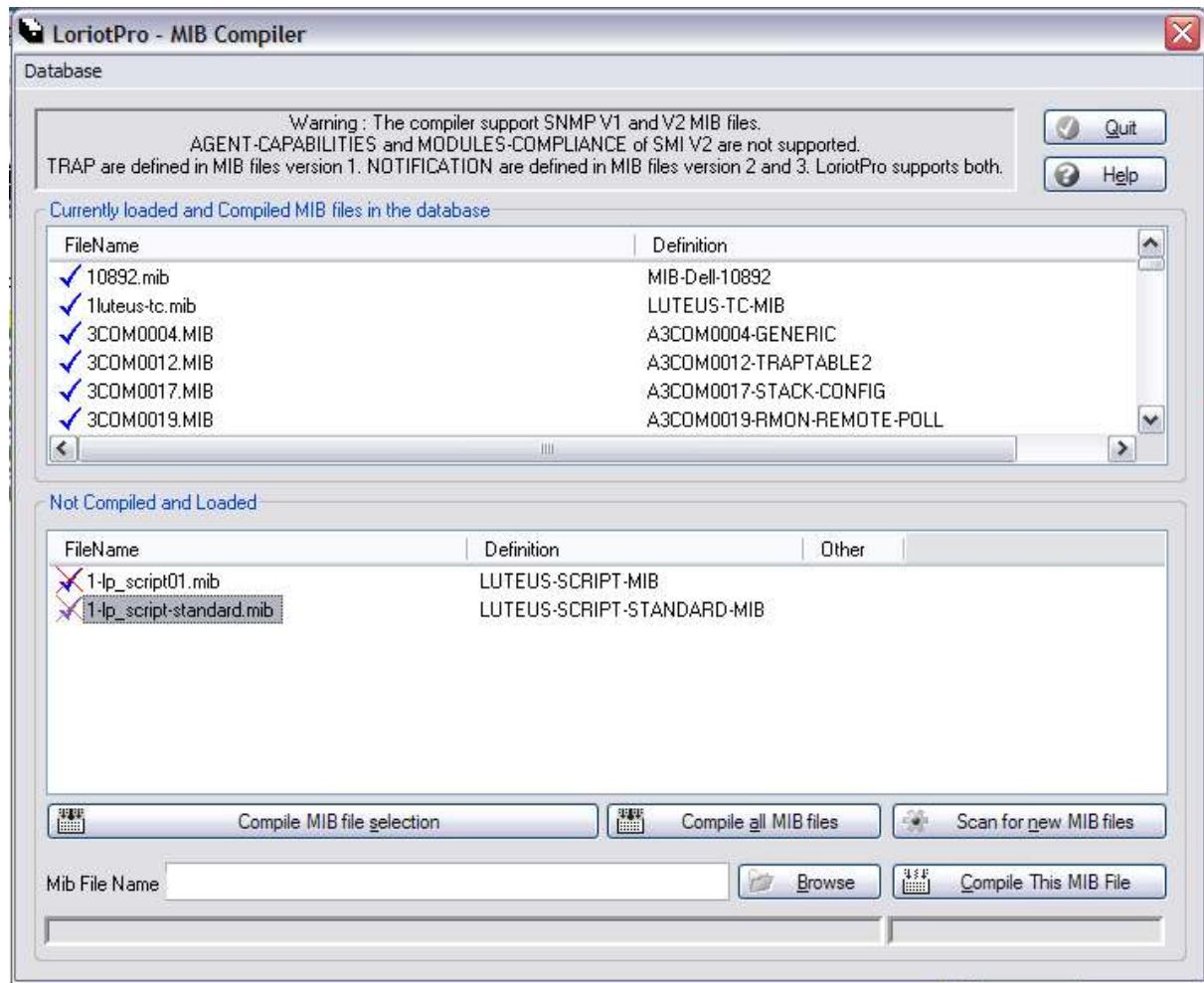
Les objets SNMP virtuels sont bien sûr utilisables dans des ActiveView.

## LoriotPro V5.00 Extended Edition – Supervision et gestion SNMP par scripts LUA Introduction au System Expert et aux scripts

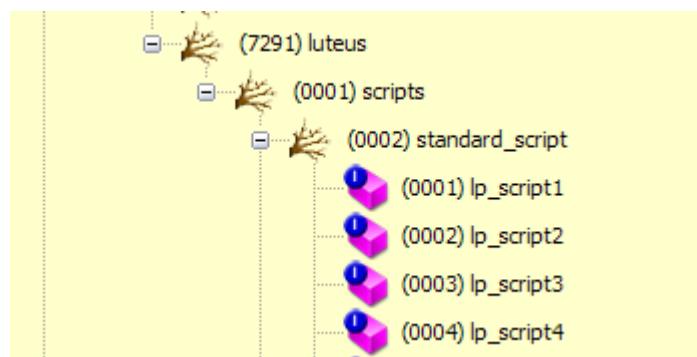


### 11.1.2 Objet snmp virtuel par défaut

Pour simplifier la réalisation d'objets snmp virtuels utilisant des scripts LUA, le logiciel inclut une quarantaine d'objets déjà prédéfinis.

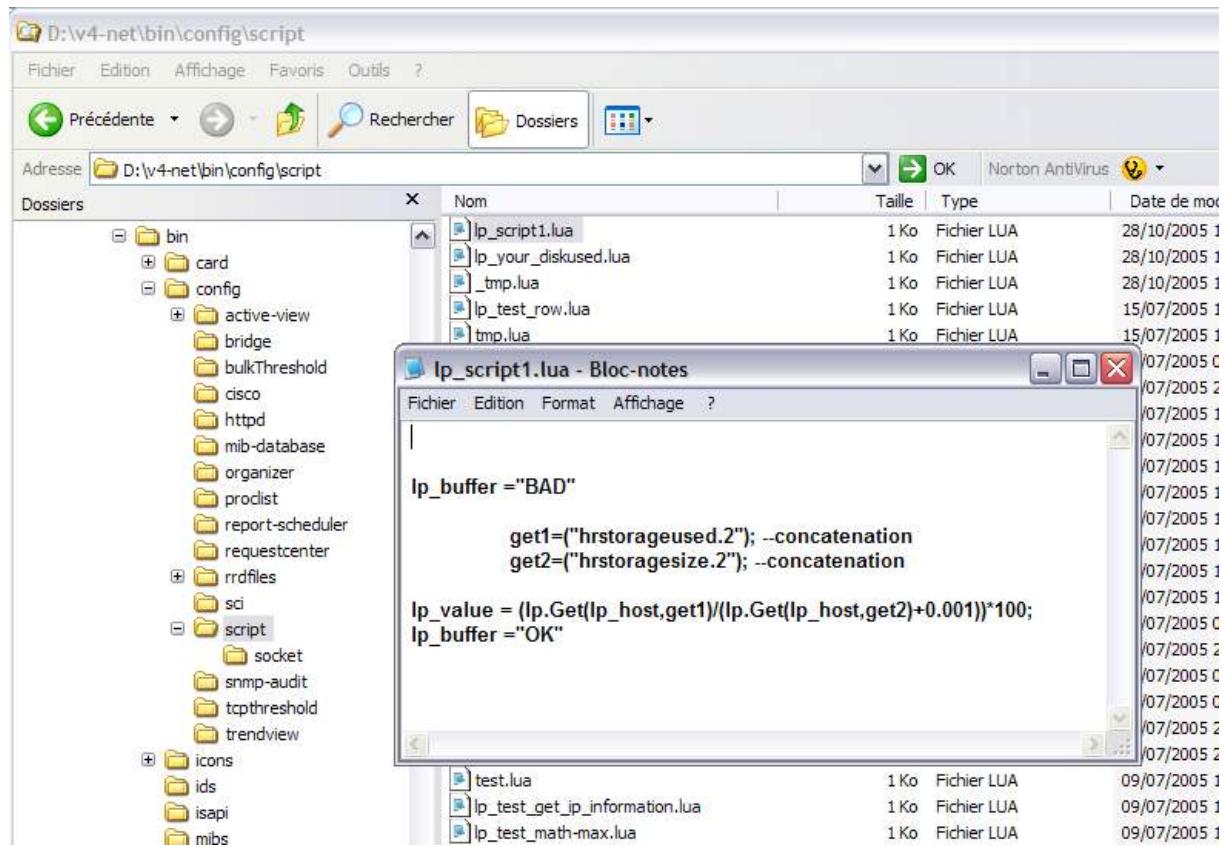


Il suffit de compiler la MIB **LUTEUS-SCRIPT-STANDARD-MIB** pour pouvoir disposer de 40 objets à définir simplement en créant les fichiers de scripts correspondant dans le répertoire bin/config/script.

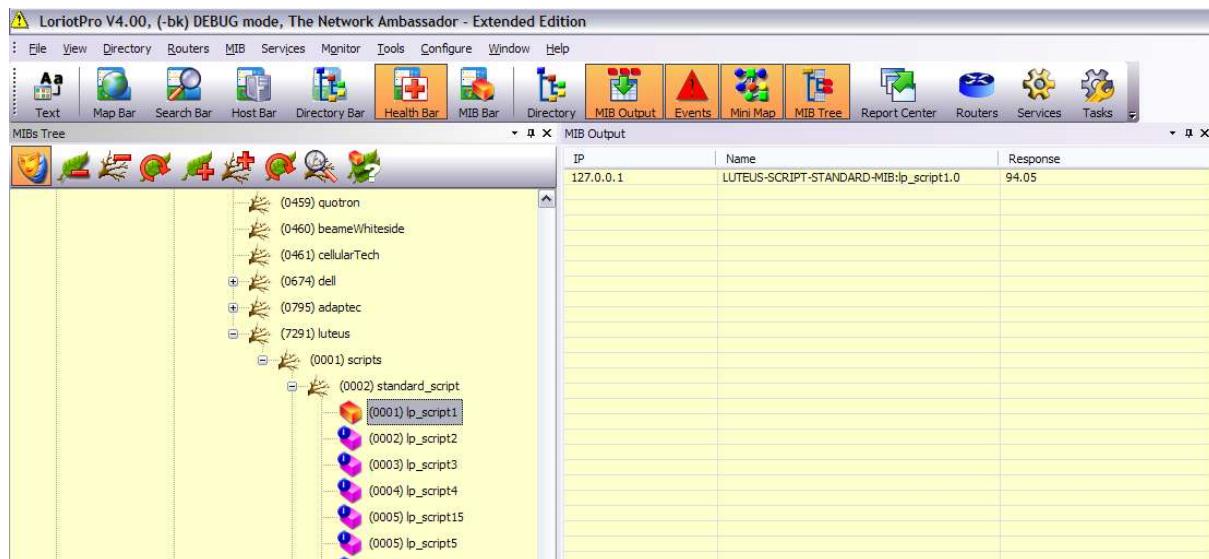


Si vous voulez utiliser l'objet **Ip\_script1**, créez simplement un script **Ip\_script1.lua** dans le répertoire **bin/config/script** et l'objet sera disponible.

## LoriotPro V5.00 Extended Edition – Supervision et gestion SNMP par scripts LUA Introduction au System Expert et aux scripts



Dans cet exemple nous n'utilisons plus le passage d'index car il est directement codé dans le script.



Par défaut, ces objets virtuels sont de type integer32 ; si vous désirez réaliser des objets qui retournent une chaîne de caractères (« string »), voici la syntaxe à utiliser dans le fichier de MIB.

<code>lp_your_string</code>	<code>OBJECT-TYPE</code>
<code>SYNTAX</code>	<code>OCTETSTRING</code>
<code>ACCESS</code>	<code>lp_access_script</code>
<code>STATUS</code>	<code>current</code>

```
DESCRIPTION
    "return a string"
<LP_SCRIPT>

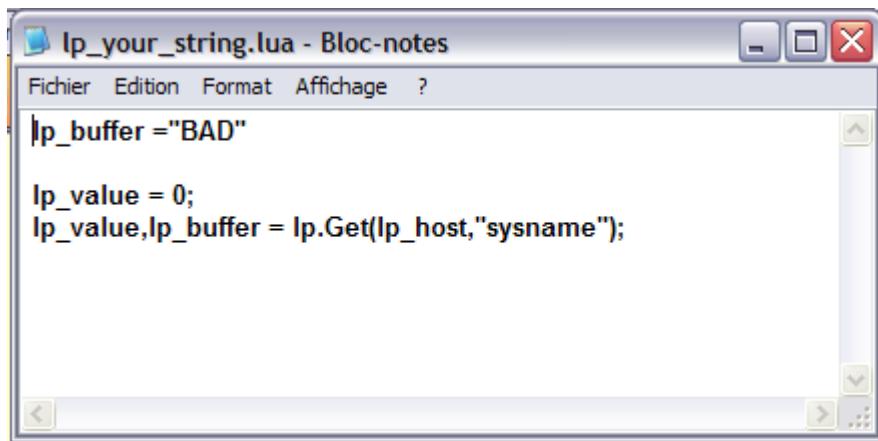
lp_buffer ="BAD"

lp_value, lp_buffer =lp.Get(lp_host,"sysname.0")

<LP_SCRIPT>

::= { your_script 2 }
```

Après compilation, le fichier **lp\_your\_string.lua** est créé dans le répertoire **bin/config/script**.



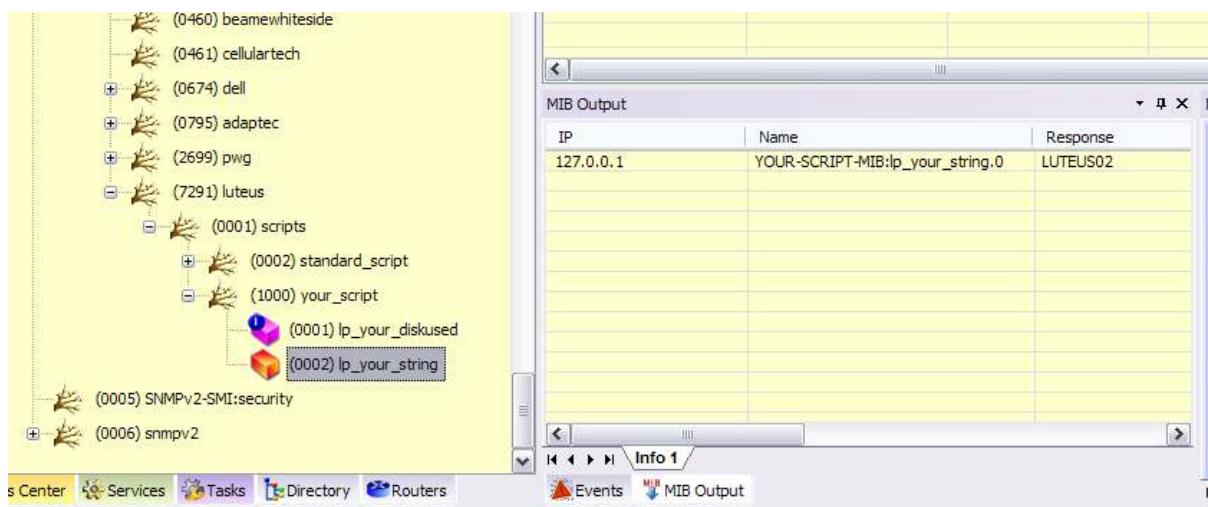
The screenshot shows the LoriotPro Script Editor window. The title bar reads "LoriotPro - Script Editor : D:\v4-net\bin\config\script\lp\_your\_string.lua". The menu bar includes Files, Edit, Insert, Search, Options, Compiler, and Value. The main code area contains the following Lua script:

```
1 ip_buffer = "BAD"
2
3 ip_value = 0;
4 ip_value,ip_buffer = Ip.Get(ip_host,"sysname");
5
```

Below the code area is a table titled "Value Name" with columns "Value Name", "Value in", and "Response". The table contains the following data:

Value Name	Value in	Response
ip_host	127.0.0.1	127.0.0.1
ip_index		
ip_trace	1	1
ip_oid		
ip_buffer		LUTEUS02
ip_value	0	

Le nouvel objet snmp virtuel est utilisable et renvoie une chaîne de caractères.



### Attention

Vous ne pouvez pas recompiler un fichier de MIB se trouvant déjà dans la data base du logiciel ; par contre vous pouvez modifier les scripts. Si vous voulez réinitialiser vos fichiers de MIB, utilisez la fonction « Reset to Default... » du compilateur de MIB et relancez LoriotPro.



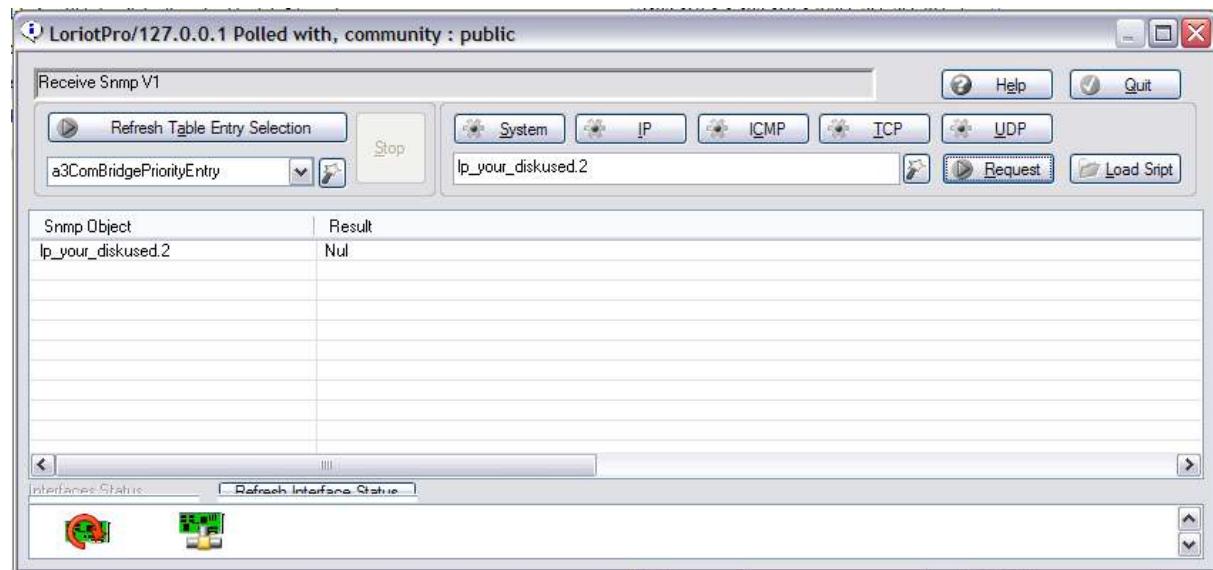
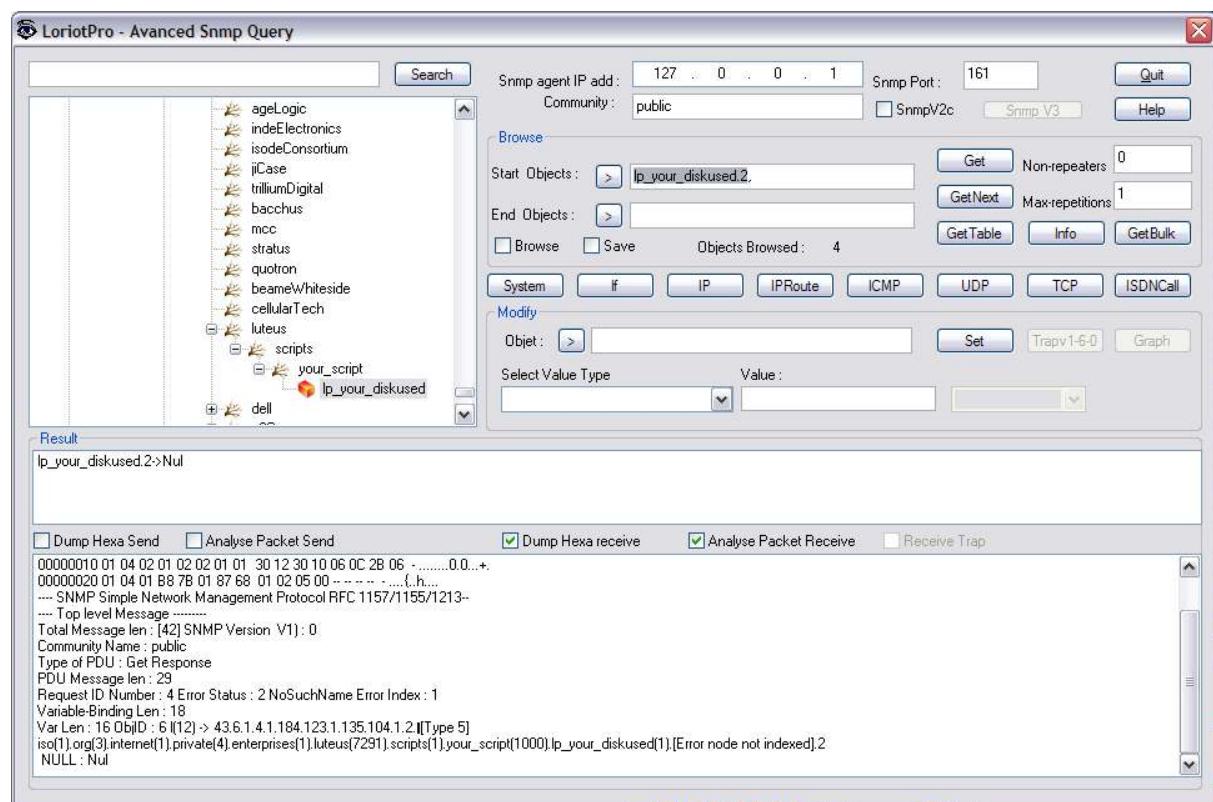
**Attention**

Si vous modifiez un script qui est utilisé par le logiciel, vous pouvez avoir des dysfonctionnements car plusieurs processus vont utiliser le même fichier en même temps. Pensez à faire des sauvegardes de votre travail régulièrement.

### 11.1.3      Exception d'utilisation des objets SNMP virtuels

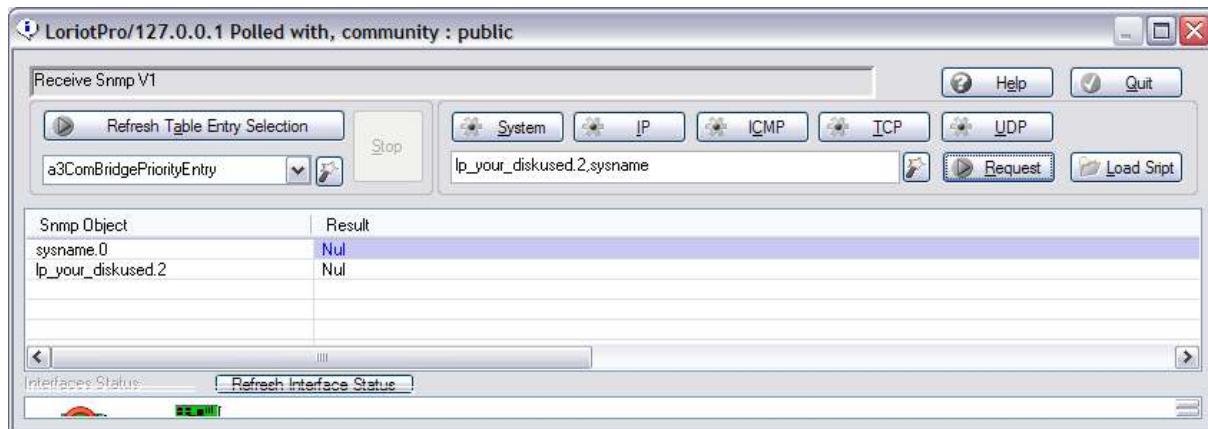
Les objets snmp virtuels ne sont pas utilisables avec des sockets de type asynchrone. Si vous développez des plugins à partir de notre SDK, ils devront utiliser une logique synchrone. Dans le Logiciel LoriotPro seuls les modules « Common Query », « Advanced Query » et le « Poller » qui travaillent en mode direct (asynchrone) n'ont pas la possibilité d'utiliser des objets snmp virtuels. Les autres plugins et modules de LoriotPro travaillent avec des requêtes en mode synchrone et ils supportent les objets snmp virtuels.

## LoriotPro V5.00 Extended Edition – Supervision et gestion SNMP par scripts LUA Introduction au System Expert et aux scripts



**Attention**

Mixer dans des requêtes des objets SNMP virtuels et des objets SNMP réels, peut, en fonction de l'agent, poser des problèmes.



L'envoi de l'objet ***lp\_yourdiskused.2*** vers l'agent inhibe la réponse sur ***sysname***.

## 12 ActiveView

### 12.1 Introduction

Les Activeviews sont un élément important du système expert car elles permettent la représentation graphique de tous les résultats collectés par les autres modules de LoriotPro ou même des résultats collectés par l'ActiveView elle-même. Cette représentation graphique, loin d'être uniquement statique avec simplement des changements de couleurs des objets pour signifier des changements d'états, peut être dynamique. Il est possible de construire entièrement dynamiquement à travers des scripts une ActiveView. Une fois créés, les objets graphiques peuvent être modifiés, déplacés, effacés...Une ActiveView va utiliser, dans certains cas, des scripts attachés aux objets graphiques de la MAP pour réaliser des opérations diverses. Il n'est cependant pas obligatoire d'écrire des scripts pour utiliser des Activeviews. Dans cet ouvrage, nous ne traiterons que de l'aspect scripté des Activeviews. Loriotpro version 5 Extended Edition n'apporte pas de nouveautés majeures autres que l'augmentation du nombre de fonctions disponibles dans les librairies standard. Certaines captures d'écran sont de la Version 4.

### 12.2 Principe d'utilisation des scripts

Les scripts utilisés par les ActiveViews peuvent être attachés aux objets graphiques de la MAP. Dans ce cas, les scripts sont exécutés à intervalles réguliers (polling interval). Le planificateur (scheduler) d'actions associé à chaque ActiveView parcourt en permanence la liste chaînée des objets de la View et exécute l'expression (Request). Lorsque le temps est échu, les objets sont chaînés, ce qui entraîne l'exécution des scripts les uns après les autres (en fonction du polling interval). Un script ne sera pas exécuté une seule fois mais à chaque intervalle de temps. Il est possible de désactiver l'exécution du script en utilisant dans le script une commande modifiant les paramètres de polling, mais un script désactivé ne pourra pas de lui-même se réactiver. Le script d'un objet peut interférer sur les paramètres d'un autre objet et réactiver le script ou changer ses paramètres.

### 12.3 Types de scripts utilisables dans les ActiveViews

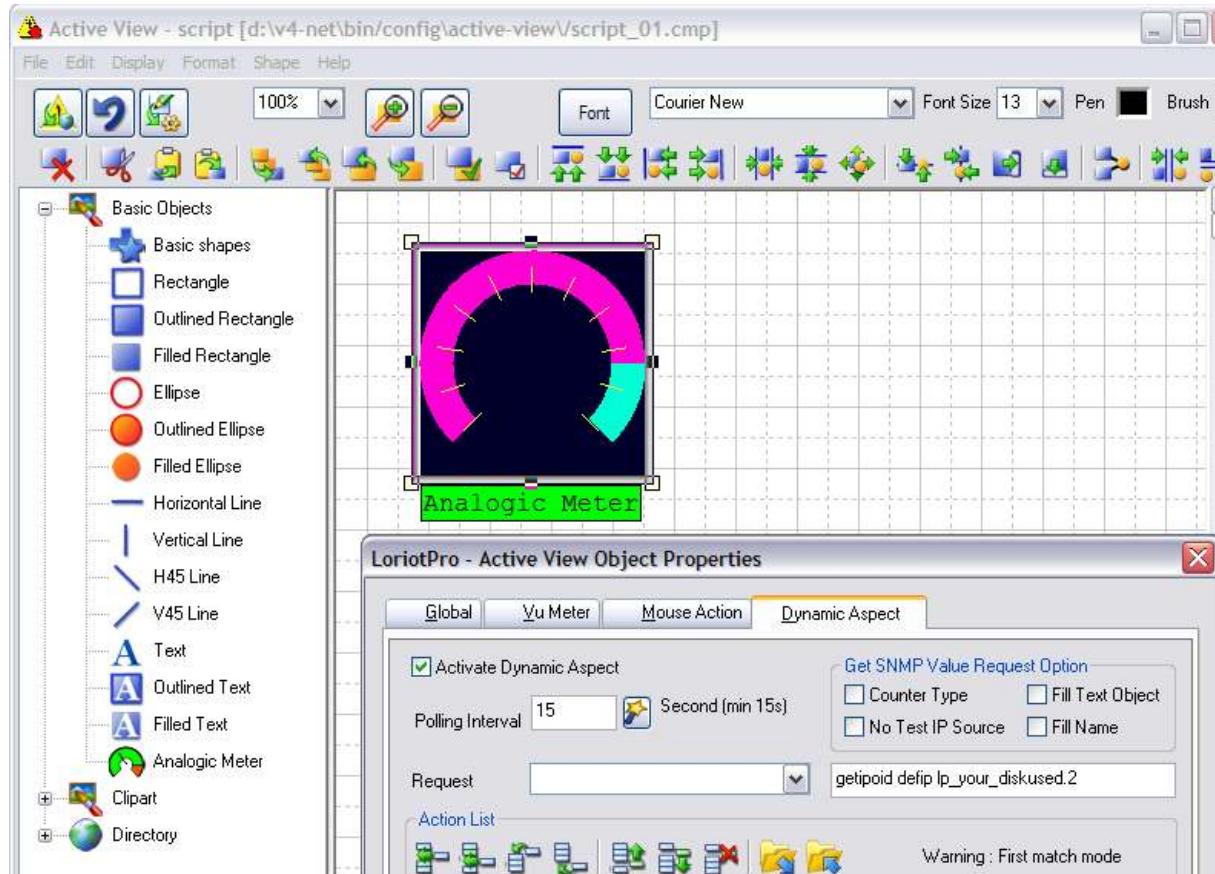
Les ActiveViews permettent un usage multiple des scripts LUA.

1. Un usage direct d'objet snmp virtuel est possible. Pour rappel un objet snmp virtuel est pour LoriotPro un objet qui déclenche un script lors de son utilisation.
2. Un usage direct est possible en ligne de commande avec l'option Getipoid sous la forme d'une expression de type calculette scientifique.
3. Un usage direct de script en mode polling est aussi possible en pointant simplement sur le fichier avec des paramètres.
4. Un script peut être attaché à l'option d'action de double clic de la souris sur l'objet graphique.

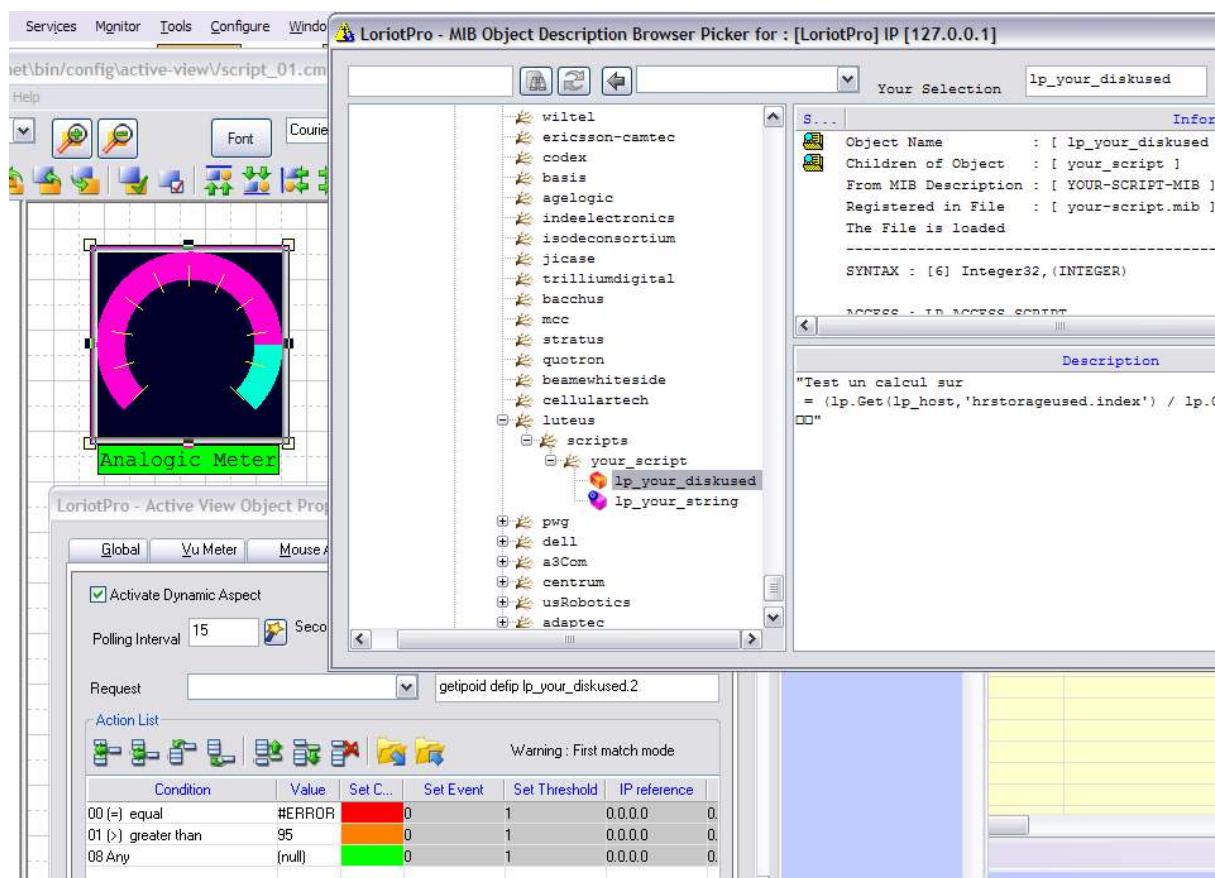
5. Un script peut être attaché au menu contextuel paramétrable de l'objet graphique.
6. Un Script peut être lancé à partir du menu principal.
7. Un script peut être associé à l'activation d'une ActiveView de type Modal.

## 12.4 Usage direct d'objets SNMP virtuels

L'utilisation d'objets snmp virtuels ne pose aucun problème, ils sont utilisés comme des objets snmp standard.



Le mode « Wizard » permet de sélectionner les objets snmp virtuels, mais il ne supporte pas l'indexation éventuelle de l'objet.



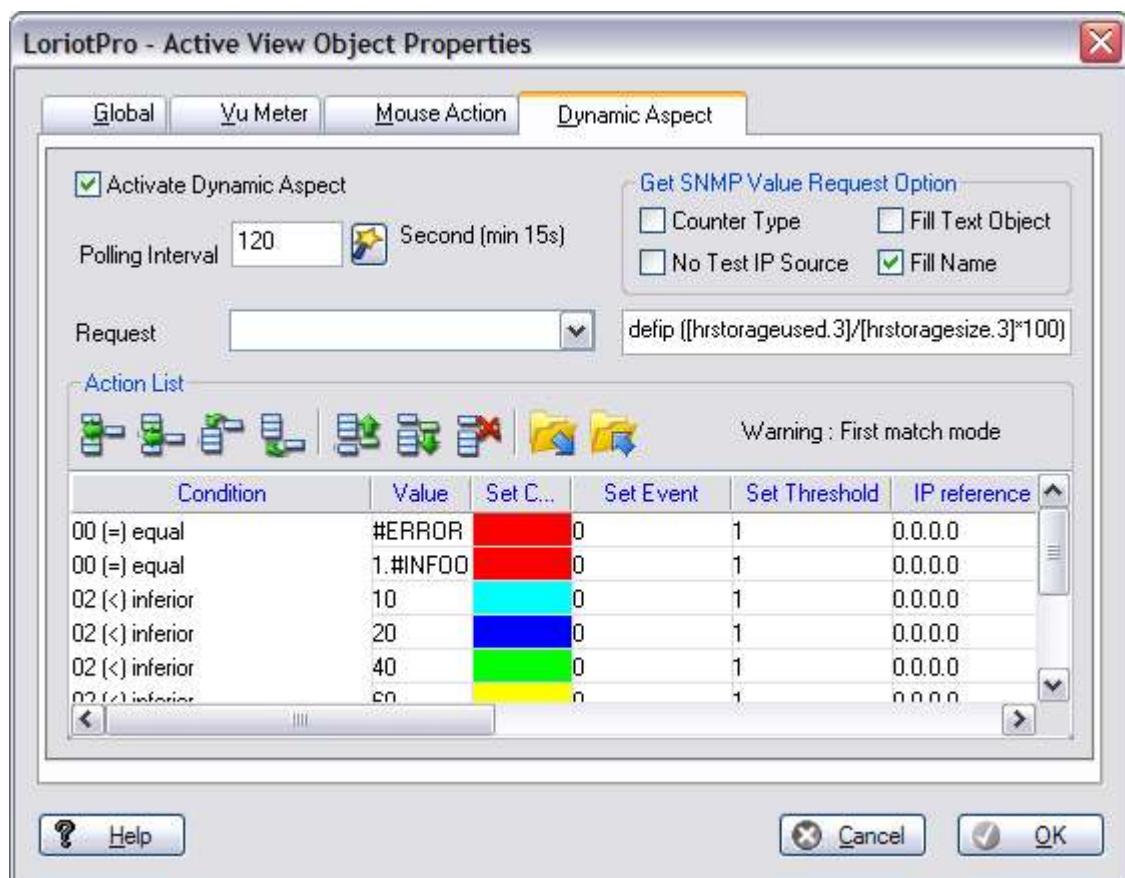
## 12.5 Calculette scientifique

En utilisant la fonction **GetipOld** standard, il est possible de réaliser des opérations mathématiques simples sur une collecte d'objets SNMP (en provenance d'un seul host).

### Getipoid defip expression

Expression :

Il suffit de mettre les objets snmp (oid) désirés entre [ ] dans la chaîne de calcul, la syntaxe est de type calculette scientifique (langage C, C++) avec des parenthèses. Pour la syntaxe des fonctions mathématiques avancées reportez-vous à la partie librairie « math » de la documentation LUA.



L'objet `[$lastvalue]` assigne dans la chaîne à son emplacement, la dernière valeur calculée.

Exemple :

Affiche l'utilisation d'un disque (d :) (host-resources-mib) en pourcentage

`getipoid defip ([hrstorageused.3]/[hrstoragesize.3]*100)`

Affiche l'espace libre sur le disque en Mbytes

`getipoid defip ((([hrStorageSize.3] - [hrstorageused.3])*[hrstorageallocationunits.3])/1024)/1024`

### Attention

Il vaut mieux éviter de mettre des blancs dans la chaîne expression.

Correct

`(([hrstorageused.3]/[hrstoragesize.3]*100)/1000)`

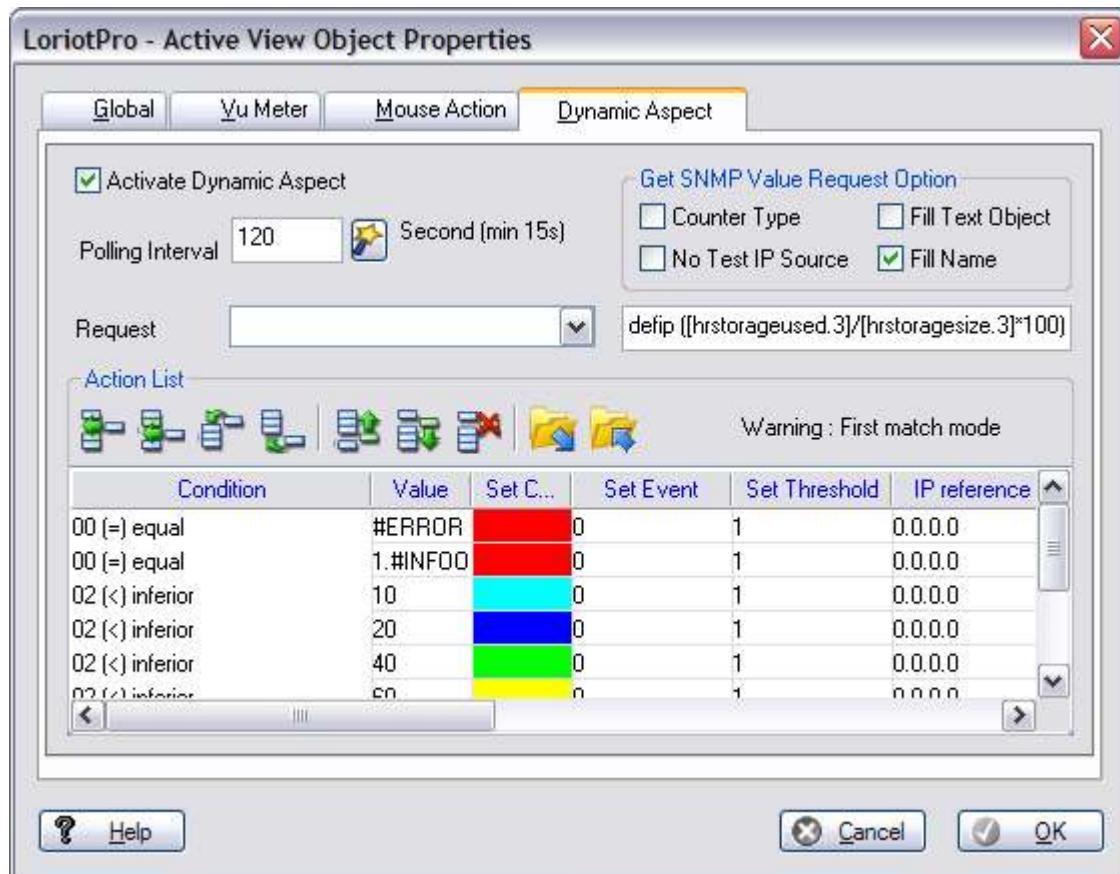
Les caractères blancs ne sont pas souhaités mais cela fonctionne (l'analyseur syntaxique ( parser ) de l'ActiveView supporte 50 blocs maximum).

(( [hrstorageused.3] / [hrstoragesize.3] \* 100 ) / 1000)

Valeur possible en retour d'expression

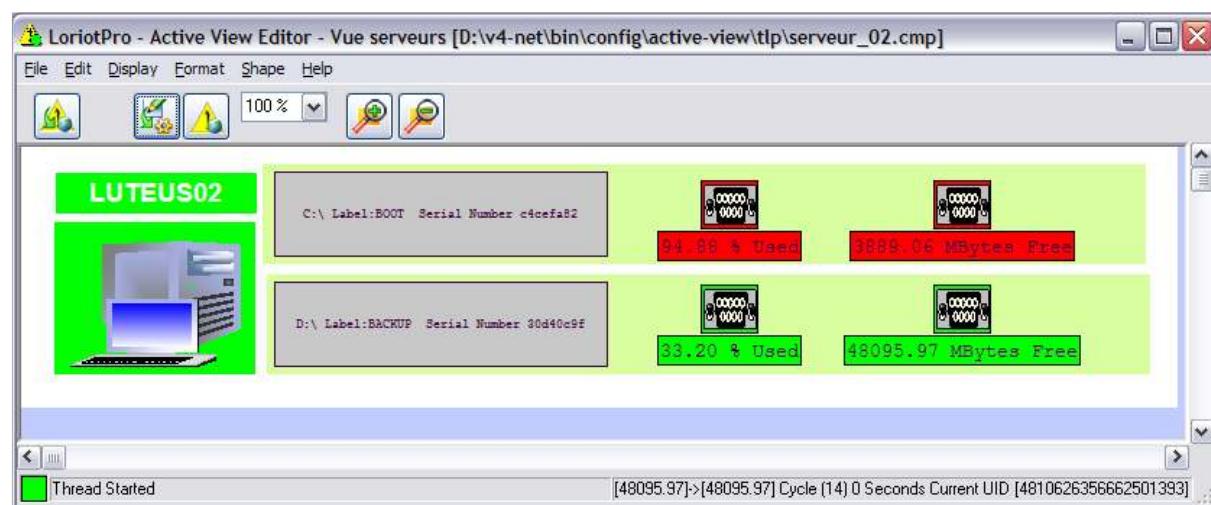
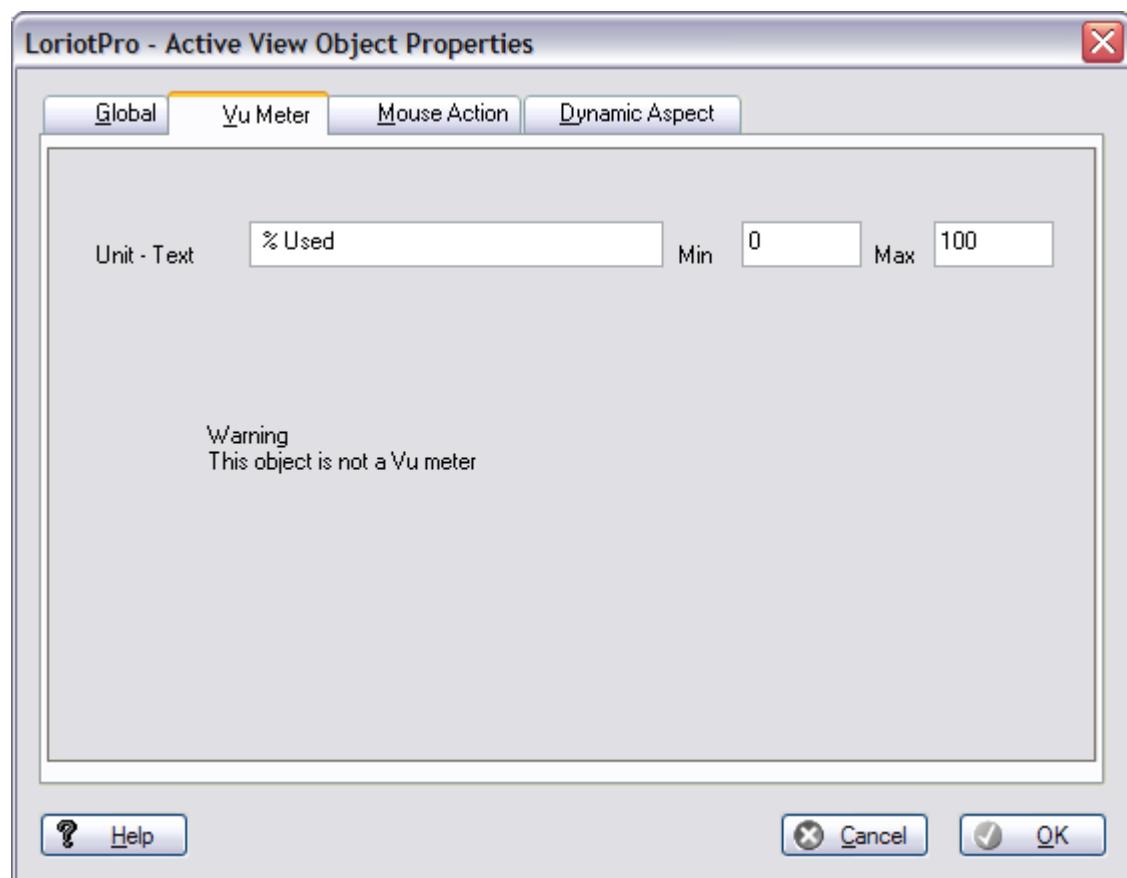
#ERROR si la syntaxe est incorrecte ou si les objets snmp (OID) ne sont pas trouvés

1.#INFO0 si il y a une erreur de division par zéro

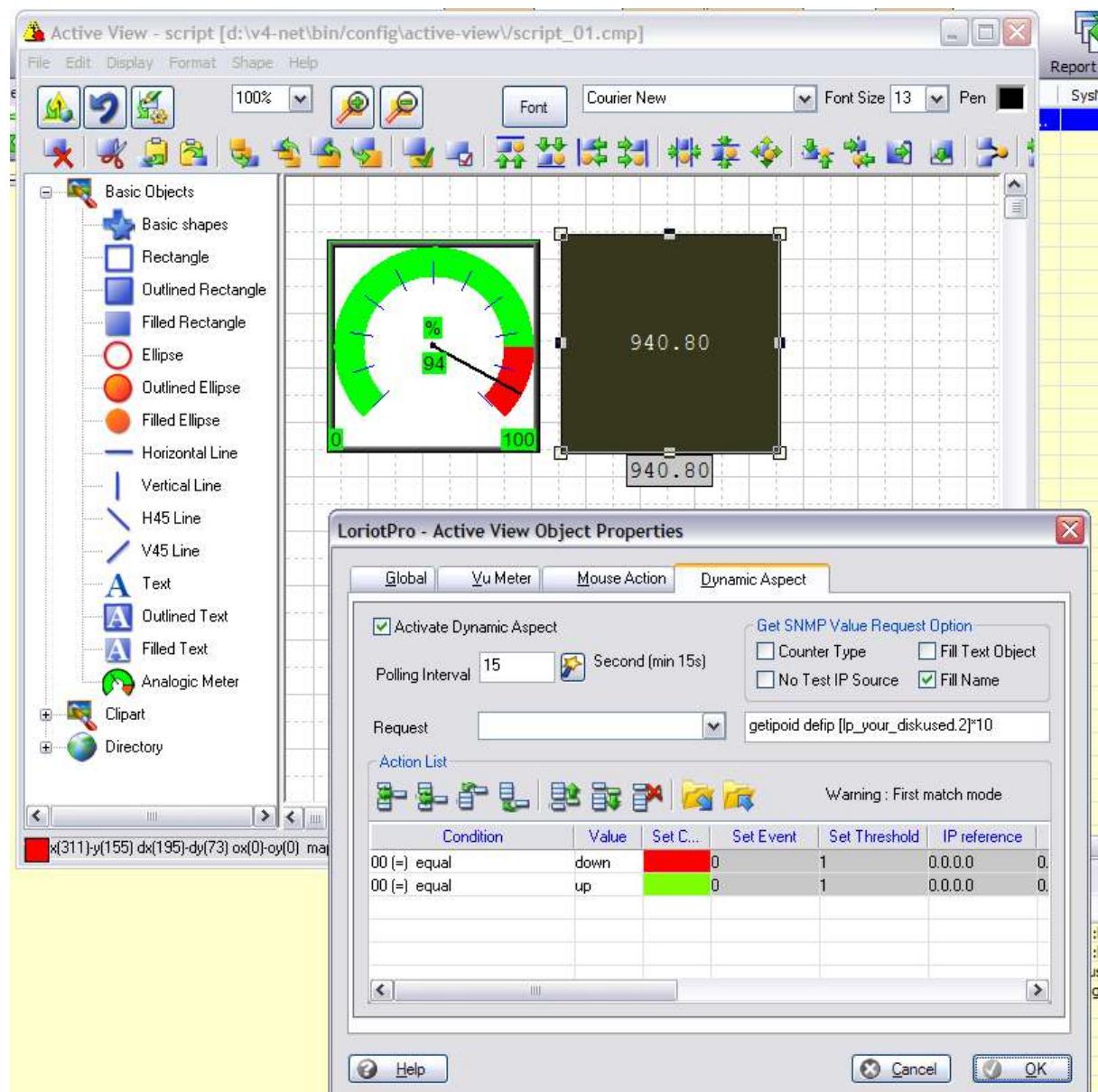


Si on met une valeur dans unit (fenêtre 'Vu Meter'), elle est automatiquement ajoutée au nom (« name ») de l'objet avec l'option 'fill name' ; cela permet d'avoir des noms dynamiques avec une unité.

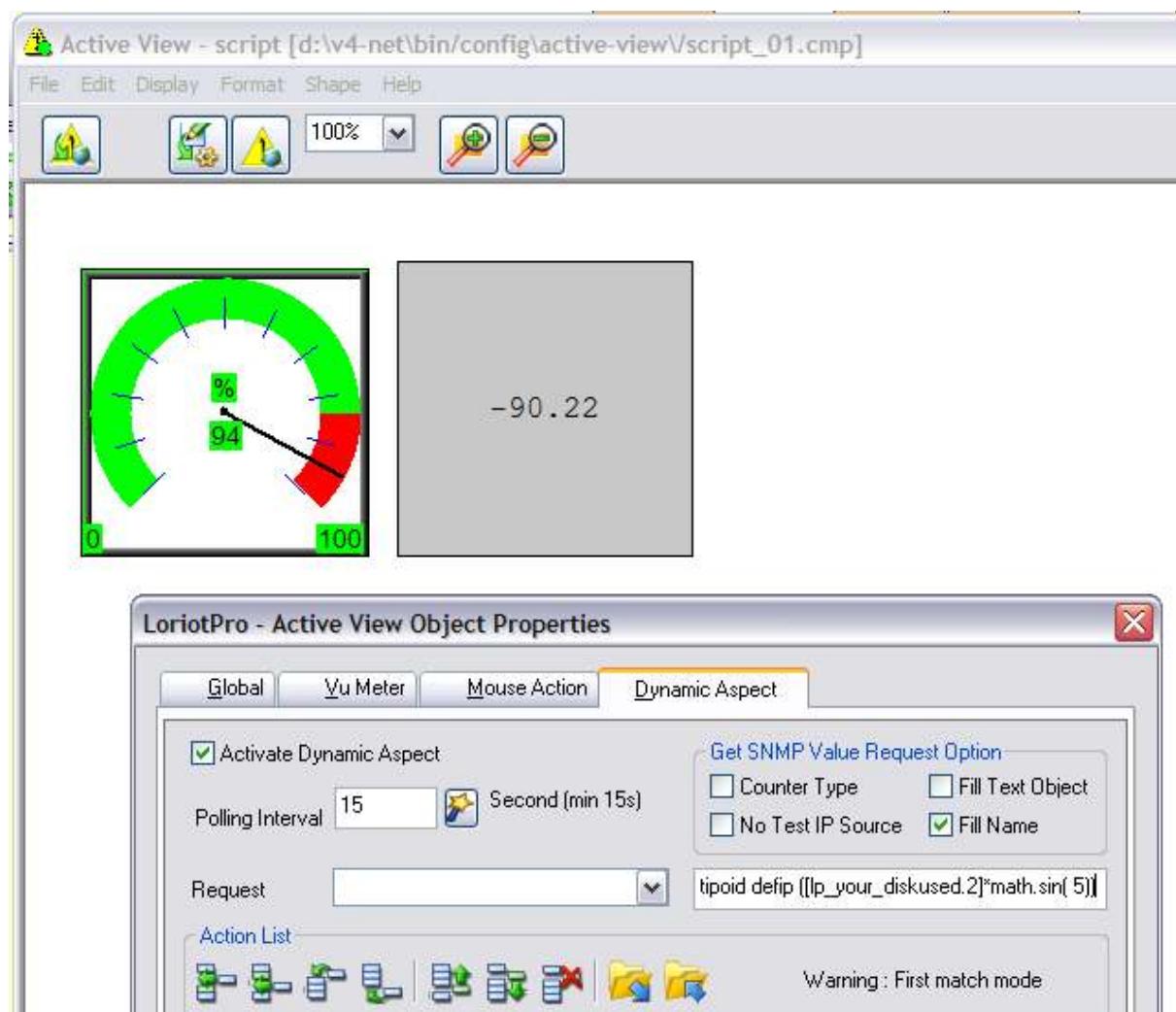
L'option : 'fill name' permet de remplacer le nom de l'objet par le résultat.  
L'option 'unit' dans (vu meter) est ajoutée au nom s'il existe.



Dans l'exemple suivant on utilise un objet snmp virtuel pour réaliser le calcul.



Cet exemple intègre du calcul avec des sinus.



### Extrait de la documentation LUA (5.5 - Mathematical Functions)

This library is an interface to most of the functions of the standard C math library. (Some have slightly different names.) It provides all its functions inside the table `math`. In addition, it registers the global `_pow` for the binary exponentiation operator `^`, so that `x^y` returns  $x^y$ . The library provides the following functions:

<code>math.abs</code>	<code>math.acos</code>	<code>math.asin</code>	<code>math.atan</code>	<code>math.atan2</code>
<code>math.ceil</code>	<code>math.cos</code>	<code>math.deg</code>	<code>math.exp</code>	<code>math.floor</code>
<code>math.log</code>	<code>math.log10</code>	<code>math.max</code>	<code>math.min</code>	<code>math.mod</code>
<code>math.pow</code>	<code>math.rad</code>	<code>math.sin</code>	<code>math.sqrt</code>	<code>math.tan</code>
<code>math.frexp</code>	<code>math.ldexp</code>	<code>math.random</code>	<code>math.randomseed</code>	

plus a variable `math.pi`. Most of them are only interfaces to the corresponding functions in the C library. All trigonometric functions work in radians (previous versions of Lua used degrees). The functions `math.deg` and `math.rad` convert between radians and degrees.

The function `math.max` returns the maximum value of its numeric arguments. Similarly, `math.min` computes the minimum. Both can be used with 1, 2, or more arguments.

The functions `math.random` and `math.randomseed` are interfaces to the simple random generator functions `rand` and `srand` that are provided by ANSI C. (No guarantees can be given for their statistical properties.) When called without arguments, `math.random` returns a pseudo-random real number in the range  $[0,1]$ . When called with a number  $n$ , `math.random` returns a pseudo-random integer in the range  $[1,n]$ . When called with two arguments,  $l$  and  $u$ , `math.random` returns a pseudo-random integer in the range  $[l,u]$ . The `math.randomseed` function sets a "seed" for the pseudo-random generator: Equal seeds produce equal sequences of numbers.

## 12.6 Lancement scheduler de script LUA

Il est possible de lancer des scripts (fichiers LUA) directement avec la commande RunScript dans le champ expression d'un objet de l'ActiveView. Chaque objet d'une ActiveView peut donc faire tourner un script. Dans le cas d'un script lancé par une ActiveView, seul le paramètre **Ip\_buffer** sera retourné par le script et un test classique pourra être réalisé dessus ; mais vous devez obligatoirement définir la variable **Ip\_value** dans votre script (`Ip_value=0` ;).

### Attention

Le script sera exécuté à chaque intervalle de polling. Si vous voulez exécuter un script de façon unitaire, utilisez les fonctions du menu contextuel associées à l'objet (création d'un nouveau champ de menu dédié) ou configurez l'option de « double clic » (Action) de la souris associée à l'objet graphique.

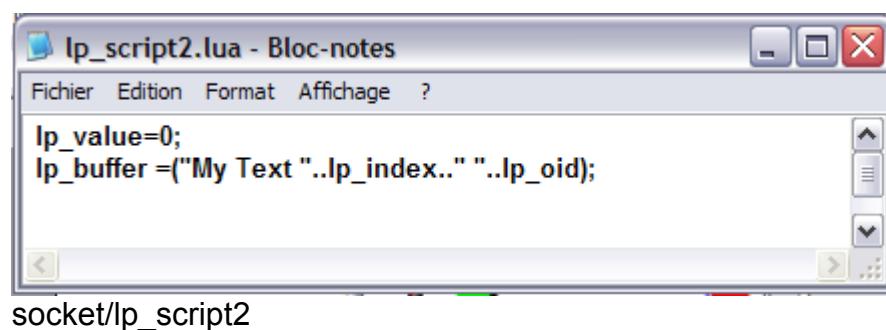
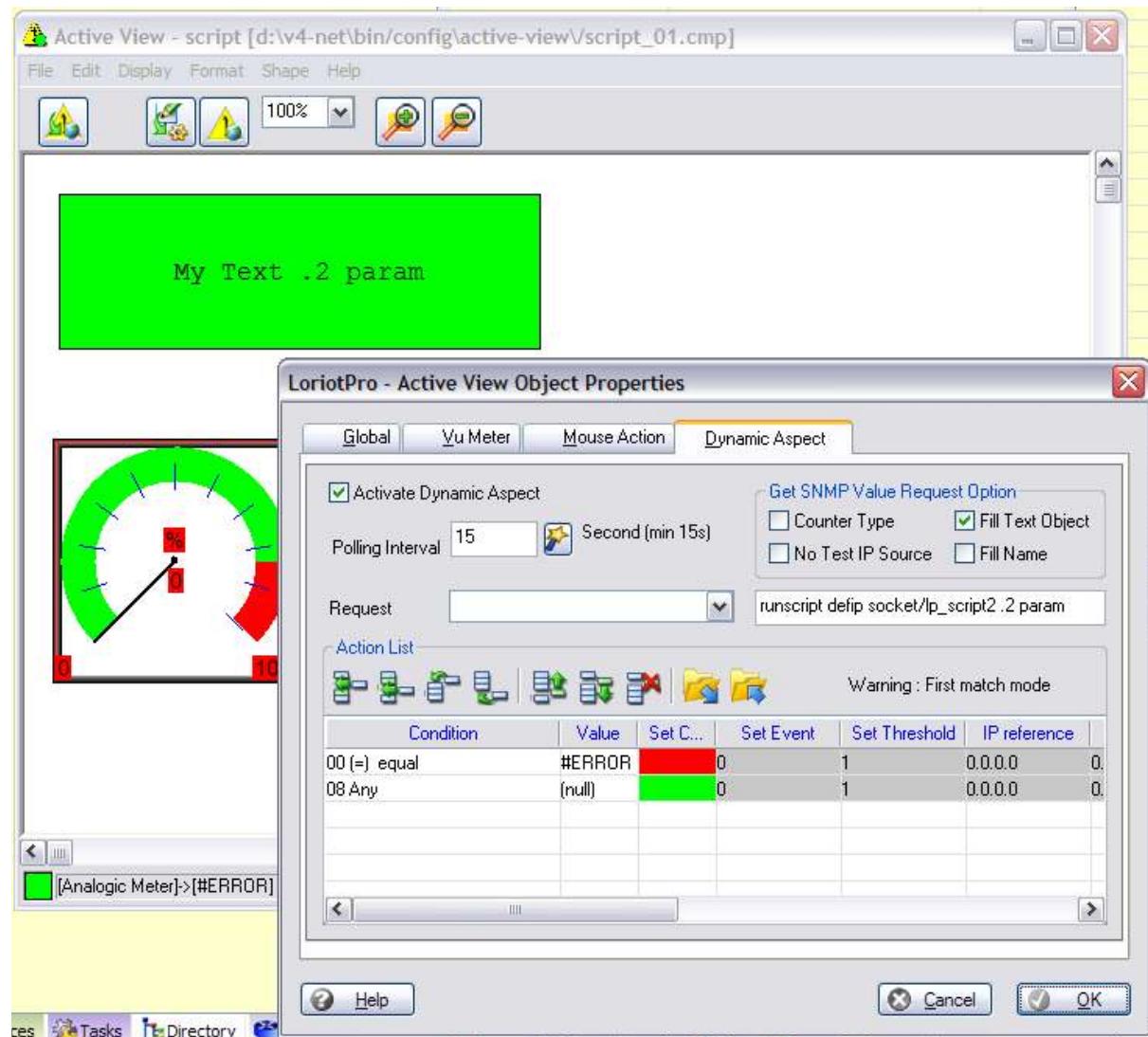
La syntaxe complète est la suivante :

**RunScript ip\_adresse script\_file Ip\_index Ip\_oid**

Termes	Syntaxe remarques
RunScript	La commande
ip_adresse	L'adresse IP de l'équipement concerné en mode doté (X.Y.Z.W) ou <i>defip</i> qui utilise l'adresse IP par défaut de l'ActiveView. Ce paramètre est passé dans la variable <b>Ip_host</b> au script.
script_file	Le nom du fichier lua. Le fichier doit se trouver impérativement dans le répertoire <i>bin/config/script</i> et il ne faut pas mettre l'extension <i>.lua</i> . Si vous créez un sous répertoire dans <i>bin/config/script</i> , par exemple <i>socket</i> , vous pouvez lancer des scripts de ce répertoire avec la syntaxe.  Socket/nom_fichier

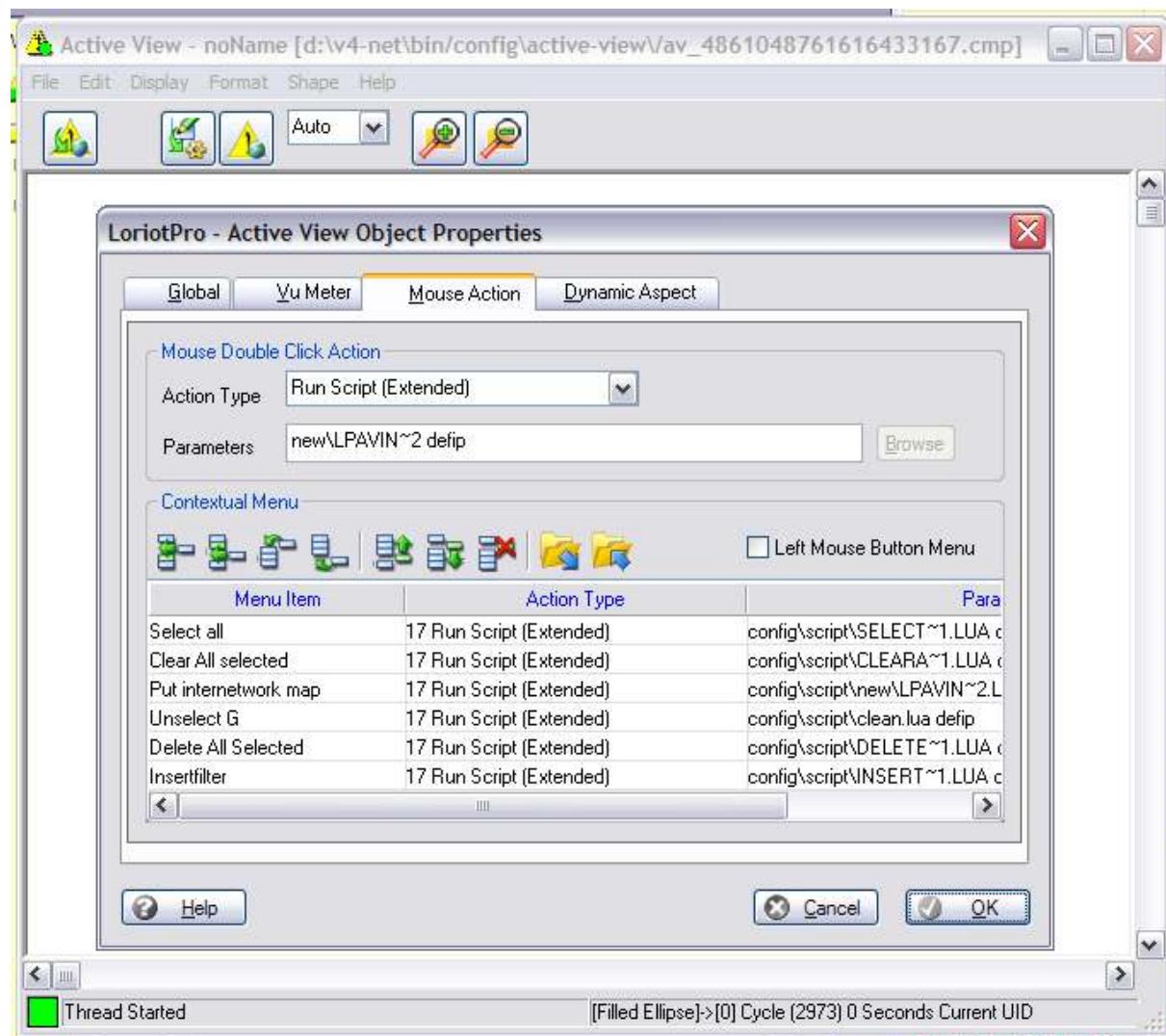
	<p><b>Exemples</b></p> <p>Soit le fichier script1.lua dans <i>bin/config/script</i>.</p> <p>RunScript defip script1</p> <p>Soit le fichier script1.lua dans <i>bin/config/script/socket</i>.</p> <p>RunScript defip socket/script1</p>
lp_index	<p>Ce paramètre est optionnel, le programme passe ce paramètre dans la variable <b>Ip_index</b> du script.</p> <p><b>Attention</b> </p> <p>Ce paramètre ne doit pas contenir de caractères blancs.</p>
	<p><b>Exemples</b></p> <p>Soit le fichier script1.lua dans <i>bin/config/script</i>.</p> <p>RunScript defip script1 .2</p> <p>Soit le fichier script1.lua dans <i>bin/config/script/socket</i>.</p> <p>RunScript defip socket/script1 .2</p>
lp_oid	<p>Ce paramètre est optionnel, le programme passe ce paramètre dans la variable <b>Ip_oid</b> du script.</p> <p><b>Attention</b> </p> <p>Ce paramètre ne doit pas contenir de caractères blancs et pour l'utiliser il faut un paramètre <b>Ip_index</b> dans la ligne même si il ne sert à rien.</p>
	<p><b>Exemples</b></p> <p>Soit le fichier script1.lua dans <i>bin/config/script</i>.</p> <p>RunScript defip script1 .2 param</p>

	Soit le fichier script1.lua dans <i>bin/config/script/socket</i> .  RunScript defip socket/script2 .2 param
--	---



## 12.7 Script utilisé avec la fonction DoubleClick

Un script peut être associé à l'action de double clic sur un objet graphique. Ceci permet de dérouler un script sur action uniquement et non plus de façon régulière (polling) et continue.

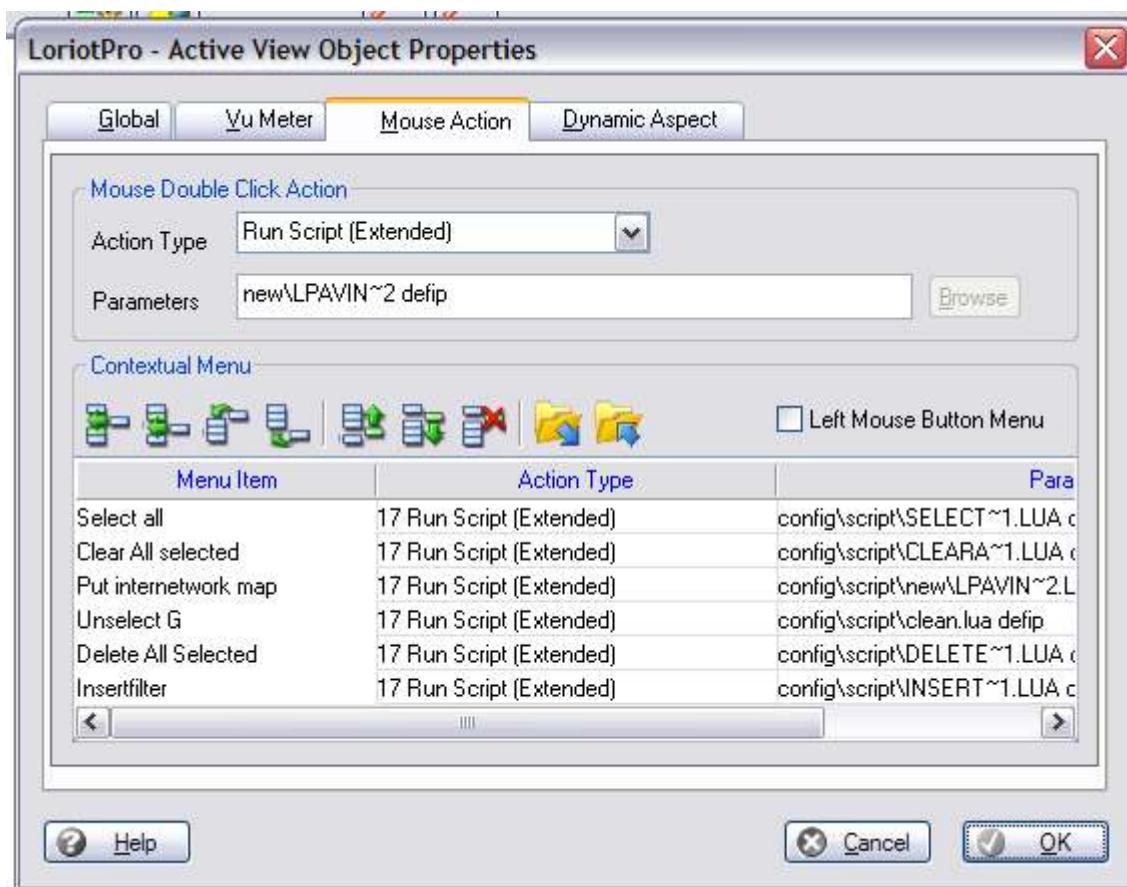


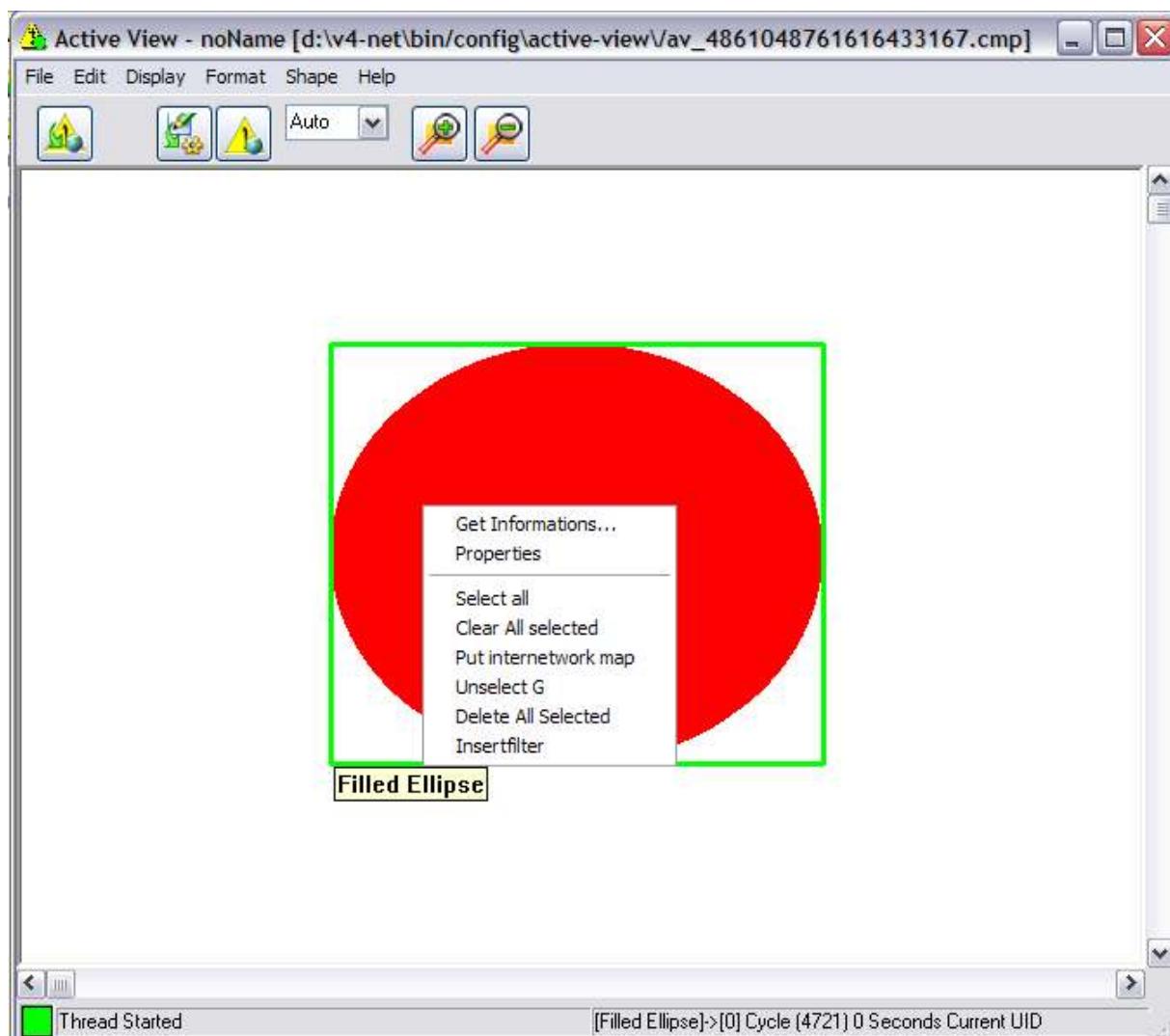
La syntaxe complète des paramètres est :

***filename.lua ip\_adresse index oid***

## 12.8 Script associé au menu contextuel d'un objet graphique

Un script peut être associé au menu contextuel paramétrable d'un objet graphique. Ceci permet de dérouler une sélection de scripts sur action uniquement et non plus de façon régulière (polling) et continue.

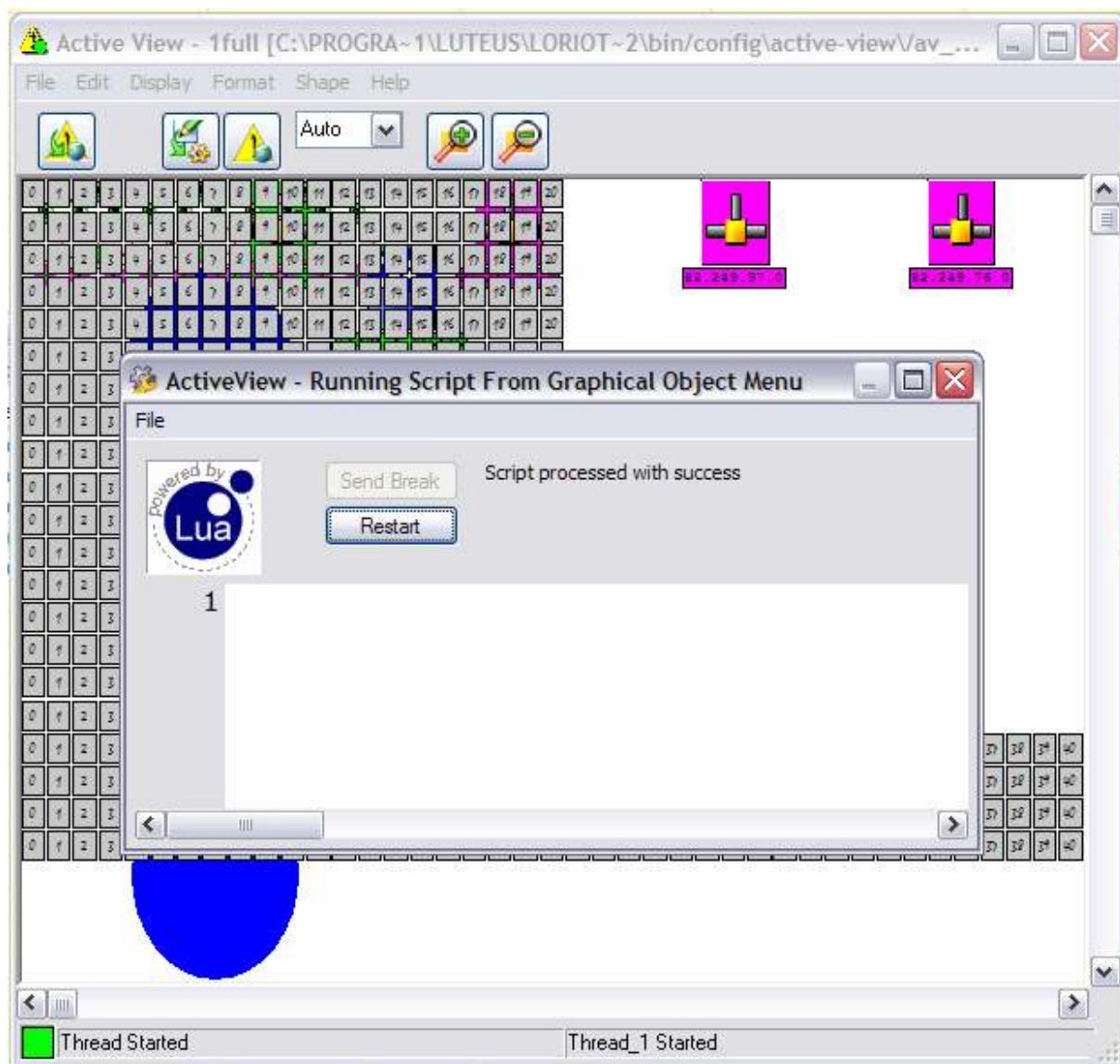




La syntaxe complète des paramètres est :

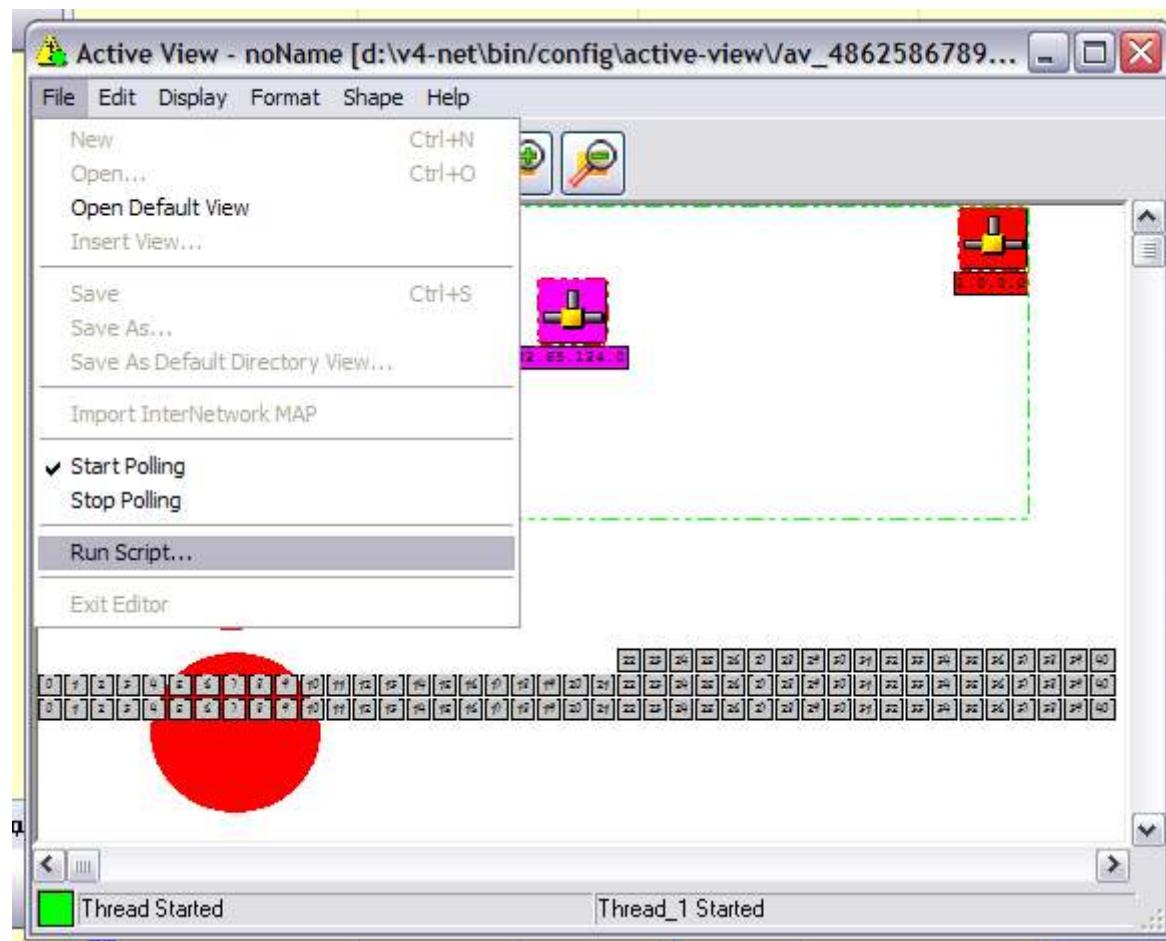
***filename.lua ip\_adresse index oid***

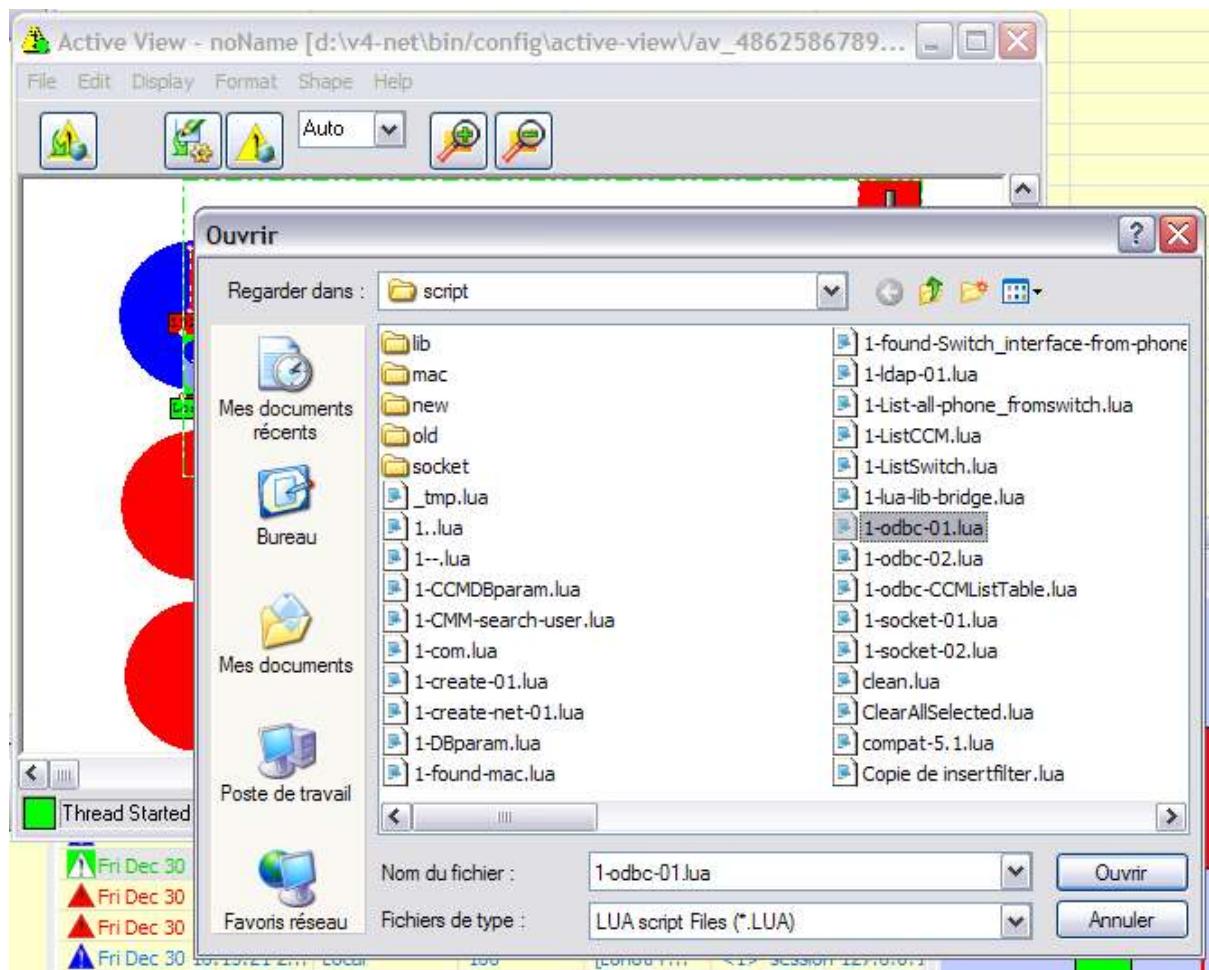
Une fenêtre pour gérer l'exécution du script est affichée.

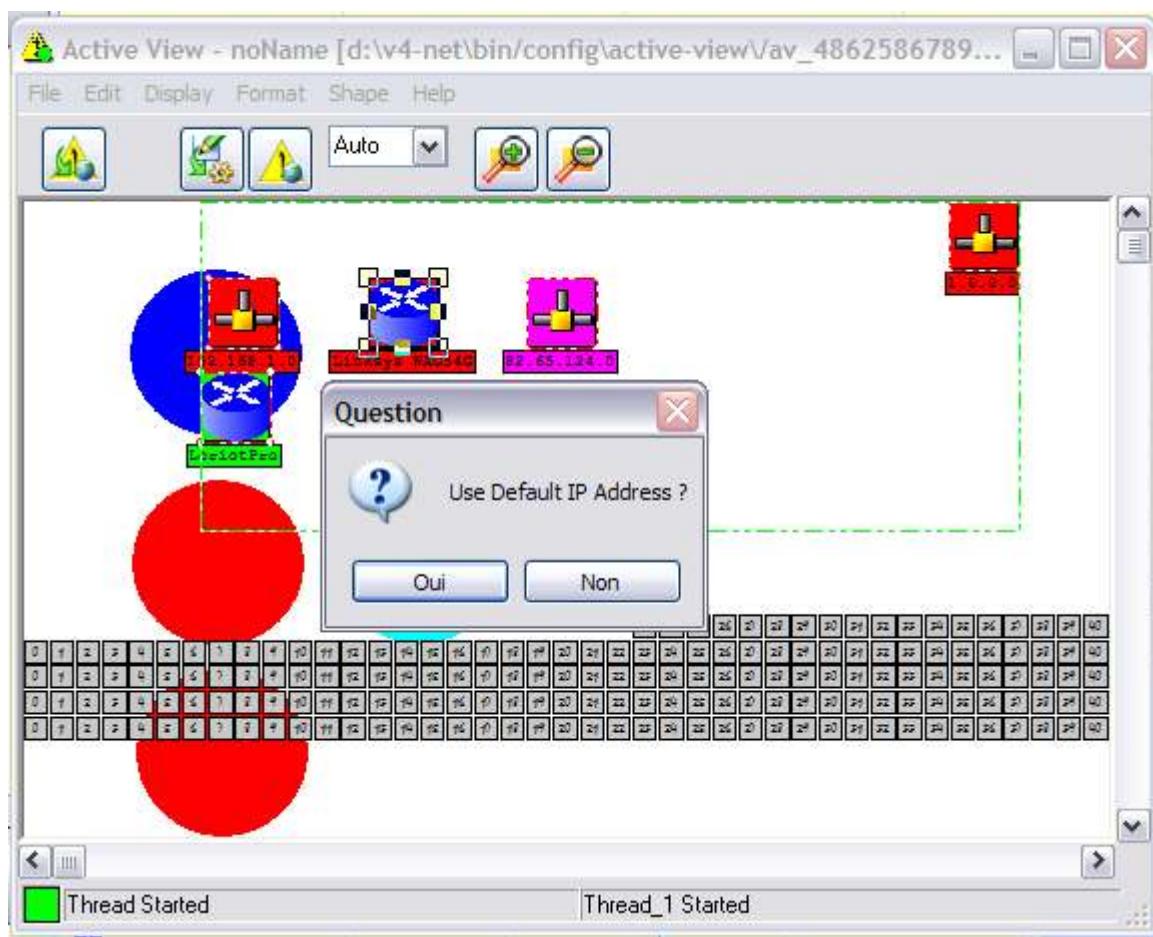


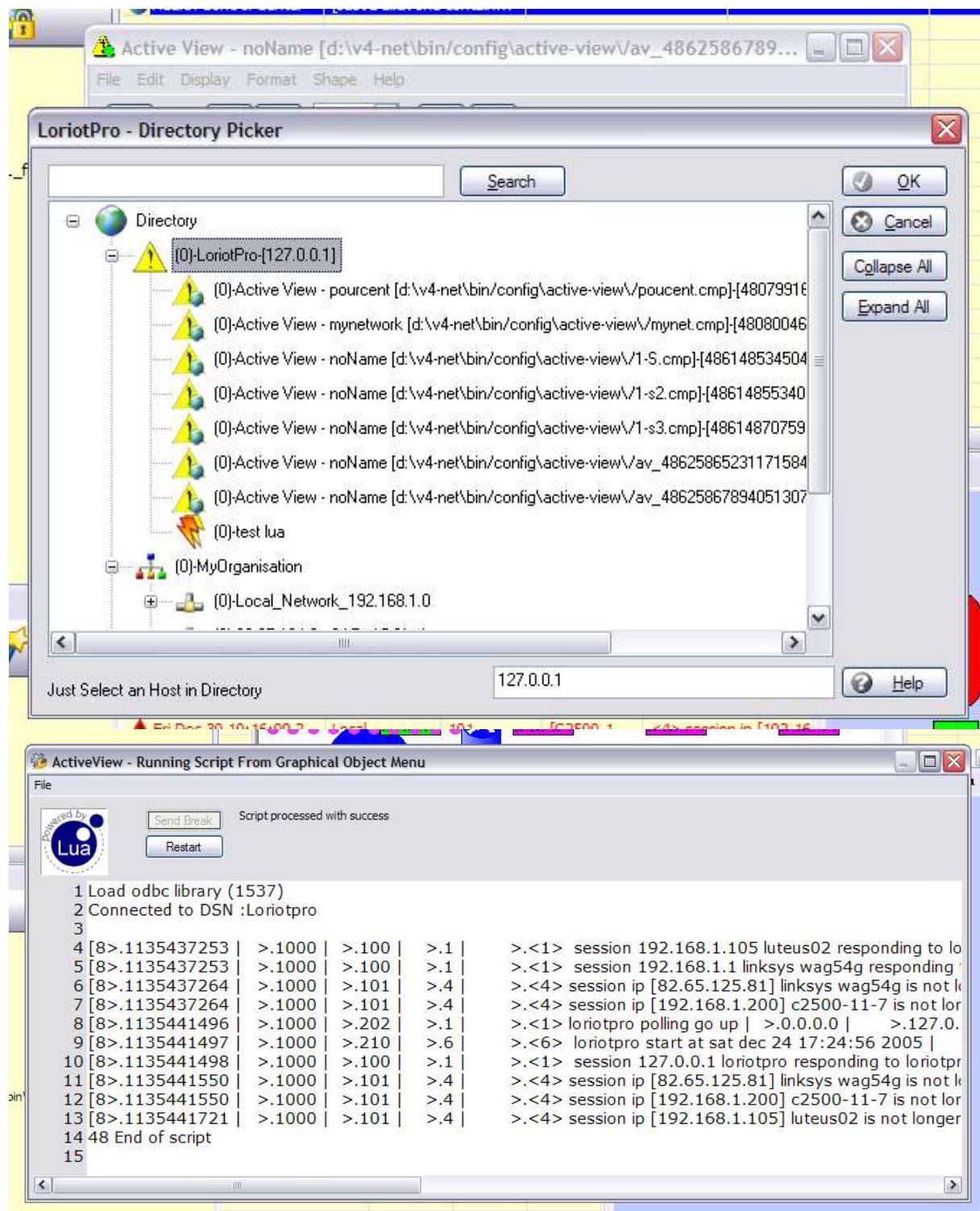
Cliquez sur l'icône croix rouge pour quitter cette fenêtre.

## 12.9 Utilisation directe de scripts dans une ActiveView



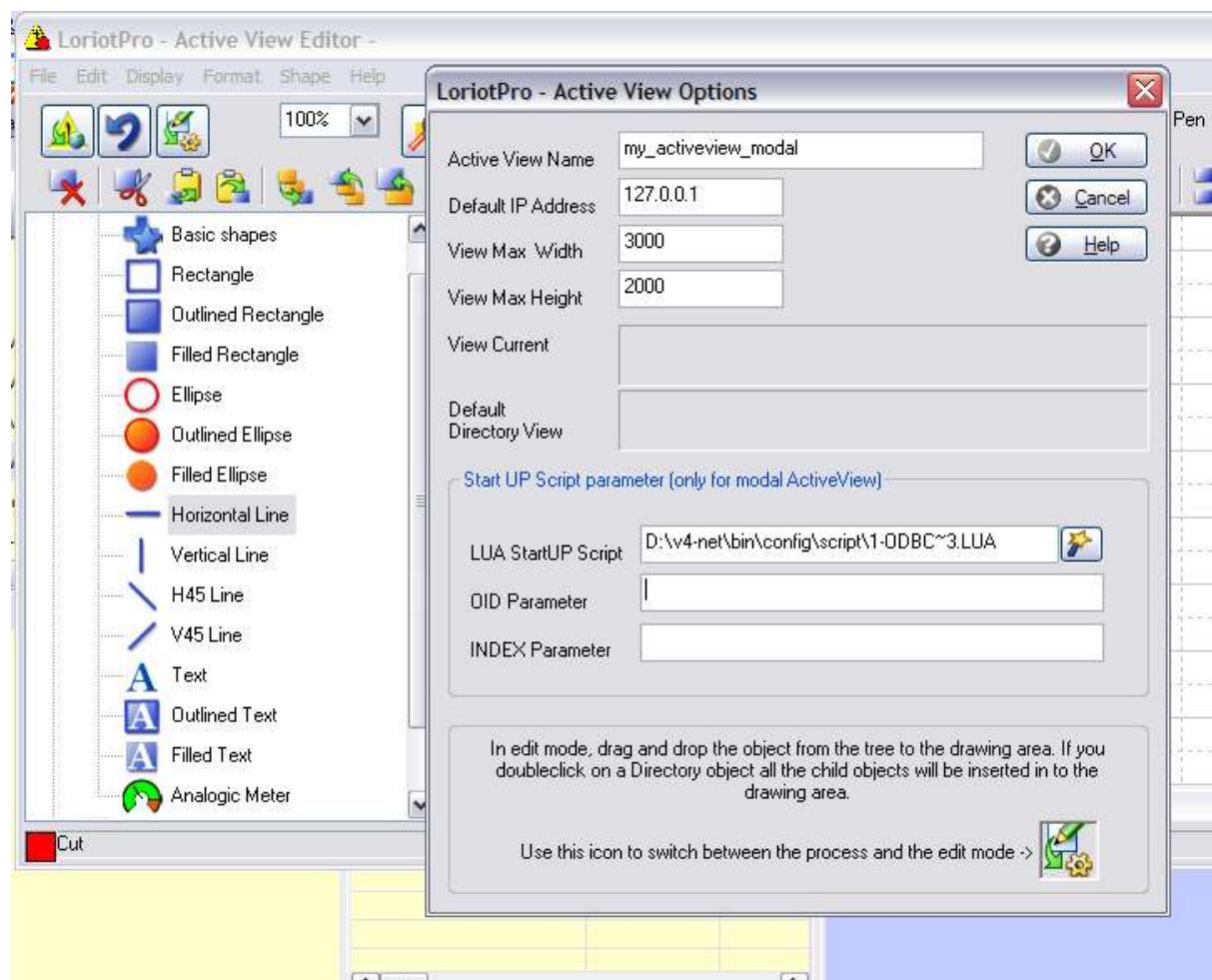






## 12.10 Utilisation de Script d'initialisation pour les ActiveView Modal

Il est possible d'associer un script d'initialisation au démarrage d'une ActiveView Modal. Pour configurer ce Script, il faut utiliser la fenêtre de propriété de l'ActiveView.



Une ActiveView Modal peut être appelée à partir du **Report Center**, d'un **ShortCut**, du menu contextuel d'un objet graphique d'une ActiveView ou d'un double clic de souris sur un objet graphique d'ActiveView.

Paramètres	Description
Default IP Adress	<p>Le paramètre <b>Ip_host</b> passé au script.</p> <p><b>Attention</b> </p> <p>Dans le cas d'une utilisation à travers un shortcut de Directory, le <b>Ip_host</b> passé au script est celui indiqué dans la ligne de commande du shortcut (cela permet de travailler avec des ActiveViews utilisant la notion de « <b>defip</b> ») et ce paramètre est remplacé par l'adresse IP du shortcut.</p>
LUA StartUP Script	Le fichier lua lancé
OID Parameter	Le paramètre <b>Ip_oid</b> passé au script
INDEX Parameter	Le paramètre <b>Ip_index</b> passé au script

Reportez-vous au chapitre « ShortCut associant une ActiveView Modal et un Script LUA » pour une description de l'utilisation de cette fonction.

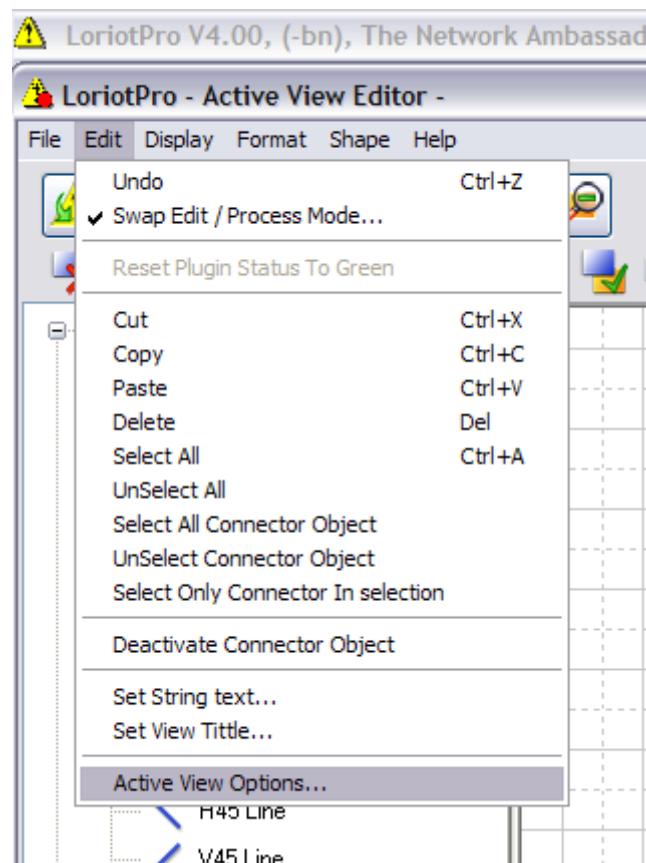
## 12.11 ShortCut associant une ActiveView Modal et un Script LUA

La fonction shortcut associée à un équipement de la Directory permet aussi de lancer une ActiveView Modal qui pourra être configurée pour lancer au démarrage un script qui initialisera l'ActiveView. Dans l'exemple suivant une ActiveView est créée à partir de l'éditeur, précisant le chargement d'un script qui permet la création automatique d'une carte « MAP » de niveau 3 (routeurs - réseaux).



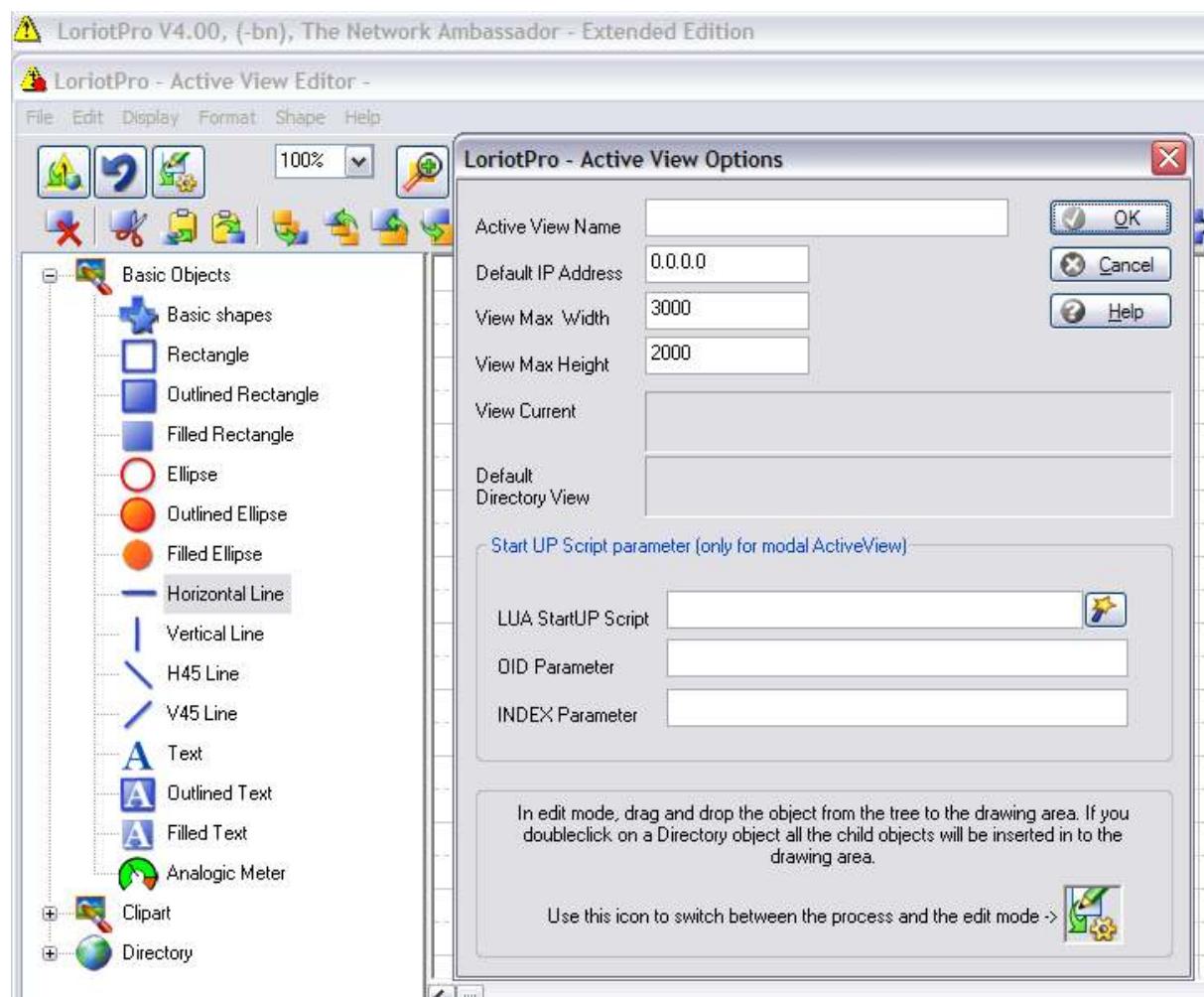
Une fois l'ActiveView ouverte, choisissez l'option :

**Edit>Active View Options**

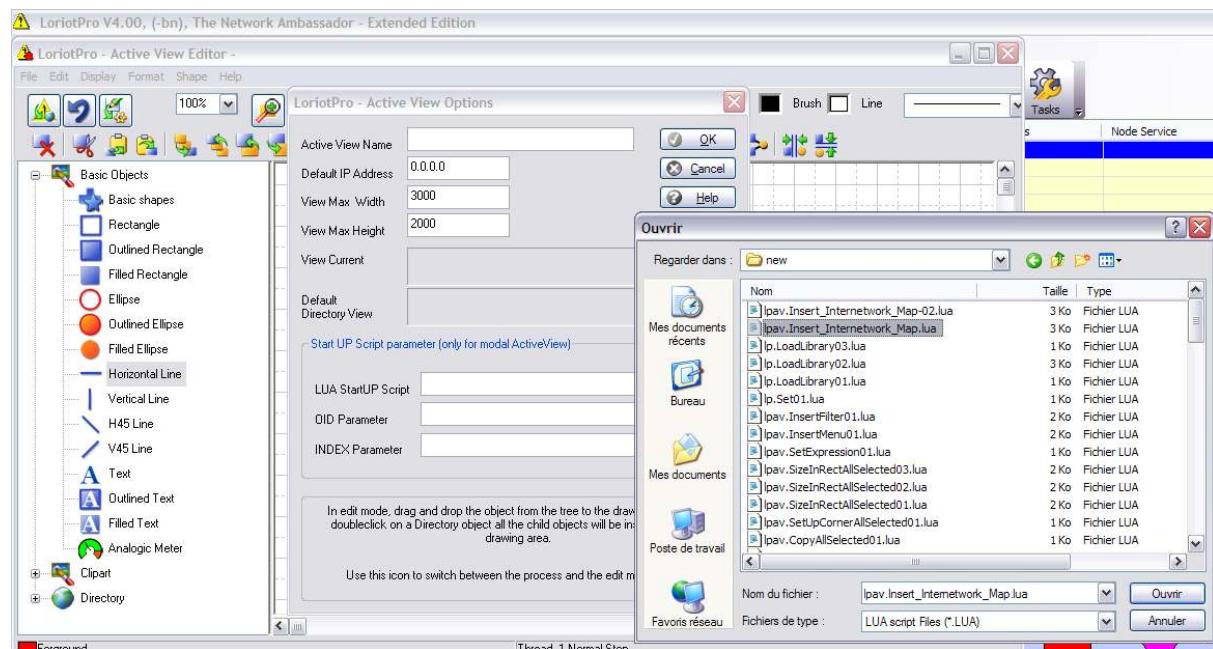


# LoriotPro V5.00 Extended Edition – Supervision et gestion SNMP par scripts LUA

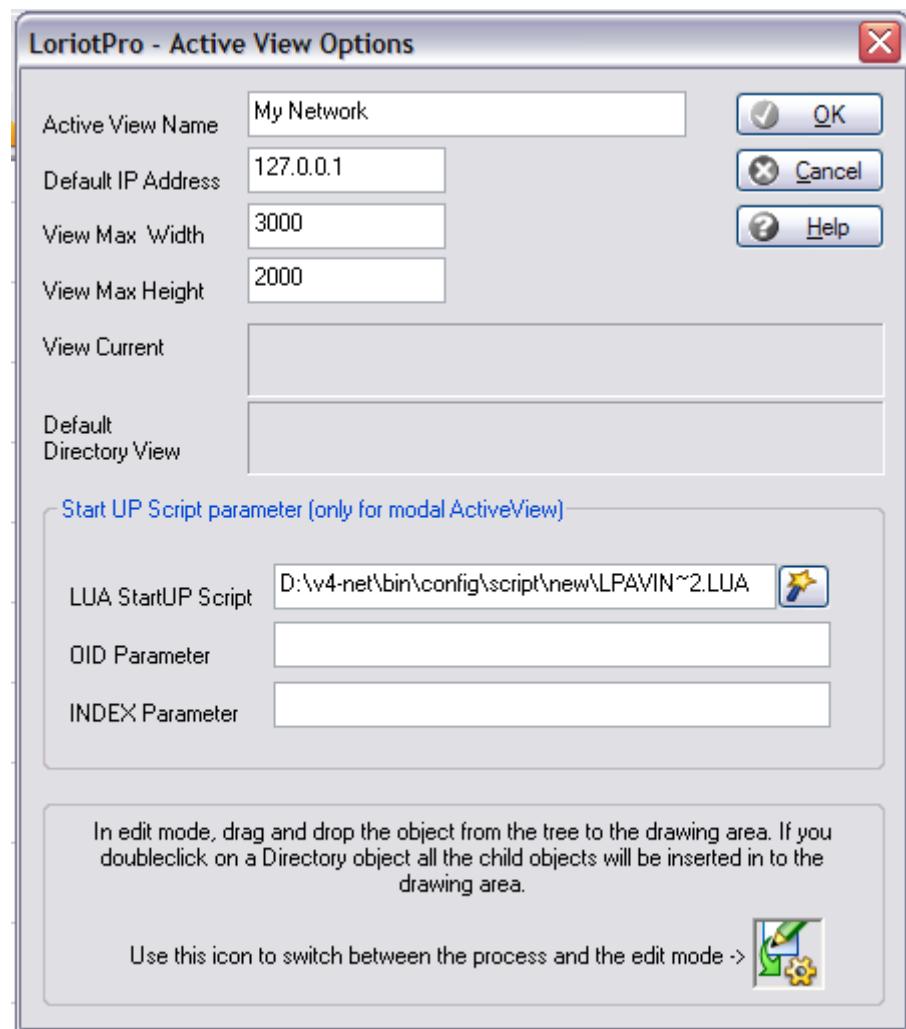
## Introduction au System Expert et aux scripts



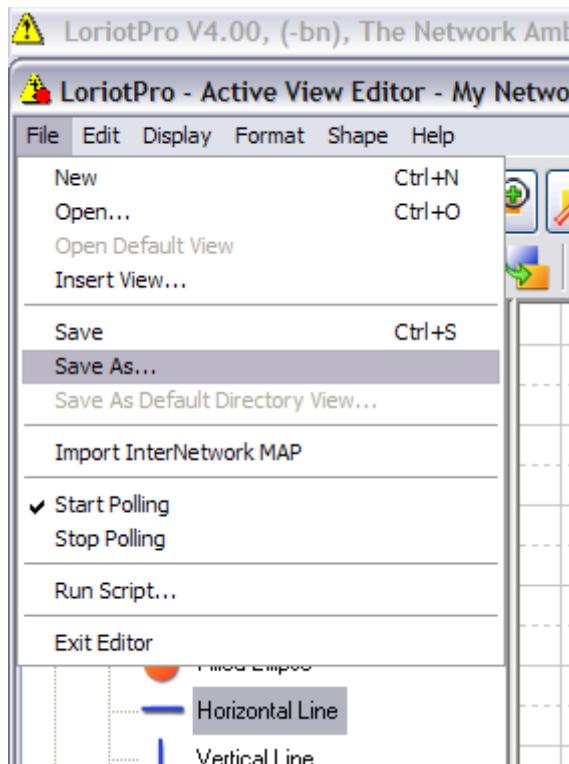
Les options LUA StartUP Script vous permettent de définir un script de démarrage pour votre future ActiveView :



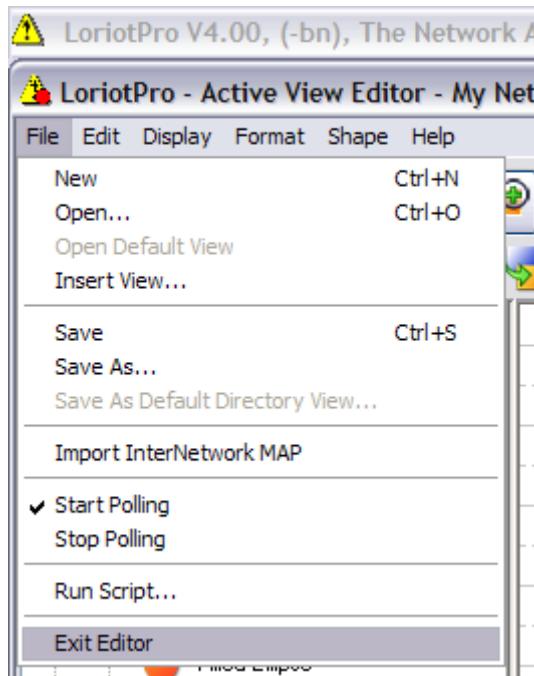
Nous choisissons le script **Ipv4.Insert\_Internetwork\_Map.lua**.



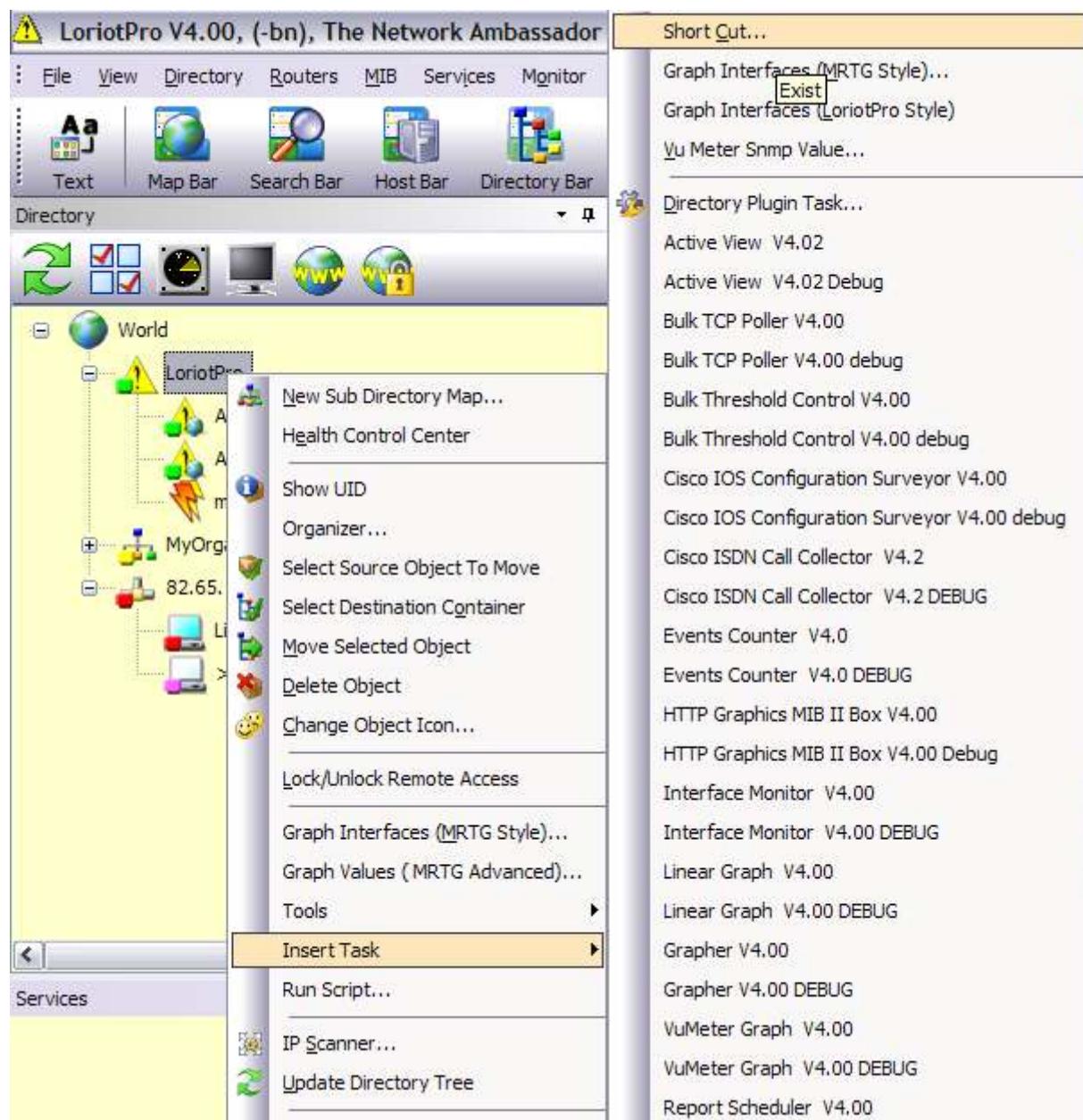
Le nom du fichier est transformé en mode court pour éviter les caractères blancs.



On sauvegarde l'ActiveView sous my\_script\_01.cmp.



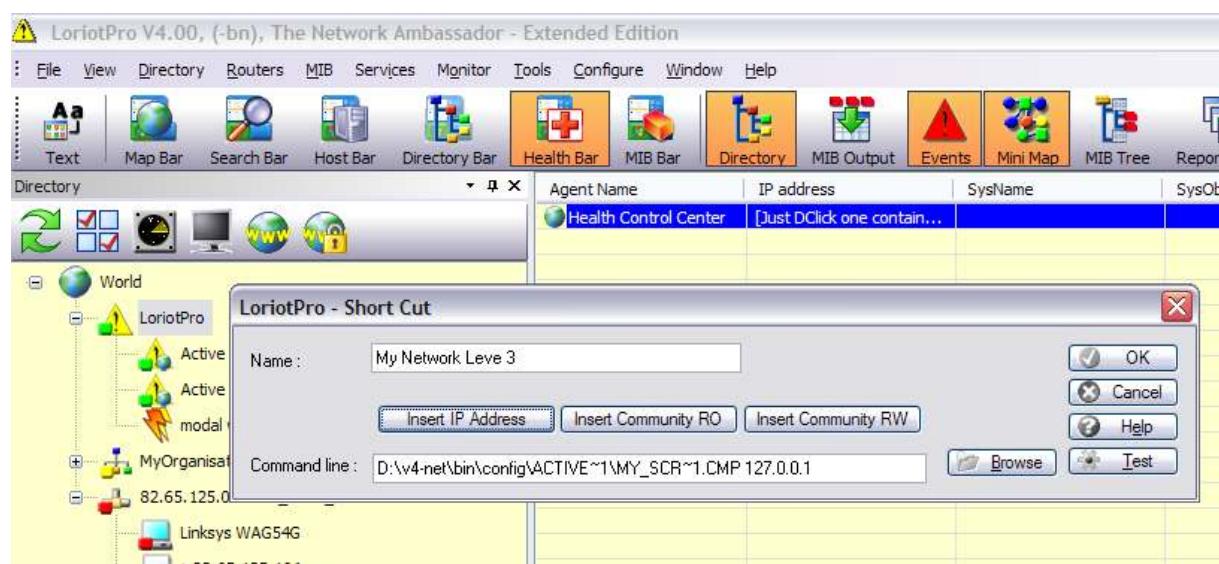
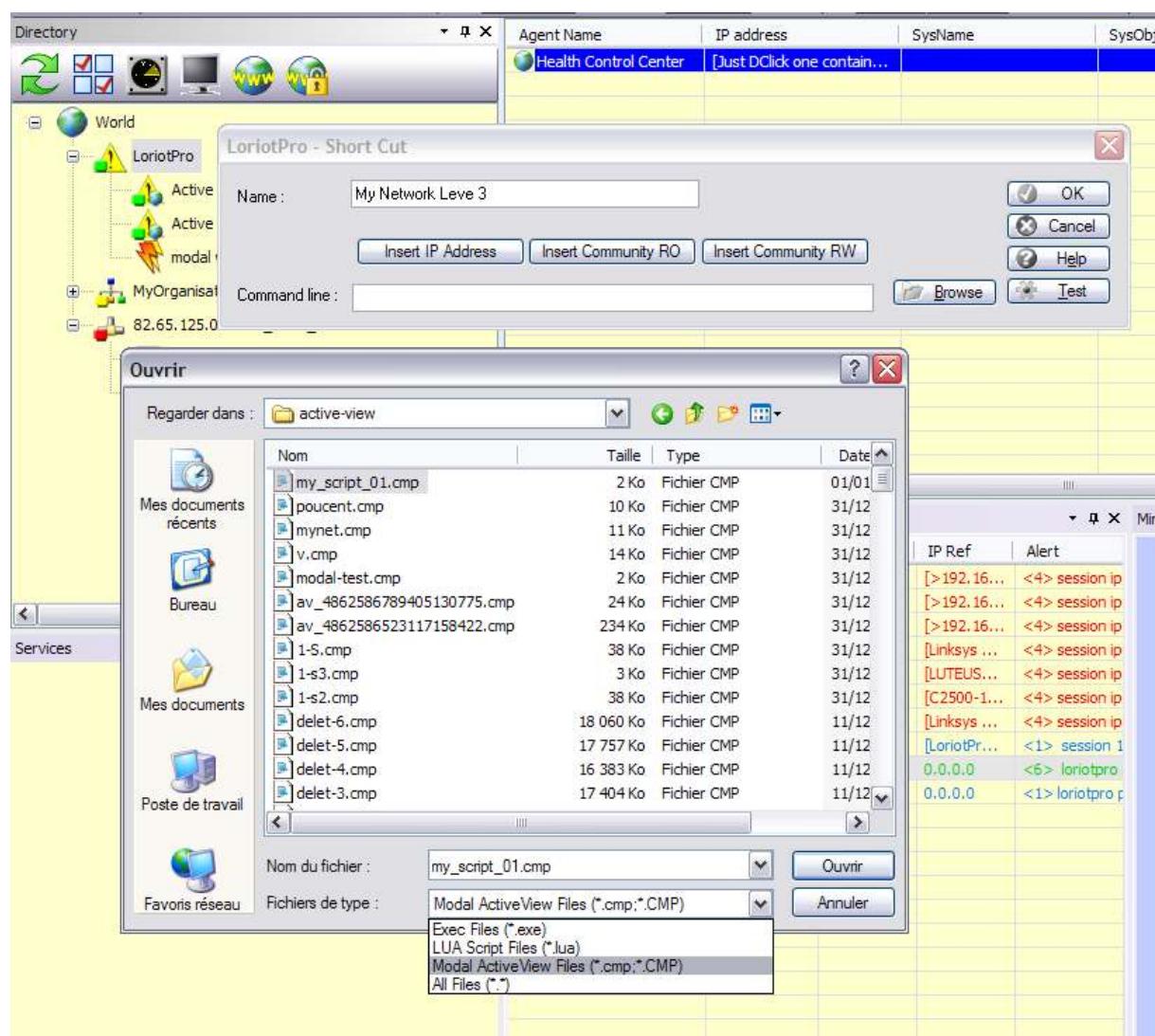
On quitte l'éditeur.



On insère un shortcut sous l'équipement désiré, dans notre cas le 127.0.0.1.

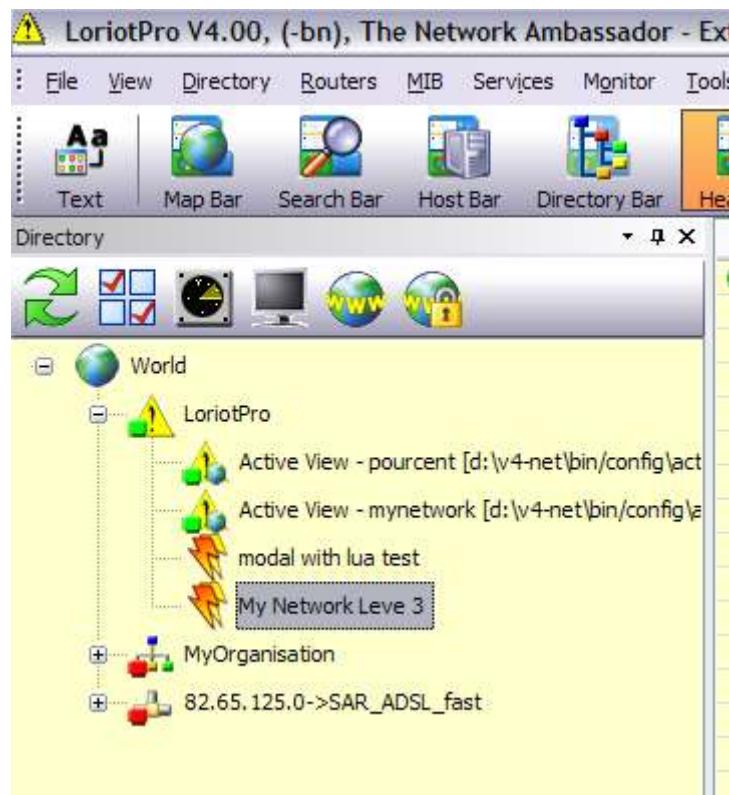
# LoriotPro V5.00 Extended Edition – Supervision et gestion SNMP par scripts LUA

## Introduction au System Expert et aux scripts



**Attention** 

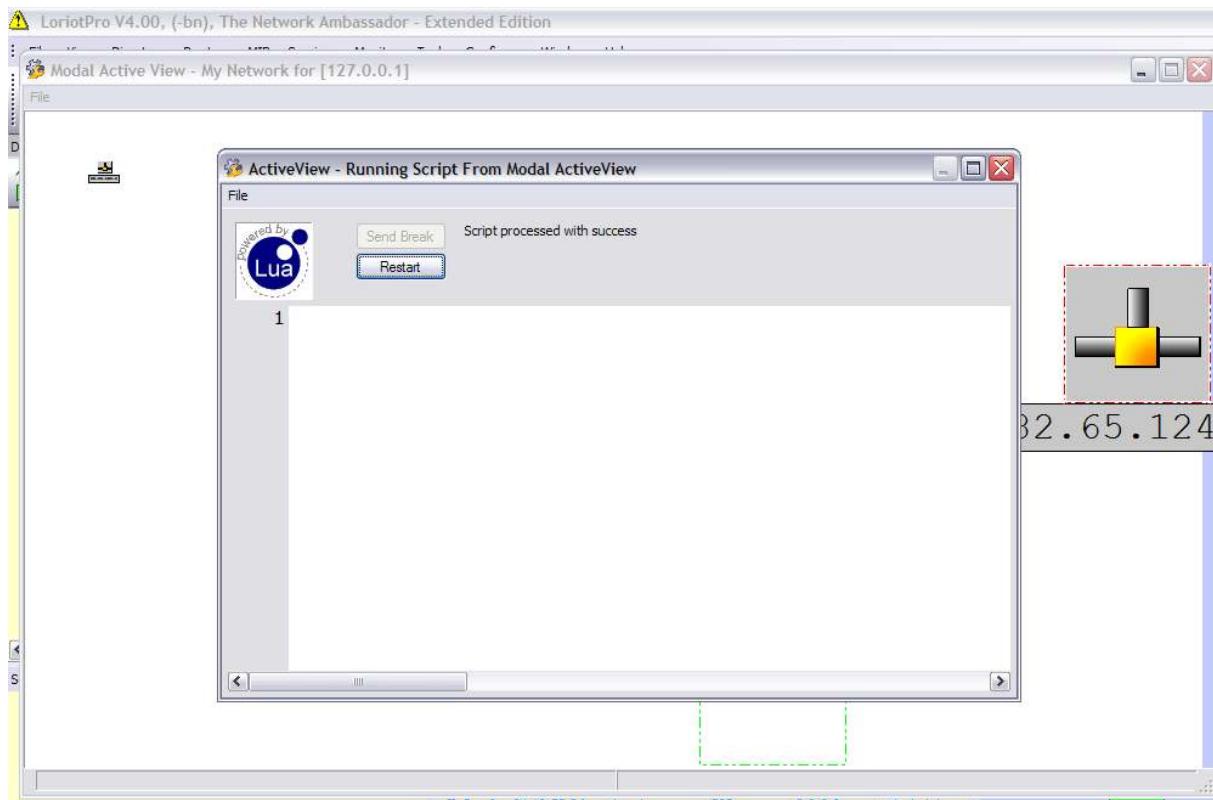
Le nom du fichier est mis en mode court et il faut impérativement mettre l'adresse IP désirée en deuxième paramètre.



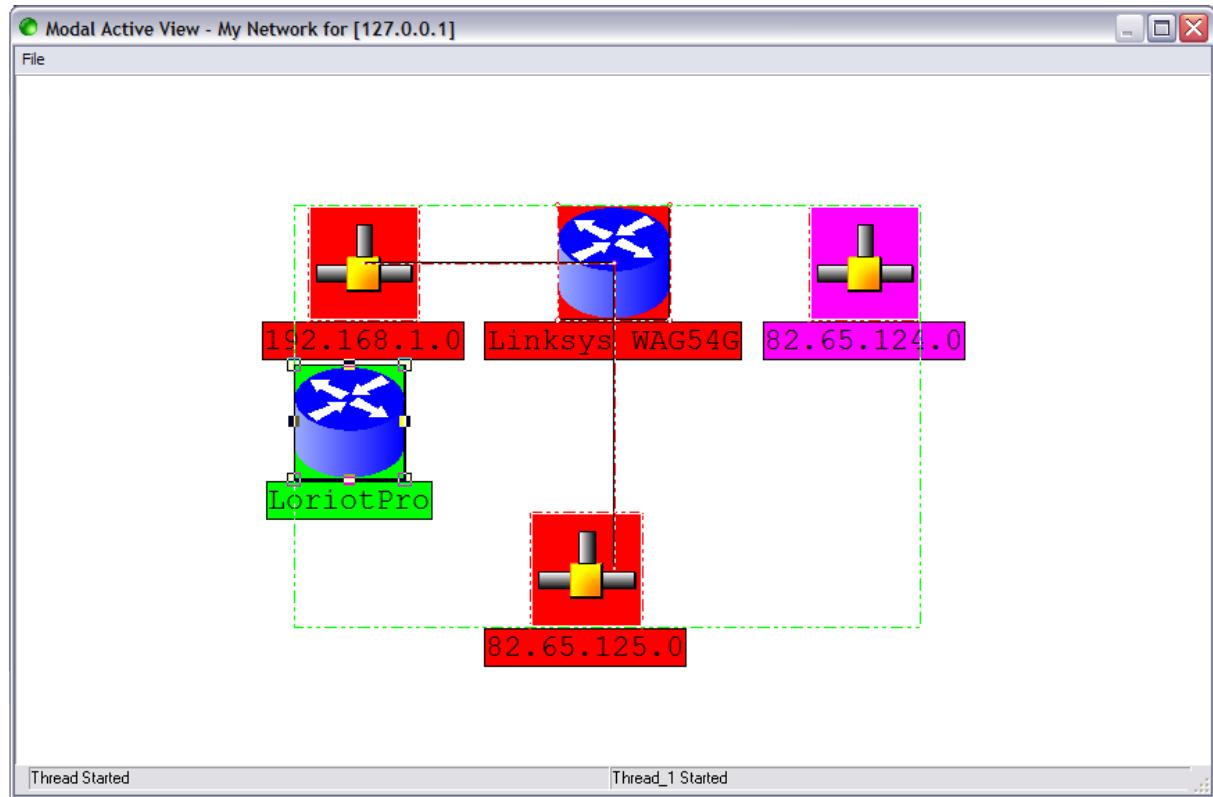
Dans notre exemple, un nouveau ShortCut est créé sous l'équipement LoriotPro, qui représente la machine locale où est installé le logiciel.

Si vous double cliquez sur l'Item « My Network Level 3 » L'ActiveView **my\_script\_01.cmp** est affichée et le script de démarrage **lpav.Insert\_Internetwork\_Map.lua** associé est lancé.

LoriotPro V5.00 Extended Edition – Supervision et gestion SNMP par scripts LUA  
Introduction au System Expert et aux scripts

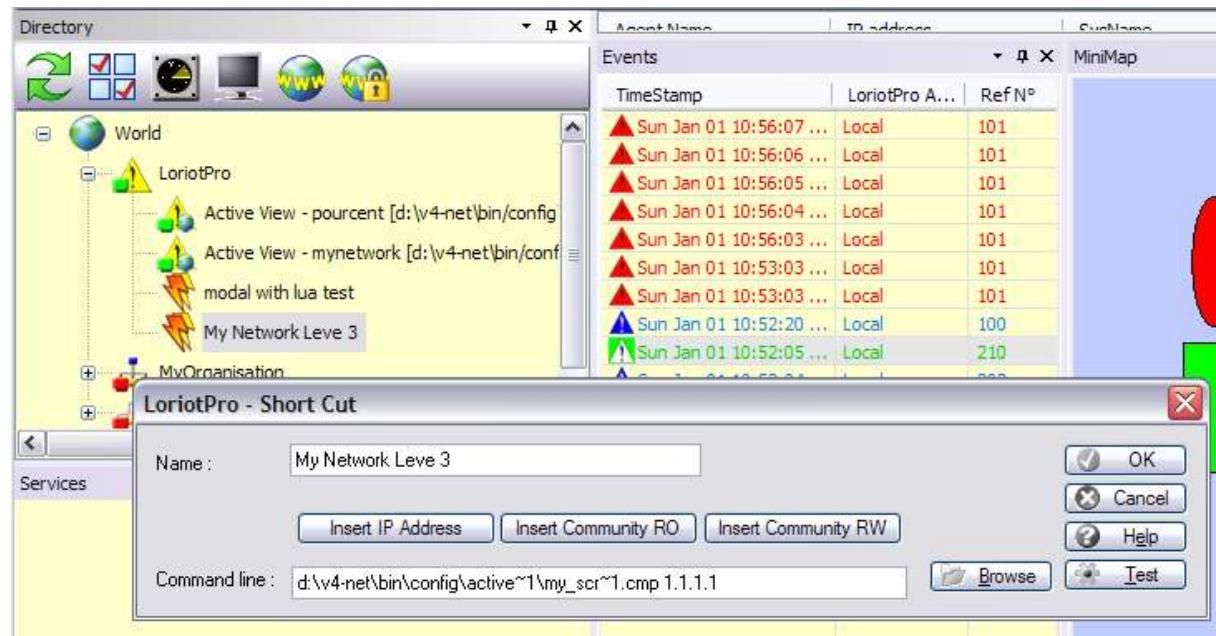


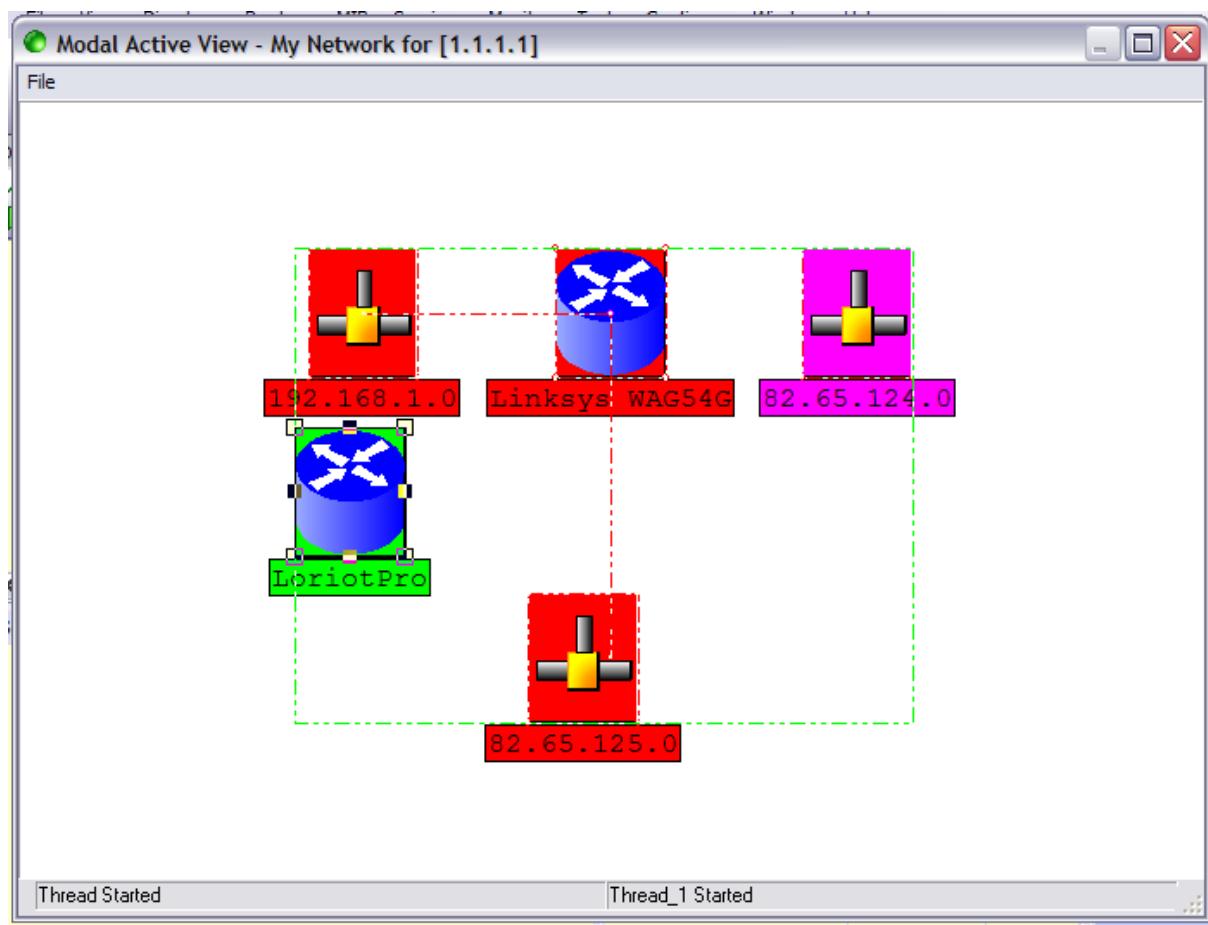
La fenêtre de script est automatiquement fermé à la fin du processus pour laisser la place à l'ActiveView.



**Attention** 

L'adresse IP positionnée dans le champ de paramètre du ShortCut est utilisée comme adresse par défaut pour l'ActiveView, il est donc possible par exemple de lancer une ActiveView pour l'adresse IP 1.1.1.1 sous l'équipement 127.0.0.1.

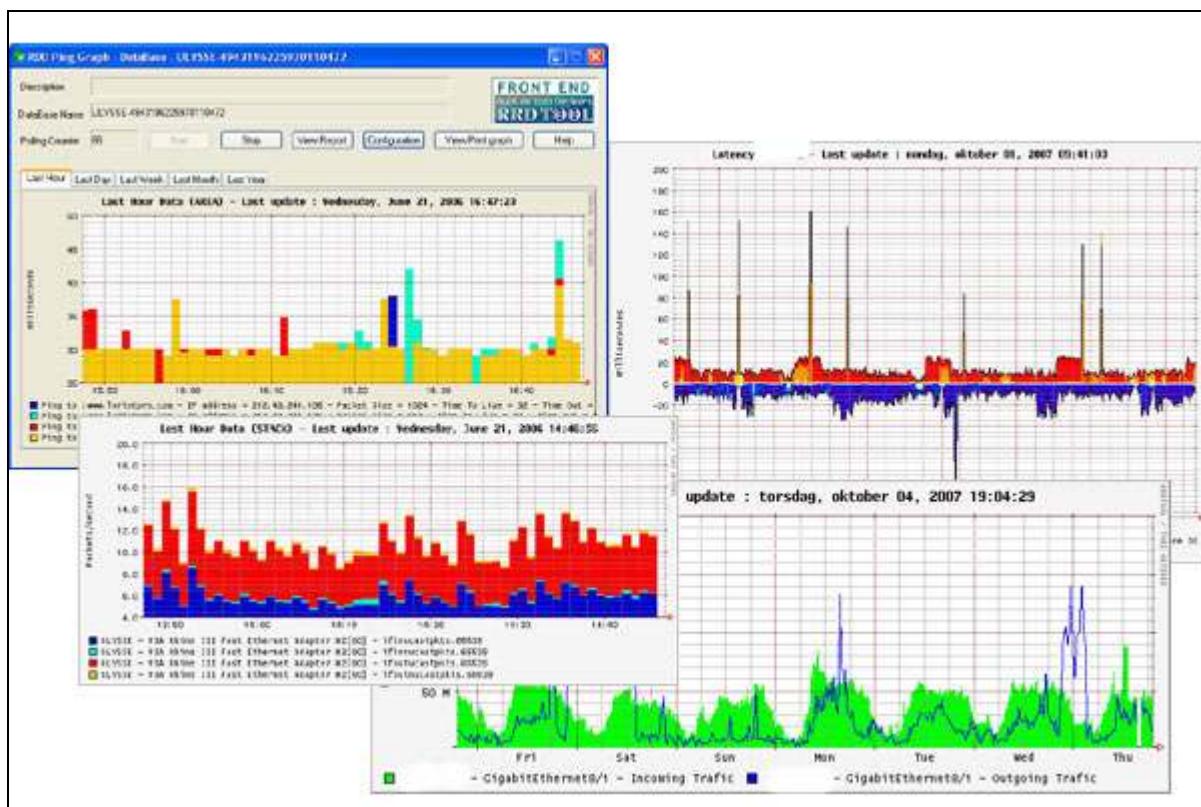




## 13 Graphe et Courbe

### 13.1 Introduction

LoriotPro intègre plusieurs modules permettant la réalisation du graphe de valeurs. Traditionnellement dans les NMS classiques, les valeurs graphées sont générées à partir de collectes d'objets SNMP simples sur l'équipement. LoriotPro vous permet de réaliser ce type d'opérations traditionnelles à travers plusieurs modules couvrant les différents types de besoin en terme de types de graphes. Pour répondre aux objectifs définis au début de ce document, LoriotPro Extended Edition intègre dans son système expert un module permettant d'obtenir des graphes directement à partir de script. Ce module crée des graphes couvrant tout type de logique, le graphe peut représenter une corrélation de valeurs ou bien un résultat mathématique ou logique.



Le module permettant ce type d'opération est le plus puissant actuellement disponible avec LoriotPro ; il se sert du moteur RRD, utilisé et reconnu mondialement pour le rendu et les capacités de ses graphes multi-courbes. Le système RRD intègre une logique de base de données qui rend possible le tracé de graphes en tendance sur de longues périodes. La solution LoriotPro s'appuie sur deux modules :

**RRD collector** ; pour la collecte et la visualisation des données associées à un équipement. RRD Collector est un plugin intégré à la Directory et placé sous un équipement.

**RRD Manager** ; pour la gestion des bases de données RRD et la réalisation de rapports permettant la fusion des données collectées par les modules RRD Collector.

RRD Manager est un plugin de service, il fédère l'ensemble des plugins RRD collector lancés dans la Directory.

La documentation complète de 'RRD Collector' et 'RRD Manager' dépasse le cadre de ce document qui est consacré uniquement à la partie Système Expert de LoriotPro.

Le lien suivant vous fournira l'ensemble de la documentation de ces modules.

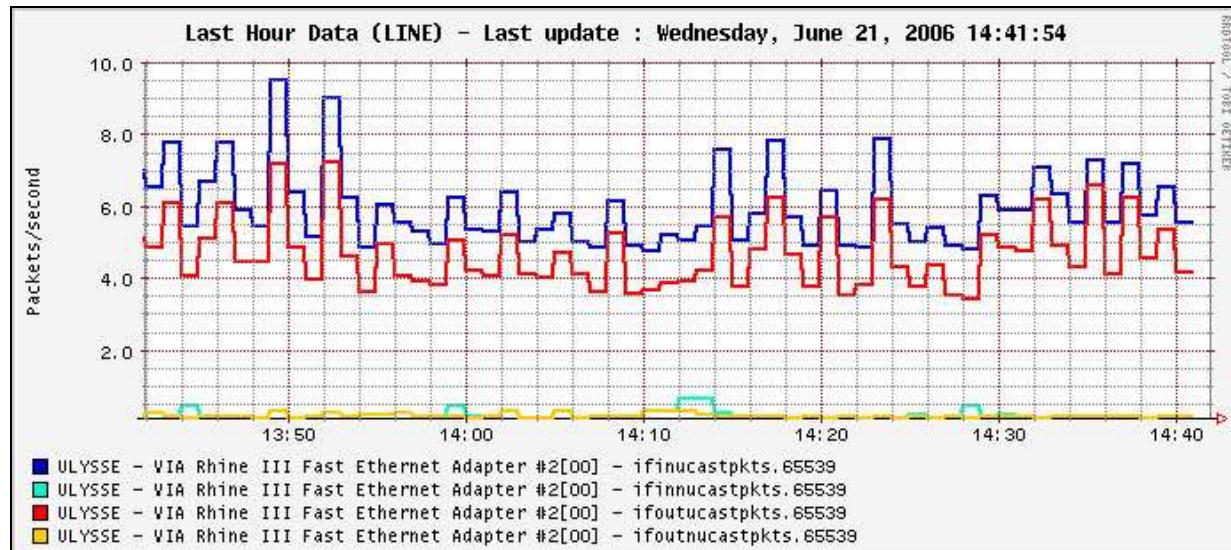
[http://www.luteus.biz/Download/LoriotPro\\_Doc/RRD%20documentation/RRD\\_Collector\\_RRD\\_Manager\\_documentation\\_EN.htm](http://www.luteus.biz/Download/LoriotPro_Doc/RRD%20documentation/RRD_Collector_RRD_Manager_documentation_EN.htm)

Nous ne traiterons ici que de la partie scripté de RRD Collector.

## 13.2 Introduction à RRD

Les modules (plugins) RRD Manager et RRD Collector disponibles avec LoriotPro Extended Edition, utilisent le logiciel RRD tool disponible en licence GNU. Le concept et le développement de RRD sont l'œuvre de Tobias Oetiker, administrateur système du Swiss Federal Institute of Technology.

Le principal objectif de l'outil RRD est la création de graphiques temporels. Un graphique temporel présente en abscisse le temps et en ordonnée une variable quelconque. Dans LoriotPro, cette variable sera issue de la collecte de valeurs sur des objets snmp, sur des temps de réponse d'application issus de ping, ou sur toute autre valeur issue de capteurs divers.



Les graphiques RRD, pour un jeu de données, sont construits à partir de données enregistrées dans une base. Cette base est un simple fichier texte qui possède une structure simple mais particulière, issue de l'ingéniosité de son auteur Tobi Oetiker.

RRD est l'abréviation de Round Robin Database mais RRD ne se contente pas de définir la structure de la base, il permet surtout de définir des graphiques temporels à partir des données issues de cette base.

Le principe de RRD s'appuie en premier sur la consolidation des données collectées. Vous voulez par exemple collecter des données avec un intervalle de une minute pour avoir des graphiques temporels très détaillés mais vous êtes aussi intéressé par des graphiques en tendance sur une période annuelle par exemple.

Une méthode classique consisterait à collecter et à conserver toutes les données dans une base. Cette méthode atteint très vite des limites par la taille disque utilisé et par le temps de traitement nécessaire pour récupérer les données. Une année d'échantillon pour une seule courbe dans un graphique temporel représente 5256000 enregistrements.

La consolidation des données est le principe clé utilisé par RRD ; à la mise en place d'une collecte et d'une base RRD, il faut choisir un intervalle, fréquence de la collecte, ensuite les données collectées seront consolidées sur un intervalle de temps égal ou supérieur. La consolidation est le résultat d'un calcul mathématique, fonction de consolidation, parmi les suivantes : moyenne arithmétique des échantillons, échantillon le plus petit, échantillon le plus grand, somme des échantillons, dernier échantillon.

La mise en place d'une consolidation RRD consiste donc à définir :

1. Un intervalle de consolidation : période de temps sur laquelle sera appliquée la fonction de consolidation
2. Une fonction de consolidation indiquant le traitement à appliquer sur les données collectées.
3. Le nombre d'échantillons maximum à collecter : taille de l'historique déterminant la taille d'espace disque nécessaire

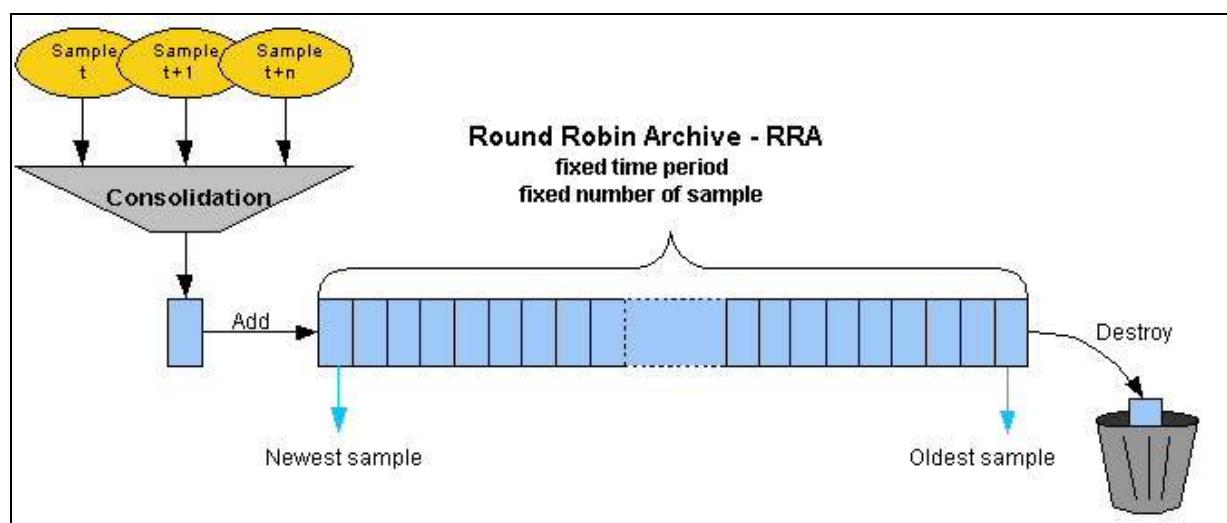
Les données résultant d'une même consolidation sont enregistrées dans la base RRD dans une « archive » en mode round robin.

Les archives RRA, comme le nom l'indique, contiennent généralement une vision historique et en tendance de l'évolution de variables, un trafic réseau par exemple.

Une archive round robin ( RRA ) a une taille fixe dépendante du jeu de données collectées, de la période de temps ( heures, jour , mois, année ..) et de l'intervalle entre chaque échantillons.

Le terme « Round Robin » définit la notion de « tour de rôle » car les données les plus anciennes sont systématiquement remplacées par les données les plus récentes.

Cette faculté permet d'avoir des archives de taille fixe, ce qui facilite grandement la maintenance de la solution, en comparaison avec des solutions en bases de données plus classiques.



En revanche, on perd de la granularité et de la précision car il est habituellement impossible de retrouver par exemple la valeur d'un échantillon initial de la collecte plusieurs semaines ou mois après, celle-ci ayant été perdue lors des consolidations.

Consolidation des données pour la création des archives.

La consolidation des données consiste, à partir des données collectées, à calculer les données à mettre dans les archives RRA.

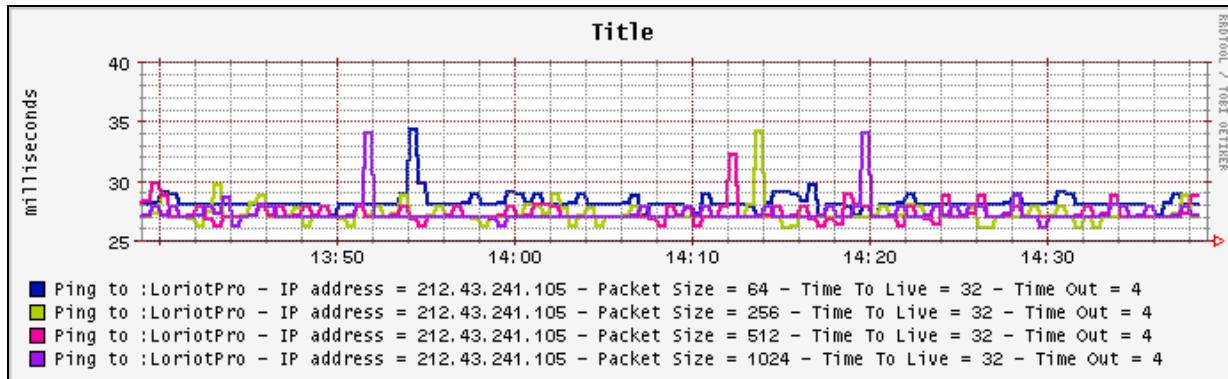
Il existe 4 types de fonctions

AVERAGE	Moyenne	Calcule la moyenne arithmétique des valeurs
LAST	Dernière valeur	Prend la dernière valeur collectée
MIN	Valeur minimale	Prend la plus petite valeur des valeurs collectées
MAX	Valeur maximale	Prend la plus grande valeur des valeurs collectées

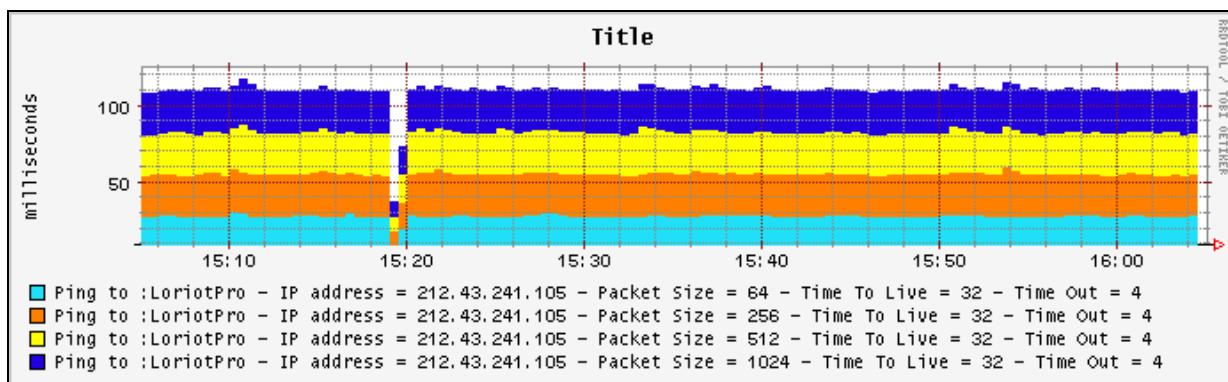
Création de graphiques temporels et de rapport avec RRD

Les données enregistrées dans les bases RRD peuvent être exploitées pour créer des graphiques temporels et des rapports. La présentation des graphiques est entièrement configurable, le type et le format des graphiques, le nombre de courbes qu'il contient, leur couleur, les légendes et les échelles etc.

Exemple de graphique sur temps de réponse des courbes de type ligne



Exemple de graphiques avec courbes de type surface et empilées.



Les caractéristiques suivantes des graphiques peuvent être personnalisés :

La hauteur de la zone du graphique, axe Y

La largeur de la zone du graphique, axe X

La légende de l'axe des Y

Le titre du graphique situé au dessus du graphique, celui-ci est optionnel et peut être supprimé du graphique

Le type d'échelle pour l'axe des Y, soit linéaire soit logarithmique

La couleur des courbes du graphique

Le mode d'affichage des courbes, soit ligne, soit en surface pleine. Les courbes peuvent aussi être empilées sur la première courbe définie en tant que ligne ou surface.

### 13.3 Implémentation de RRD dans LoriotPro

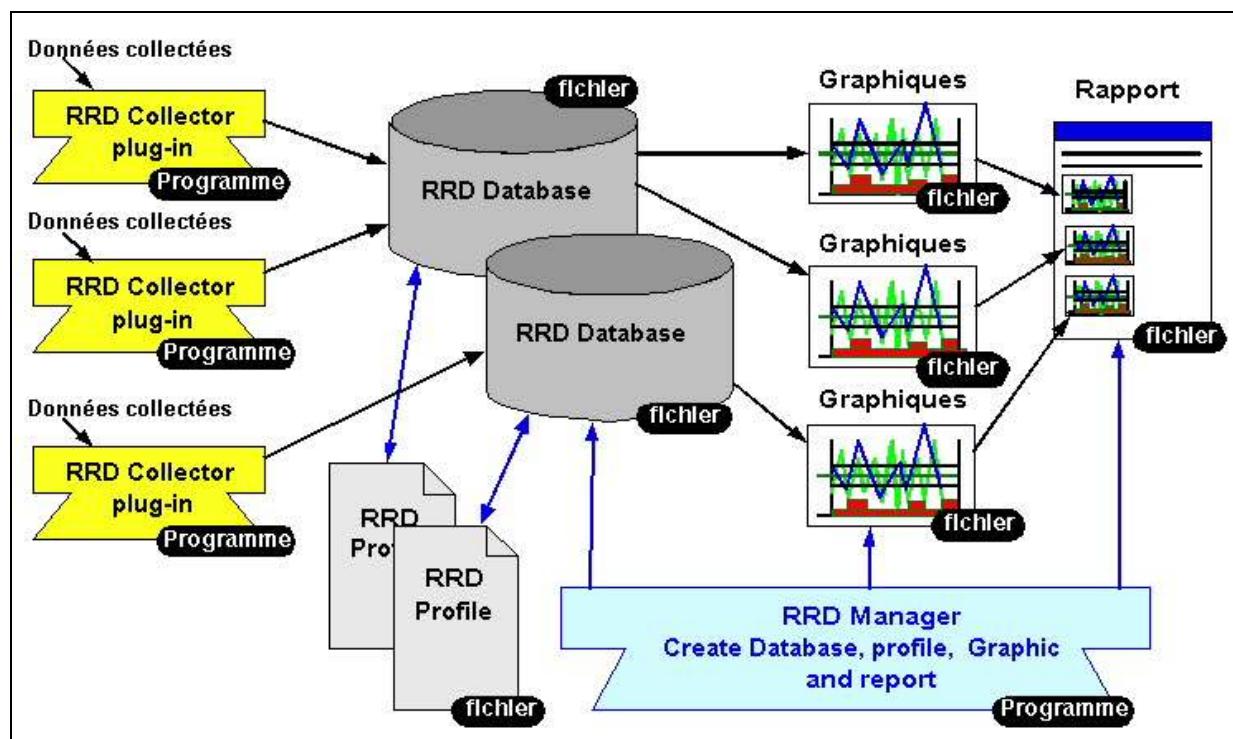
LoriotPro utilise les programmes de RRD pour la génération des bases et des graphiques. Les programmes propres à LUTEUS viennent compléter l'offre RRD pour en faciliter l'utilisation au sein de LoriotPro.

On distingue deux modules :

RRD Collector en charge de la collecte des données et de leur enregistrement dans les bases RRD.

RRD Manager dédié aux administrateurs avertis qui souhaitent créer des graphiques non standard ou personnaliser des graphiques créés par RRD Collector.

Le graphique suivant représente l'ensemble des briques constituant l'architecture de collecte et de management autour de RRD.



Cette architecture permet en finale de créer des rapports en format html qui contiennent plusieurs graphiques ainsi que leur description.

A la source des données, on distingue le module RRD Collector (Plug-in de diretctory LoriotPro). Il est attaché à un équipement de la Directory de LoriotPro.

Dans le cas de collecte SNMP, il collecte des informations propres à cet équipement.

Les données sont issues d'une source de données, une requête snmp sur un objet snmp dans un équipement réseau ou sur un serveur par exemple.

Les requêtes snmp sont réalisées à intervalles réguliers pour la construction de l'historique.

Les données collectées sont ensuite enregistrées dans une base RRD. Une source de données ne peut être enregistrée que dans une base RRD et une seule. Par contre deux RRD Collector peuvent enregistrer des sources de données dans une même base. Une base peut contenir en effet plusieurs sources de données.

Les profils RRD permettent de définir pour une liste de source de données, un jeu d'archives associées. Les archives sont des jeux de données issues des sources de données sur lesquelles ont été appliquées des fonction de consolidation.

A partir des données consolidées, les archives RRA, il est possible de générer des graphiques. Ce sont des fichiers en format gif ou png qui pourront être visualisés directement ou insérées dans des rapports en format WEB html.

## 13.4 Licence RRD

```
RRDTOOL - Round Robin Database Tool
A tool for fast logging of numerical data graphical display
of this data.

Copyright (c) 1998, 1999 Tobias Oetiker
All rights reserved.

This program is free software; you can redistribute it and/or modify it
under the terms of the GNU General Public License as published by the Free
Software Foundation; either version 2 of the License, or (at your option)
any later version.

This program is distributed in the hope that it will be useful, but WITHOUT
ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or
FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for
more details.

You should have received a copy of the GNU General Public License along
with this program; if not, write to the Free Software Foundation, Inc.,
59 Temple Place - Suite 330, Boston, MA 02111-1307, USA
```

## 13.5 Introduction à RRD Collector

RRD Collector est le module (Directory plugin) de collecte des données d'enrichissement de bases de données RRD et de génération de graphiques RRD. La collecte des données consiste soit à réaliser des requêtes SNMP sur un équipement particulier, soit à réaliser des requêtes PING pour mesurer des temps de traversée réseaux (Round trip Time).

Actuellement RRD Collector dispose de trois modes différents de configuration.

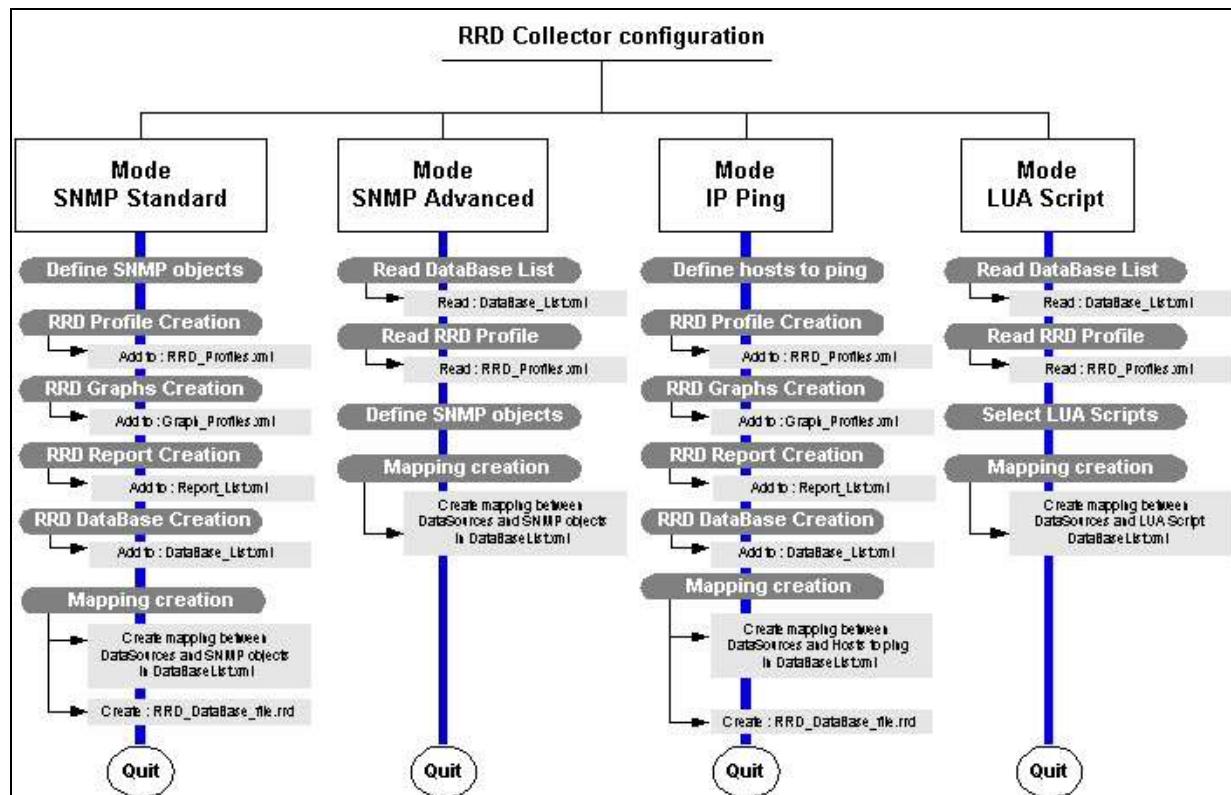
Mode SNMP standard : il consiste à sélectionner par un assistant graphique, un jeu d'objets SNMP pour la création de 5 graphiques, représentant les tendances pour les périodes suivantes, l'heure, le jour, la semaine, le mois et l'année en cours.

Mode SNMP Advanced : ce mode s'appuie sur l'utilisation d'une base RRD créée avec RRD Manager. Ce mode consiste à associer les sources de données (DS) disponibles dans la base sélectionnée à des objets SNMP que l'on sélectionne à partir de l'arbre des MIB. Il nécessite une compréhension complète des concepts de RRD mais offre en revanche une totale liberté dans la création des bases, des graphiques et des rapports.

Mode PING : il consiste à sélectionner un jeu d'adresses IP pour la création de 5 graphiques, représentant les tendances pour les périodes suivantes, l'heure, le jour, la semaine, le mois et l'année en cours. Les adresses IP peuvent être acquises à partir de nom DNS (ex : www.loriotpro.com)

RRD Collector est autonome pour les options 1 et 3 mais dépend totalement du RRD Manager pour le mode SNMP Advanced et LUA Script.

Le diagramme suivant montre les différents modes de configurations disponibles dans RRD Collector.



En mode SNMP standard et IP Ping, les fichiers de configurations et la base de données RRD sont créés automatiquement. Ce sont les modes les plus simples à configurer et qui nécessitent peu de connaissances dans les concepts de RRD. En revanche, ils offrent peu de liberté dans le choix des graphiques, de leur visuel et des données collectées.

En mode SNMP advanced et LUA script, la configuration consiste à utiliser une base de données RRD et des jeux de graphiques déjà définis à l'aide de RRD manager.

Cette méthode de configuration nécessite une connaissance approfondie des concepts de RRD et de l'utilisation de RRD manager mais offre une totale liberté dans la création des graphiques et des bases de données.

### 13.5.1 Principe d'exécution des scripts

Le module 'RRD Collector' exécute un script en lui passant trois variables globales :

**lp\_host** : l'adresse IP de l'équipement concerné

**lp\_oid** : une variable globale

**lp\_index** : une variable globale

il attend en retour deux variables globales traitées par le module :

**lp\_value** : la valeur utilisée pour réaliser le graphe

**lp\_buffer** : une chaîne de caractères indiquant une éventuelle erreur.

Ce principe est le même que la plupart des scripts standard de LoriotPro.

### 13.5.2 Exemple de script – Disk Usage

***bin/config/script/RRDScript /Get\_Disk\_Usage.lua***

```
--<SCRIPT_NAME>Disk Usage</SCRIPT_NAME>
--<SCRIPT_DESCRIPTION>lp-oid should be initialized to either ifinoctets or
ifoutoctets</SCRIPT_DESCRIPTION>
-- Loriotpro V4

-- Ce fichier doit impérativement se trouver dans bin/config/script/rrdscript

-- Parametres passés au script
-- lp_index l'index a ajouté au requête Exemple ".1"
-- lp_oid "No SNMP oid required in this script"
-- lp_host l'add ip du host en string "127.0.0.1" required

-- valeurs à retourner impérativement
lp_value = 0
lp_buffer ="error"

get1=("hrstorageused"..lp_index);
get2=("hrstoragesize"..lp_index);

lp_value = (lp.Get(lp_host,get1)/(lp.Get(lp_host,get2)+0.001))*100;
lp.Trace(lp_value);
lp_buffer ="OK"
```

Il est possible de définir un nom et une description pour ce type de script en utilisant des tags au début du test du script.

Pour le nom :

<SCRIPT\_NAME> </SCRIPT\_NAME>

```
--<SCRIPT_NAME>Disk Usage</SCRIPT_NAME>
```

Pour la description :

```
< SCRIPT_DESCRIPTION > < /SCRIPT_DESCRIPTION >
```

```
--<SCRIPT_DESCRIPTION>lp-oid should be initialized to either ifinoctets or  
ifoutoctets</SCRIPT_DESCRIPTION>
```

La structure de « nomage » de ces scripts est légèrement différente des scripts IP Expert et IP Audit car le plugin utilise une syntaxe de type XML.

### 13.5.3 Lancement de RRD Collector

Le module (plugin) RRD Collector est un module enfichable de l'annuaire (Directory LoriotPro). Il peut être attaché de multiple fois à des équipements situés dans cet annuaire.

Dans les modes SNMP standard et SNMP advanced de RRD Collector, l'équipement auquel il s'attache est utilisé pour toutes les requêtes SNMP émises. C'est pourquoi dans ces deux modes, il doit s'attacher à un équipement configuré et fonctionnant en SNMP. Un RRD collector ne peut donc en SNMP collecter des données SNMP que sur un seul équipement IP.

EN mode IP Ping et LUA Script l'équipement auquel est attaché RRD Collector n'a pas d'importance. Il est usuel de les attacher dans ce mode à l'objet (équipement) LoriotPro.

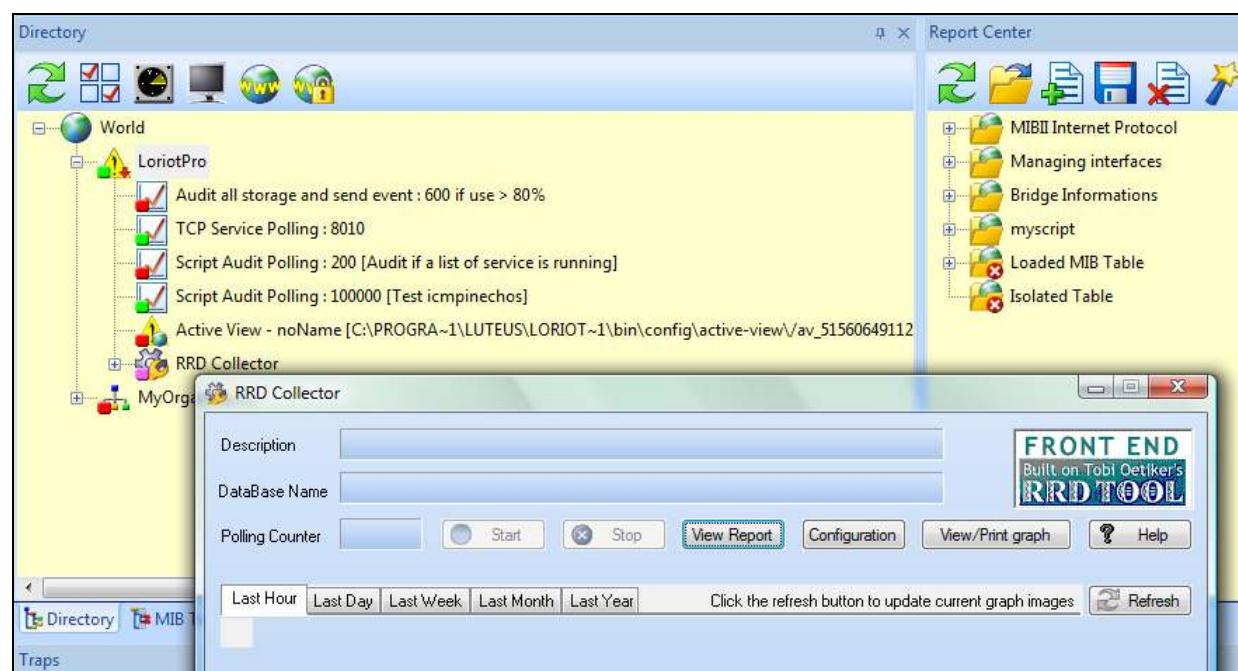
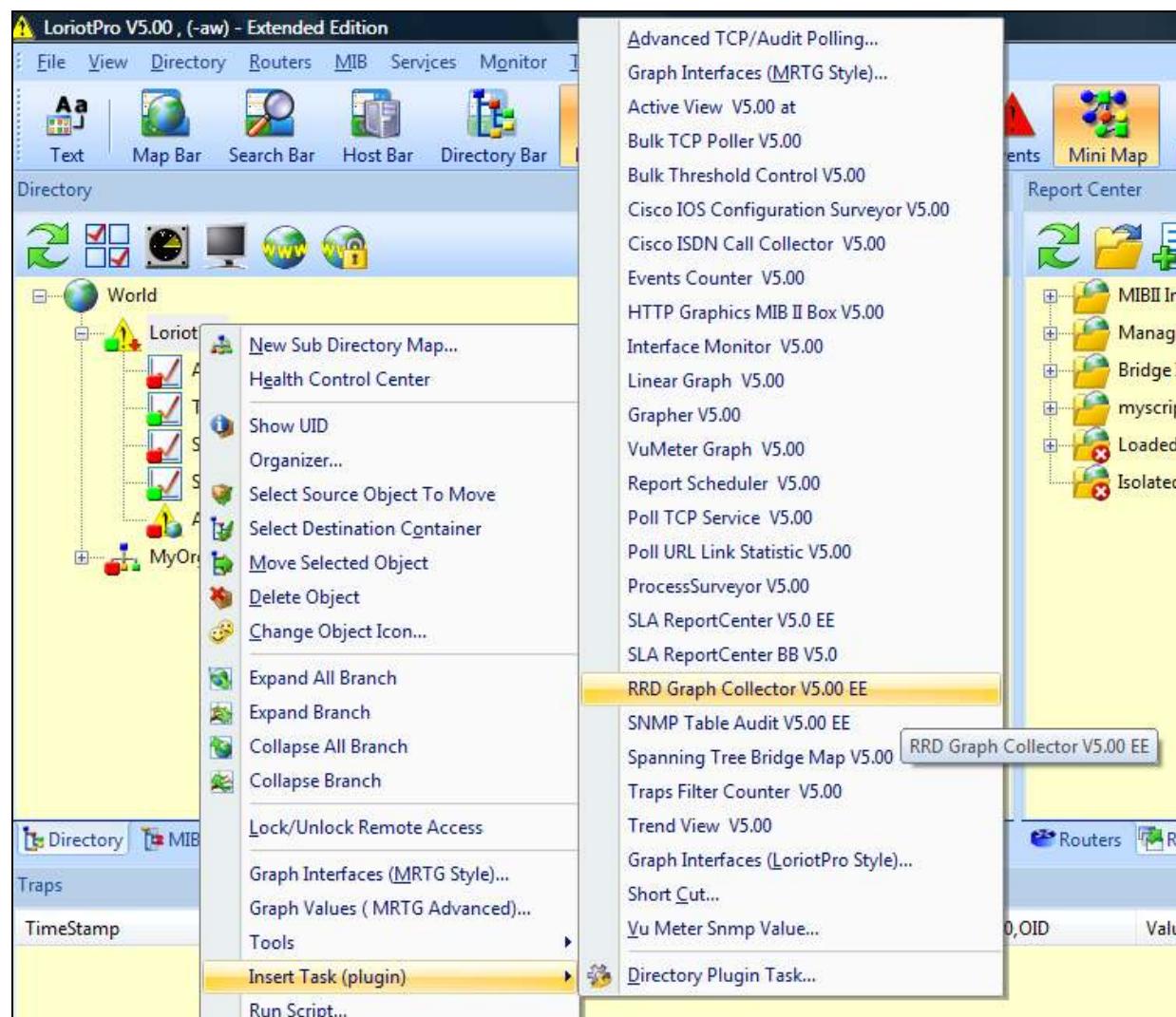
#### 13.5.3.1 Insertion du plugin RRD Collector

Pour attacher un module RRD Collector à un équipement, sélectionnez-le dans l'annuaire puis faites un clic droit de façon à ouvrir le menu contextuel.

Sélectionnez l'option Insert task (plugin) puis, dans la liste des plugins, sélectionnez RRD Collector.

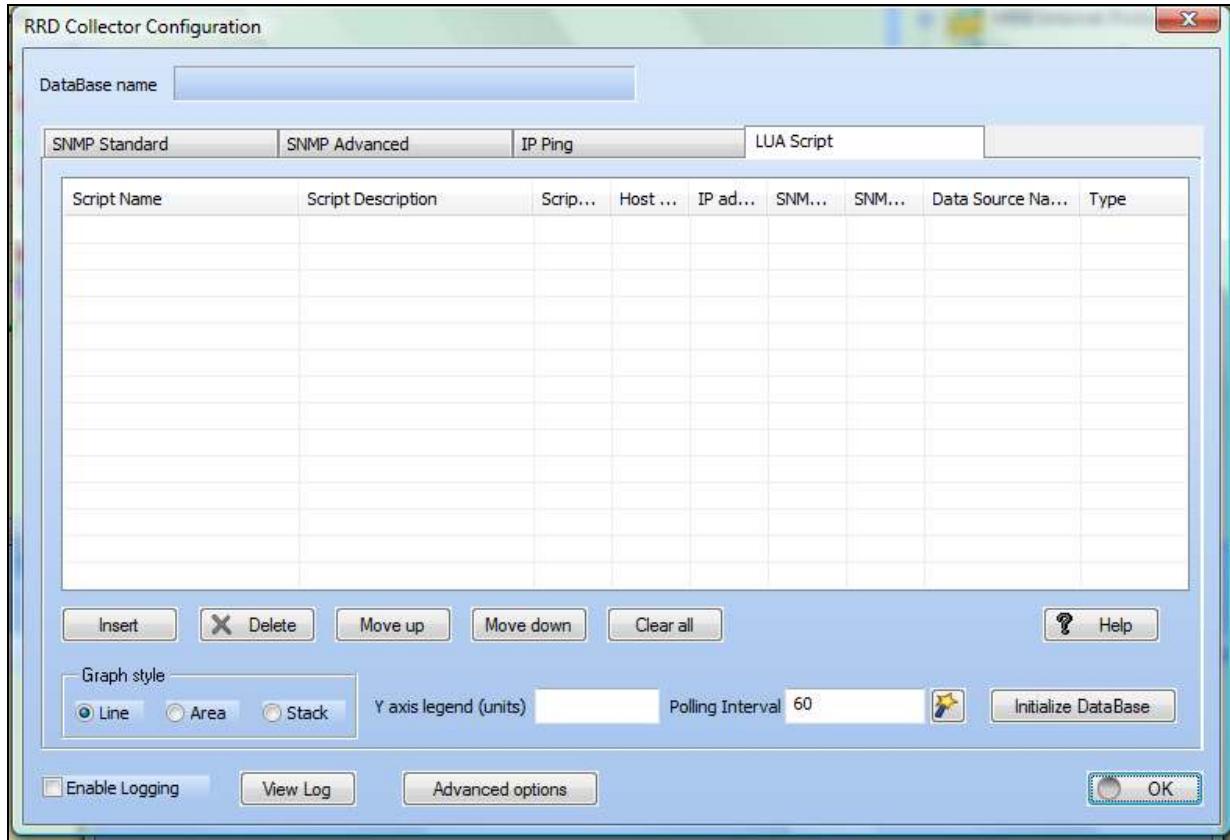
# LoriotPro V5.00 Extended Edition – Supervision et gestion SNMP par scripts LUA

## Introduction au System Expert et aux scripts



### 13.5.3.2 Configuration de la partie Script du plugin

Nous ne traiterons ici que de l'option script du module. Cliquez sur le bouton 'Configuration' puis LUA Script.

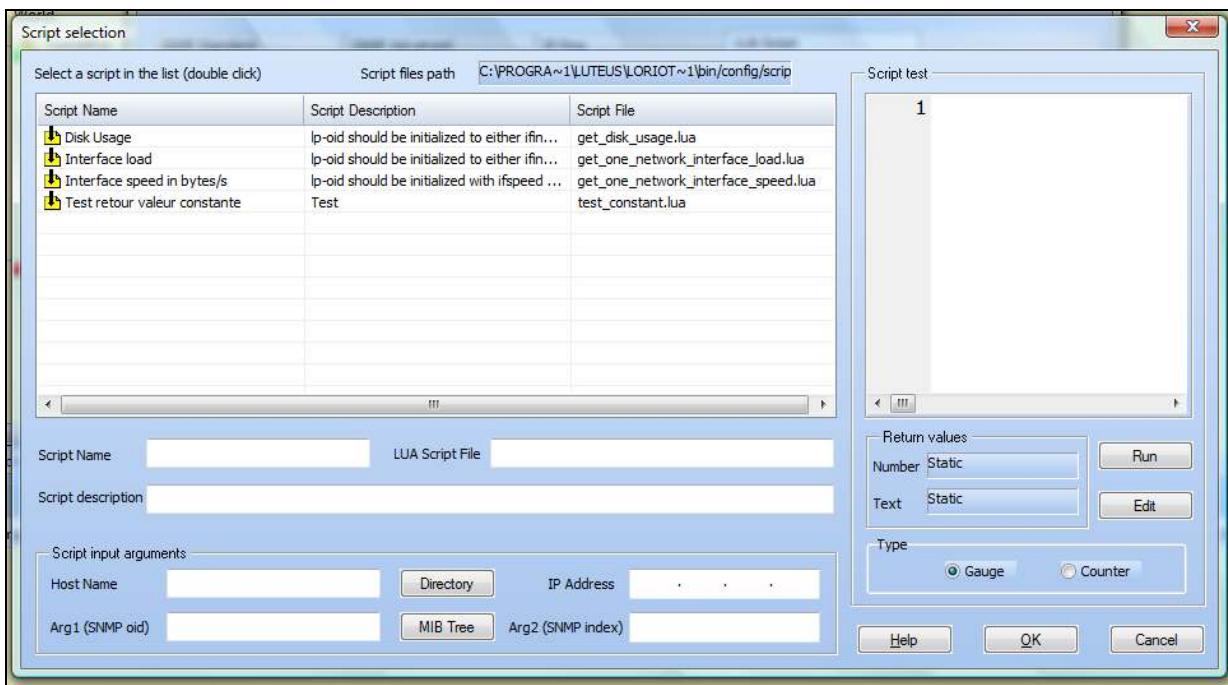


Il faut insérer un ou plusieurs scripts qui réaliseront une collecte d'informations, ces scripts doivent être écrits avant l'insertion du plugin.

#### 13.5.3.2.1 Option – Insert

Le bouton 'Insert' appelle la fenêtre de sélection des fichiers de scripts dédiés au plugin 'RRD Collector'. L'ensemble des fichiers est situé dans le répertoire

***bin/config/script/RRDScript***

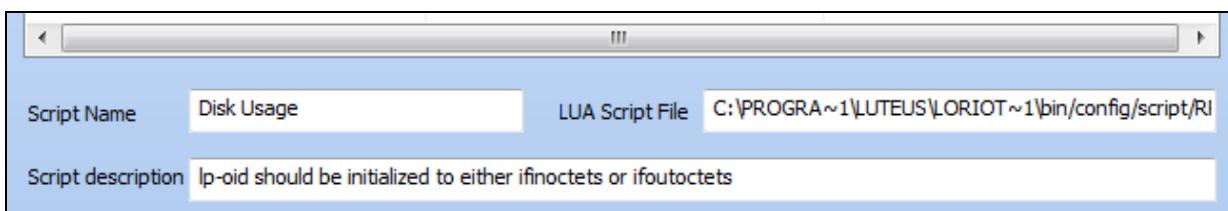


Cette fenêtre de sélection permet de tester et définir les paramètres qui seront associés au script. Il est possible de lancer plusieurs scripts pour réaliser plusieurs courbes dans votre graphe. Le module permet donc de définir une adresse IP par script et des paramètres indépendants.

#### Attention

L'adresse IP utilisée n'est pas forcément l'adresse IP de l'équipement auquel est attaché le plugin.

Nous double cliquons sur le script : Disk usage pour notre exemple.



#### 13.5.3.2.1.1 Option – Script Name

Ce champ est automatiquement renseigné lorsque vous sélectionnez un script dans la listbox.

#### 13.5.3.2.1.2 Option – LUA Script File

Ce champ est automatiquement renseigné lorsque vous sélectionnez un script dans la listbox.

Le fichier est obligatoirement dans : **bin/config/script/RRDScript**

#### 13.5.3.2.1.3 Option – Script description

Ce champ est automatiquement renseigné lorsque vous sélectionnez un script dans la listbox.

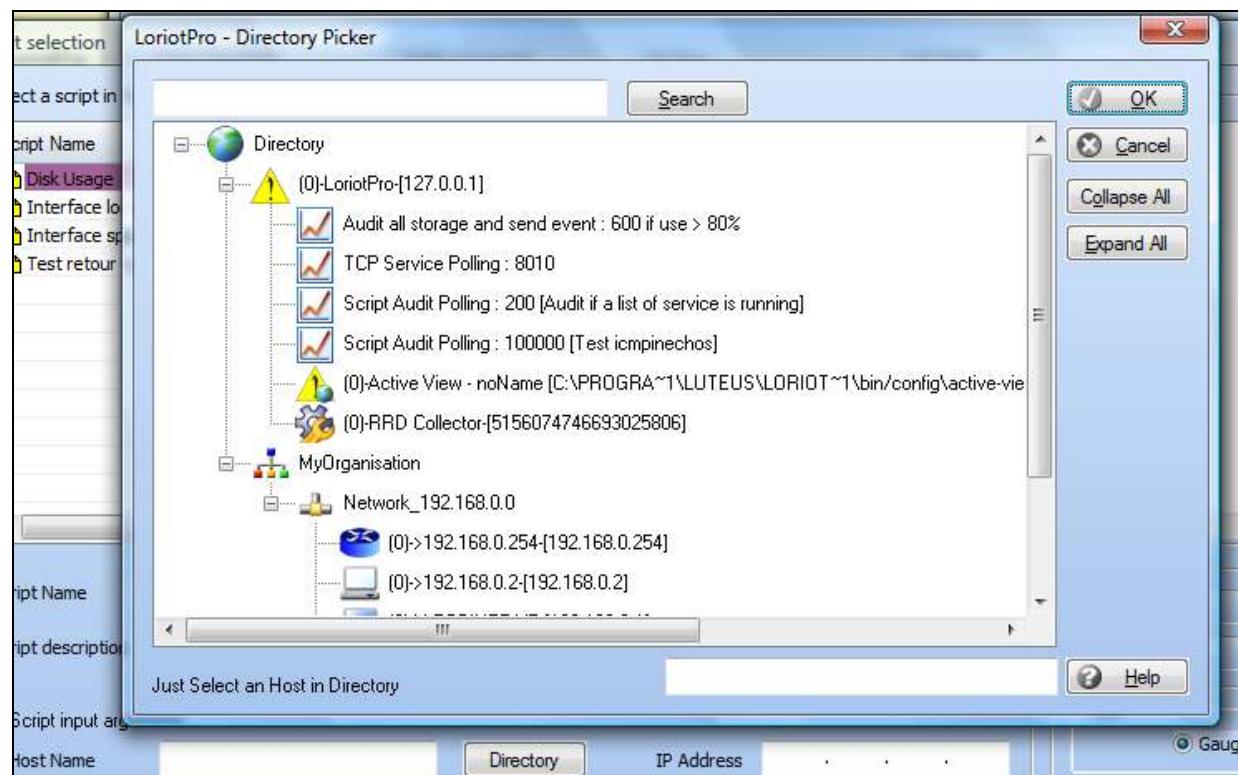
##### Attention

Modifiez ce texte en fonction de vos besoins car il sera utilisé pour générer vos rapports et vous ne pourrez plus le modifier une fois la data base RRD initialisée.

#### 13.5.3.2.2 Zone – Script Input arguments

##### 13.5.3.2.2.1 Option – Host Name

Cliquez sur le bouton Directory pour remplir ce champ automatiquement.



Selectionnez votre équipement.

### 13.5.3.2.2.2 Option – IP Address

Cliquez sur le bouton Directory pour remplir ce champ automatiquement. La variable [Ip\\_host](#) sera transmise au script en utilisant cette adresse IP.

### 13.5.3.2.2.3 Option – Arg1 (SNMP oid)

Vous pouvez passer deux variables à un script, [Ip\\_oid](#) et [Ip\\_index](#). Ces deux variables seront utilisées par votre script pour réaliser votre traitement.

Ici vous renseignez la variable [Ip\\_oid](#).

### 13.5.3.2.2.4 Option – Arg2 (SNMP index)

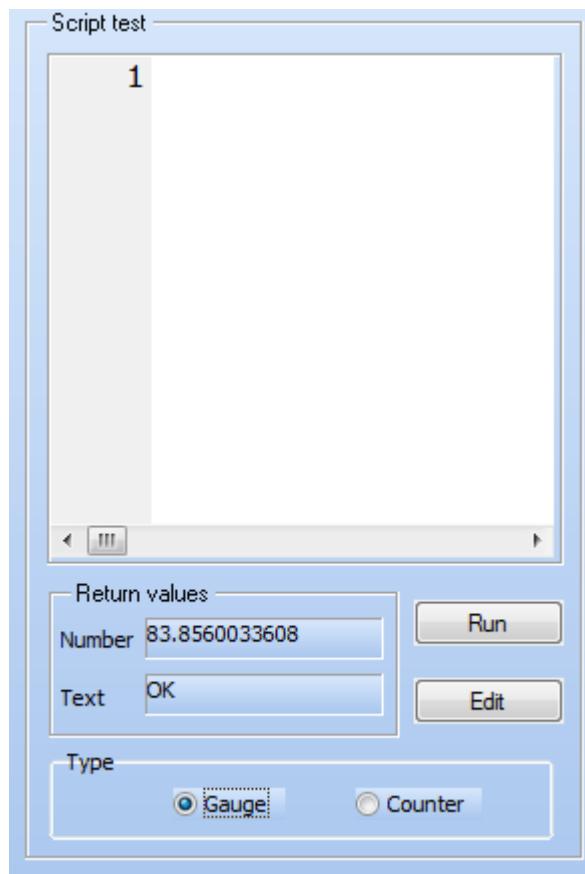
Vous pouvez passer deux variables à un script, [Ip\\_oid](#) et [Ip\\_index](#). Ces deux variables seront utilisées par votre script pour réaliser votre traitement.

Ici vous renseignez la variable [Ip\\_index](#).

Dans notre exemple le script ‘Disk Usage’ a besoin de connaître l’index du disque à grapher, ici nous utilisons l’index ‘[.1](#)’.

The screenshot shows a configuration dialog box titled 'Script input arguments'. It contains four input fields: 'Host Name' (LoriotPro), 'IP Address' (127 . 0 . 0 . 1), 'Arg1 (SNMP oid)' (empty), and 'Arg2 (SNMP index)' (.1). There are also two buttons: 'Directory' (highlighted in blue) and 'MIB Tree'.

### 13.5.3.2.3 Zone – Script test



Il est possible de tester le script avec les paramètres **Ip\_oid** et **Ip\_index** fournis avec la zone ‘script test’ .

#### 13.5.3.2.3.1 Zone – Type



Il faut choisir le type de valeur renvoyée par le script pour réaliser le graphe.

Si le résultat vous convient, cliquez le bouton ‘OK’

Une entrée de collecte est insérée dans la liste (base RRD).

LUA Script								
Script Name	Script Description	Scri...	Host ...	IP ad...	SNM...	SNM...	Data Source Na...	Type
Disk Usage	Ip-oid should be initialized ...	C:\P...	Lorio...	127....	.1	DS0		GAUGE

Nous pouvons insérer une ligne par disque présent sur notre équipement.

#### 13.5.3.2.4 Option – Delete

Delete la ligne sélectionnée

#### **13.5.3.2.5 Option – Move up**

Permet le déplacement d'une ligne vers le haut.

#### **13.5.3.2.6 Option – Move down**

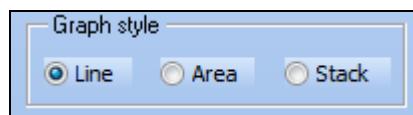
Permet le déplacement d'une ligne vers le bas.

#### **13.5.3.2.7 Option – Clear all**

Delete toutes les lignes.

#### **13.5.3.2.8 Option – Graph style**

Le style du graphe : line, Area ou Stack.



#### **13.5.3.2.9 Option – Y axis legend**

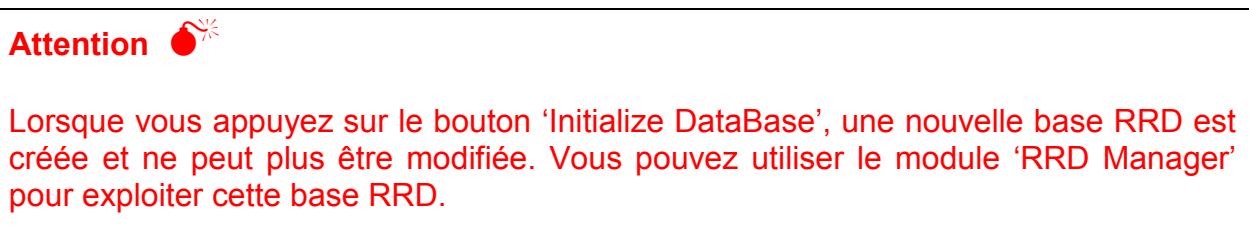
La légende de l'axe des Y

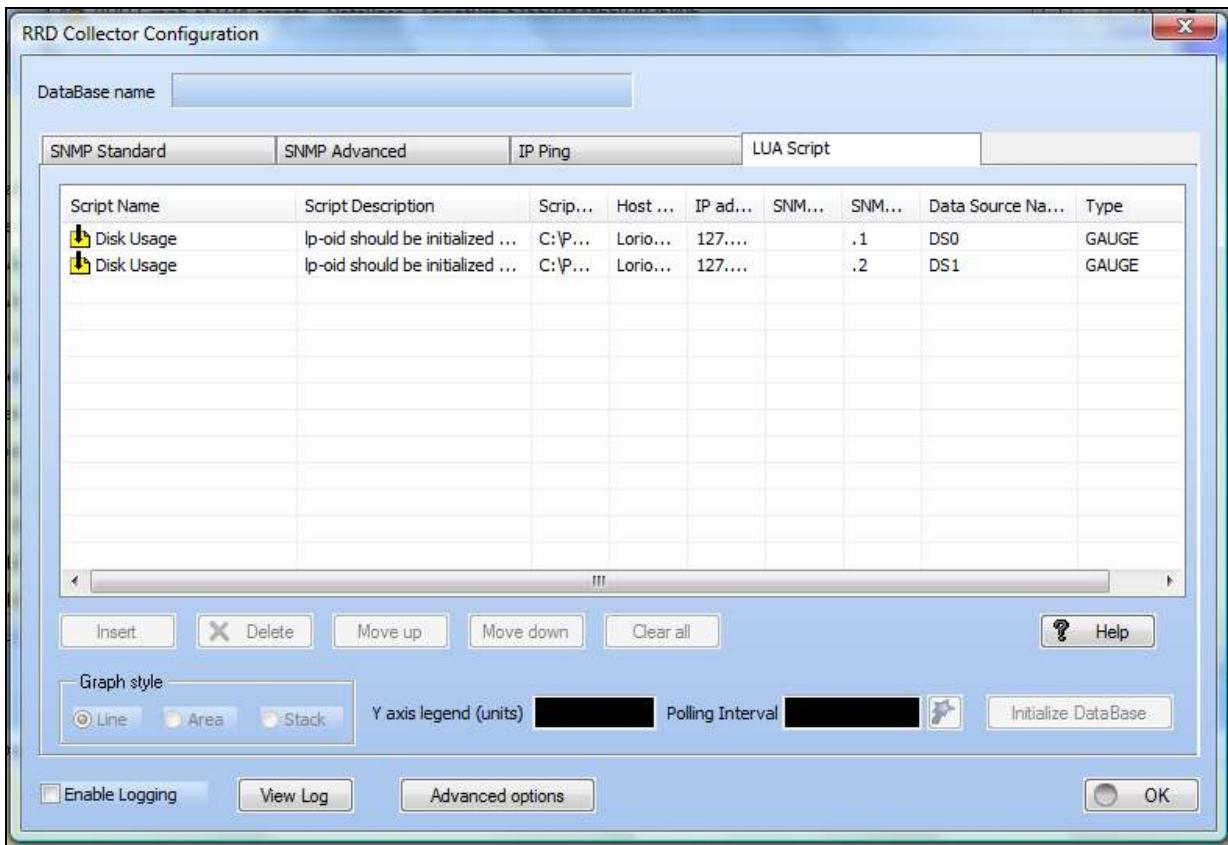
#### **13.5.3.2.10 Option – Polling Interval**

L'intervalle de lancement des scripts.

#### **13.5.3.2.11 Option – Initialize DataBase**

Cette option doit être validée lorsque votre graphe est entièrement configuré.





Les boutons de configurations sont grisés pour interdire toutes modifications.

### 13.5.3.2.12 Option – Enable Logging

Log les résultats de traitement du plugin pour réaliser un ‘débugging’ des scripts.

### 13.5.3.2.13 Option – View Log

Visualise la ‘log’ générée par le plugin.

```

19/01/08 18:20:06;LoriotPro-5156074746693025806;Start collecting
19/01/08 18:20:06;LoriotPro-5156074746693025806;rrdtool.exe update ".\RRDataBases\LoriotPro-5156074746693025806.rrd" -t DS0:DS1 N:83.9387278071:0.0000000000
19/01/08 18:21:06;LoriotPro-5156074746693025806;rrdtool.exe update ".\RRDataBases\LoriotPro-5156074746693025806.rrd" -t DS0:DS1 N:83.9387278071:0.0000000000
19/01/08 18:22:06;LoriotPro-5156074746693025806;rrdtool.exe update ".\RRDataBases\LoriotPro-5156074746693025806.rrd" -t DS0:DS1 N:83.9396491288:0.0000000000
19/01/08 18:23:06;LoriotPro-5156074746693025806;rrdtool.exe update ".\RRDataBases\LoriotPro-5156074746693025806.rrd" -t DS0:DS1 N:83.9396491288:0.0000000000
19/01/08 18:24:06;LoriotPro-5156074746693025806;rrdtool.exe update ".\RRDataBases\LoriotPro-5156074746693025806.rrd" -t DS0:DS1 N:83.9427048456:0.0000000000
19/01/08 18:24:14;LoriotPro-5156074746693025806;rrdtool.exe graph ".\www\RRDgraphs\LoriotPro-5156074746693025806-monthly.gif" --start -2419200 --imgformat GIF --width 600 --height 200 --title "Last Month Data (LINE) - Last update : Saturday, January 19, 2008 18:24:14"
DEF:"DS0"=".\\RRDataBases\\LoriotPro-5156074746693025806.rrd":DS0:AVERAGE
DEF:"DS1"=".\\RRDataBases\\LoriotPro-5156074746693025806.rrd":DS1:AVERAGE LINE2:"DS0"#0000CC:"LUA Script\\:lp-oid should be initialized to either ifinoctets or ifoutoctets\\1"
LINE2:"DS1"#00FFCC:"LUA Script\\:lp-oid should be initialized to either ifinoctets or ifoutoctets\\1"
19/01/08 18:24:14;LoriotPro-5156074746693025806;RRDtool.exe return: 680x285

```

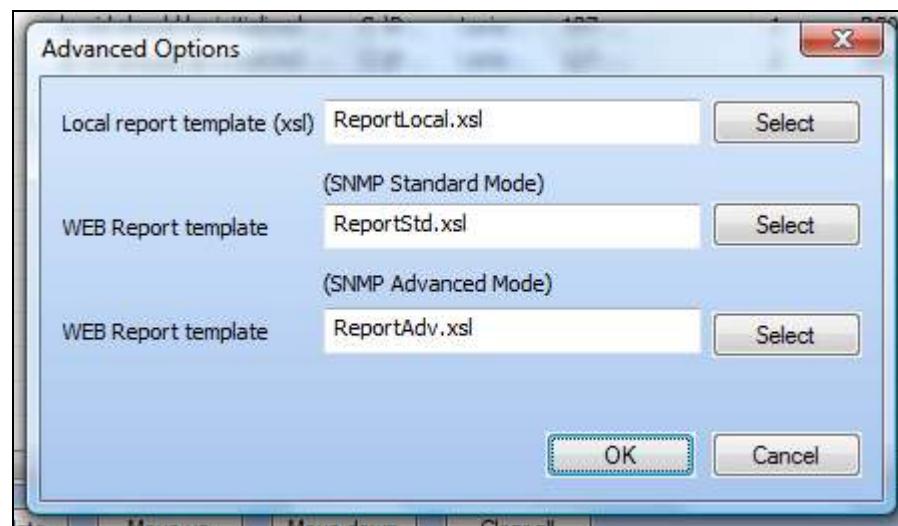
## LoriotPro V5.00 Extended Edition – Supervision et gestion SNMP par scripts LUA Introduction au System Expert et aux scripts

```
19/01/08 18:24:21;LoriotPro-5156074746693025806;rrdtool.exe graph ".\www\RRDgraphs\LoriotPro-5156074746693025806-hourly.gif" --start -3600 --imgformat GIF --width 600 --height 200 --title "Last Hour Data (LINE) - Last update : Saturday, January 19, 2008 18:24:21"  
DEF:"DS0"=".\\RRDataBases\LoriotPro-5156074746693025806.rrd":DS0:AVERAGE  
DEF:"DS1"=".\\RRDataBases\LoriotPro-5156074746693025806.rrd":DS1:AVERAGE LINE2:"DS0"#0000CC:"LUA Script\lp-oid should be initialized to either ifinoctets or ifoutoctets\l"  
LINE2:"DS1"#00FFCC:"LUA Script\lp-oid should be initialized to either ifinoctets or ifoutoctets\l"  
19/01/08 18:24:22;LoriotPro-5156074746693025806;RRDtool.exe return: 680x285
```

```
19/01/08 18:25:06;LoriotPro-5156074746693025806;rrdtool.exe update ".\RRDataBases\LoriotPro-5156074746693025806.rrd" -t DS0:DS1 N:83.9427048456:0.0000000000  
19/01/08 18:25:26;LoriotPro-5156074746693025806;rrdtool.exe graph ".\www\RRDgraphs\LoriotPro-5156074746693025806-hourly.gif" --start -3600 --imgformat GIF --width 600 --height 200 --title "Last Hour Data (LINE) - Last update : Saturday, January 19, 2008 18:25:26"  
DEF:"DS0"=".\\RRDataBases\LoriotPro-5156074746693025806.rrd":DS0:AVERAGE  
DEF:"DS1"=".\\RRDataBases\LoriotPro-5156074746693025806.rrd":DS1:AVERAGE LINE2:"DS0"#0000CC:"LUA Script\lp-oid should be initialized to either ifinoctets or ifoutoctets\l"  
LINE2:"DS1"#00FFCC:"LUA Script\lp-oid should be initialized to either ifinoctets or ifoutoctets\l"  
19/01/08 18:25:26;LoriotPro-5156074746693025806;RRDtool.exe return: 680x285
```

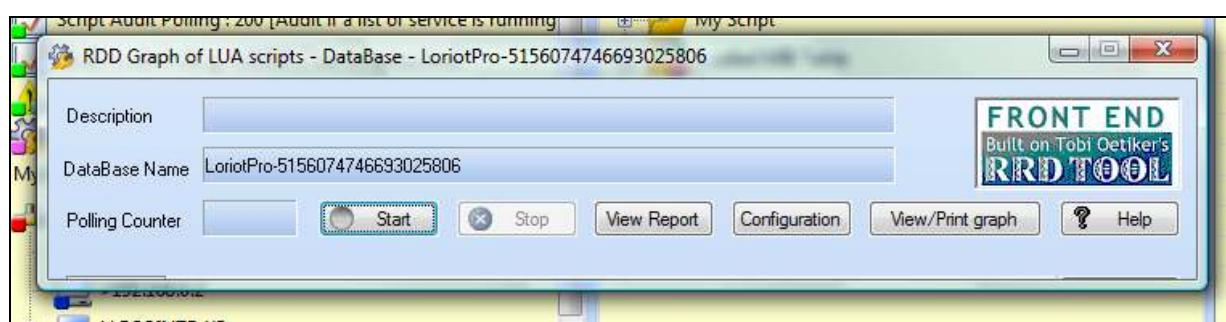
### 13.5.3.2.14 Option – Advanced options

Les fichiers de rapport générés sont au format XML, il est possible d'utiliser des fichiers de description non standard pour coller aux besoins de votre entreprise en terme de visuel.



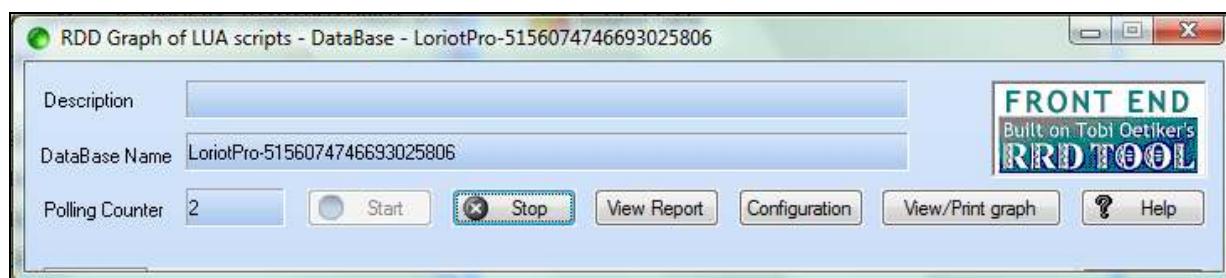
### 13.5.3.3 Lancement du plugin

Une fois le plugin configuré, faites 'OK' pour exploiter les graphes réalisés.



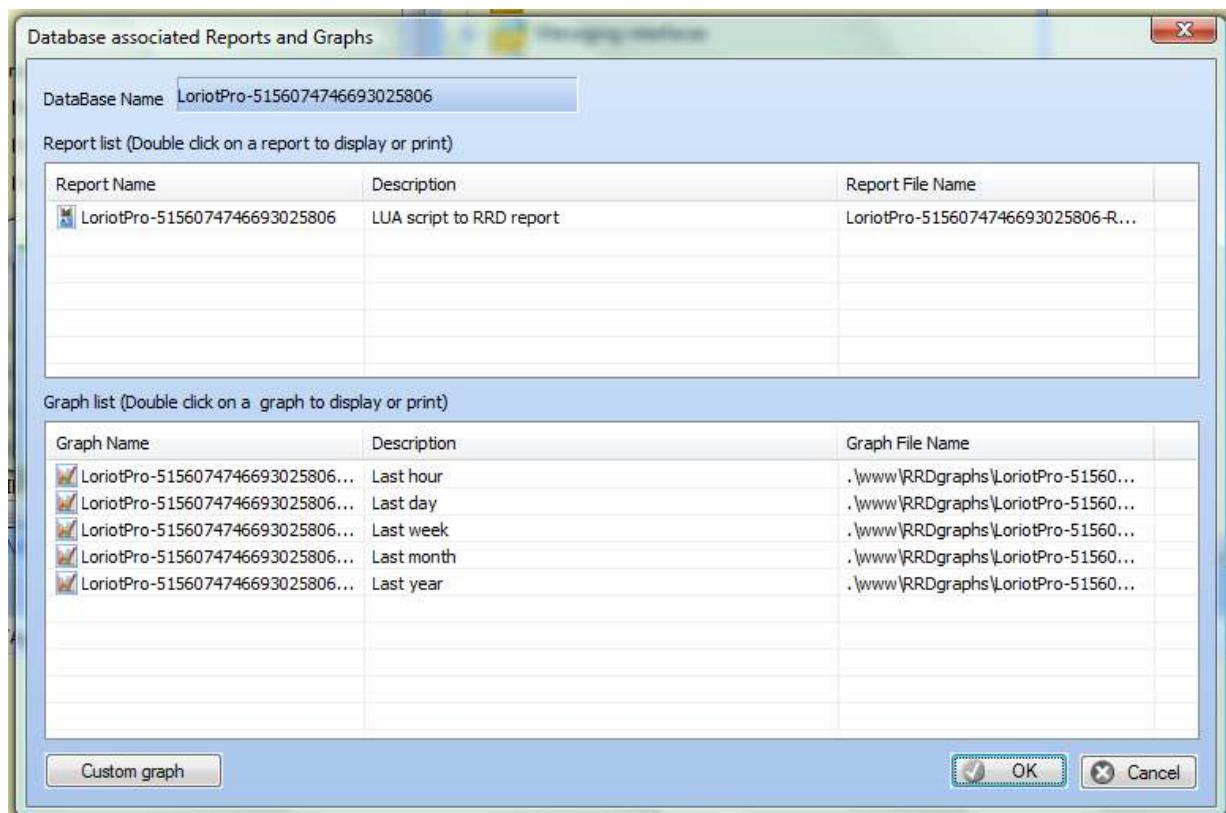
Appuyer sur le bouton ‘Start’ pour lancer le plugin.

Le champ ‘Pollign Counter’ vous informe sur le nombre de lancements des scripts réalisés.



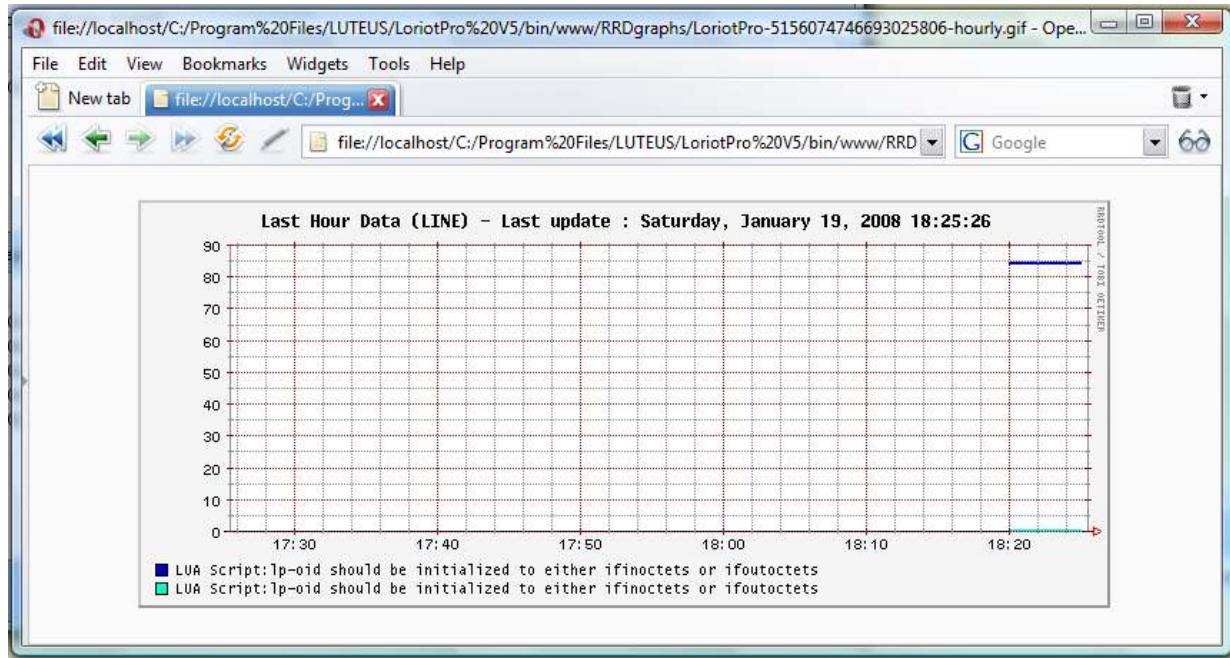
#### 13.5.3.4 Visualisation des graphes

Appuyez sur le bouton ‘View Report’, une fenêtre de sélection apparaît.

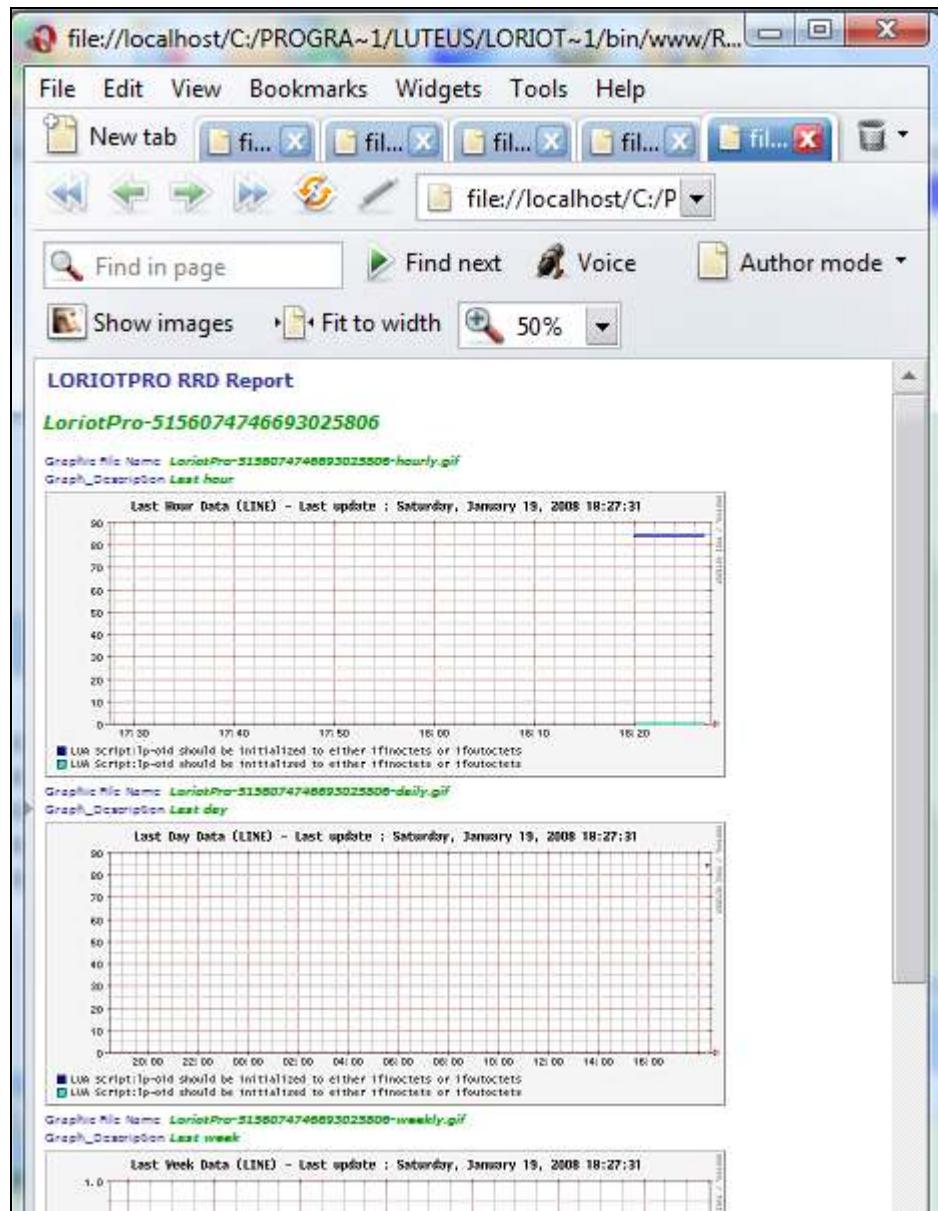


Le module vous propose de voir le graphe par heure, jour, semaine, mois ou année.

Un ‘double click’ sur la ligne ‘Graph List’ affiche le graphe désiré, par exemple ‘Last hour’ :

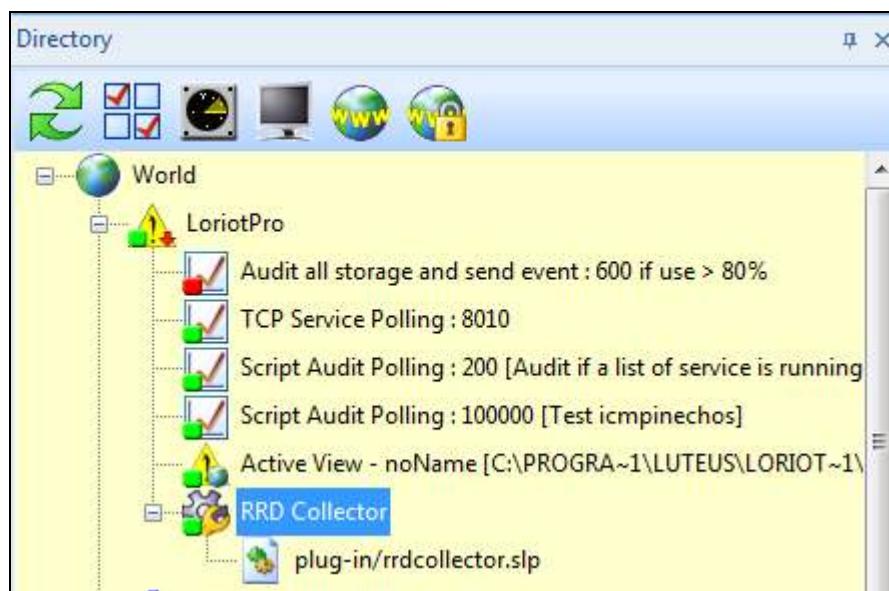


Pour réaliser un rapport incluant tous les graphes, vous pouvez utiliser le module ‘RRD Manager’ qui est décrit dans la documentation générale du logiciel, ou ‘double cliquer’ sur la ligne ‘Report Graph’

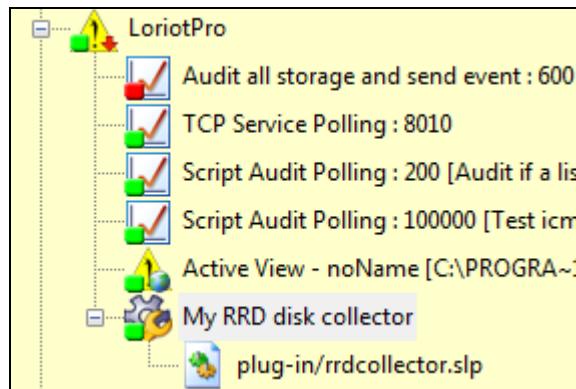


### 13.5.3.5 Visualisation des graphes avec les fenêtres volantes

Le plugin est inséré dans la Directory en dessous de l'équipement préalablement sélectionné.

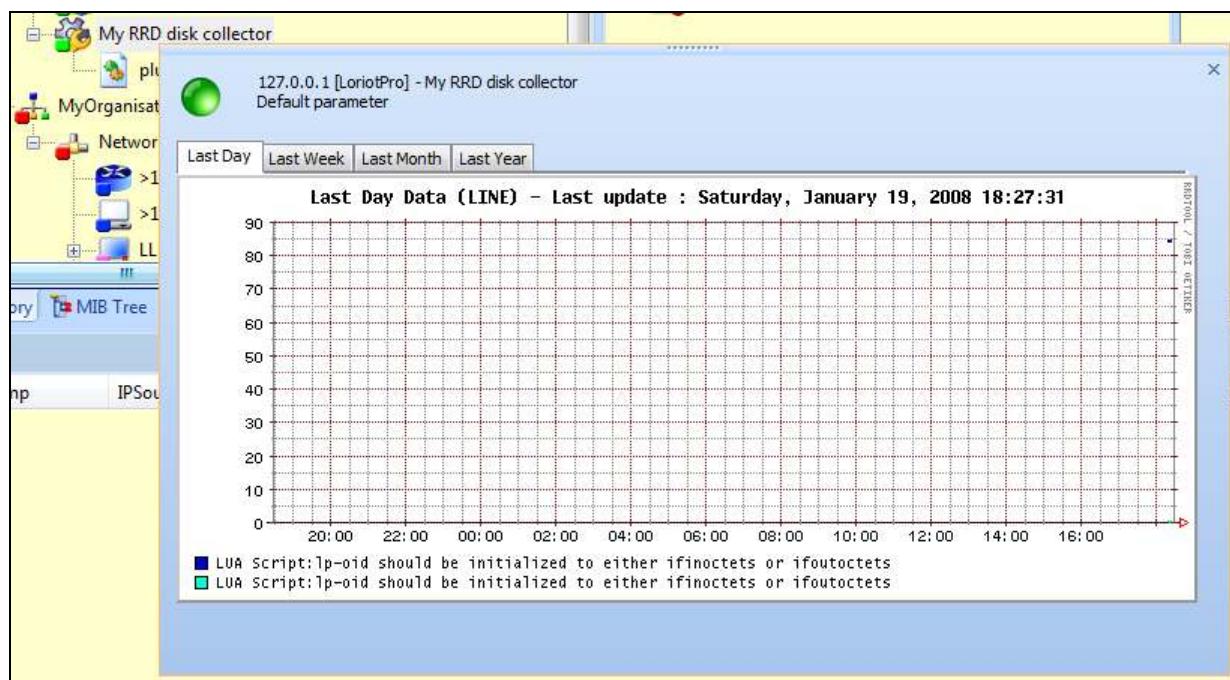


Il est possible d'identifier le nom du graphe directement dans la Directory.



Un simple clic de souris sur l'item dans la Directory ou les autres 'WorkSpace' (Task Manager) affiche une fenêtre volante avec l'affichage des graphes associés au plugin.

LoriotPro V5.00 Extended Edition – Supervision et gestion SNMP par scripts LUA  
Introduction au System Expert et aux scripts



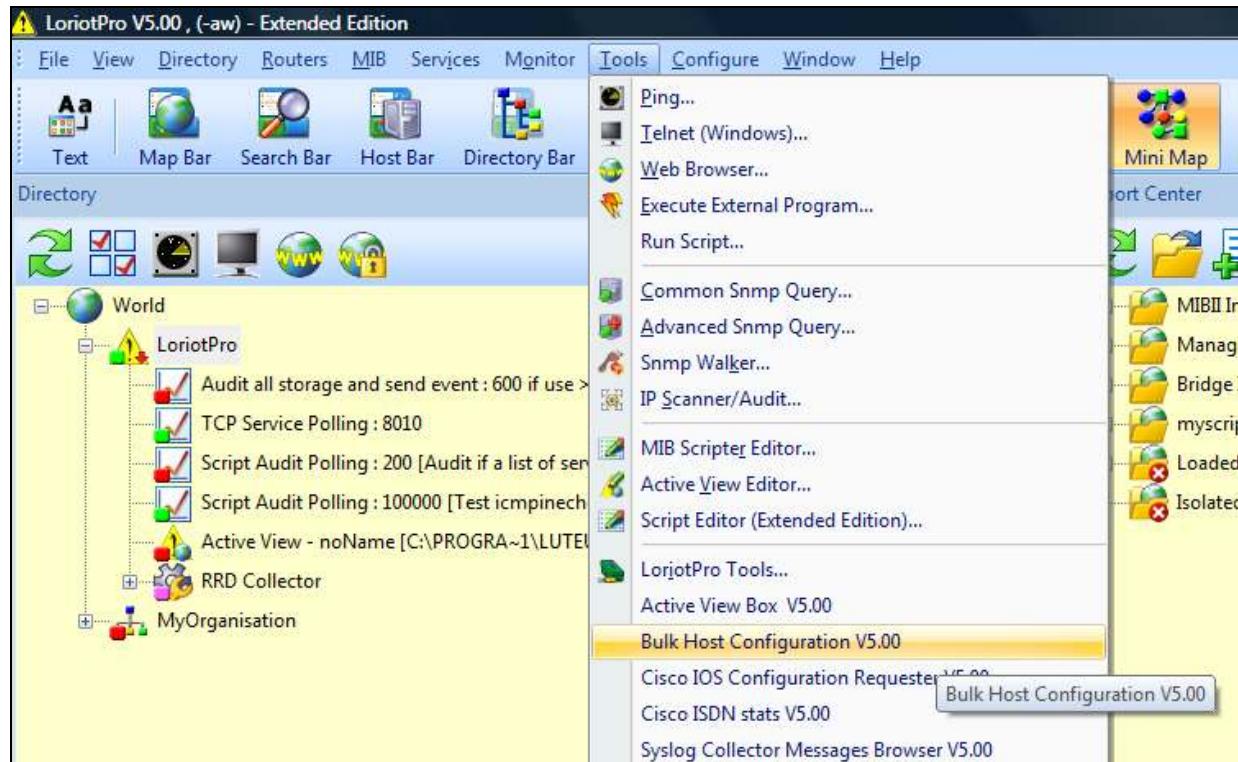
## 14 Le module de configuration multiple – Bulk configuration

### 14.1 Introduction

LoriotPro Extended Edition propose un module permettant de réaliser des modifications sur une sélection d'équipement. Ce module de configuration par lot peut être utilisé pour lancer des scripts pour un ensemble d'équipements présélectionnés.

Lancement du module :

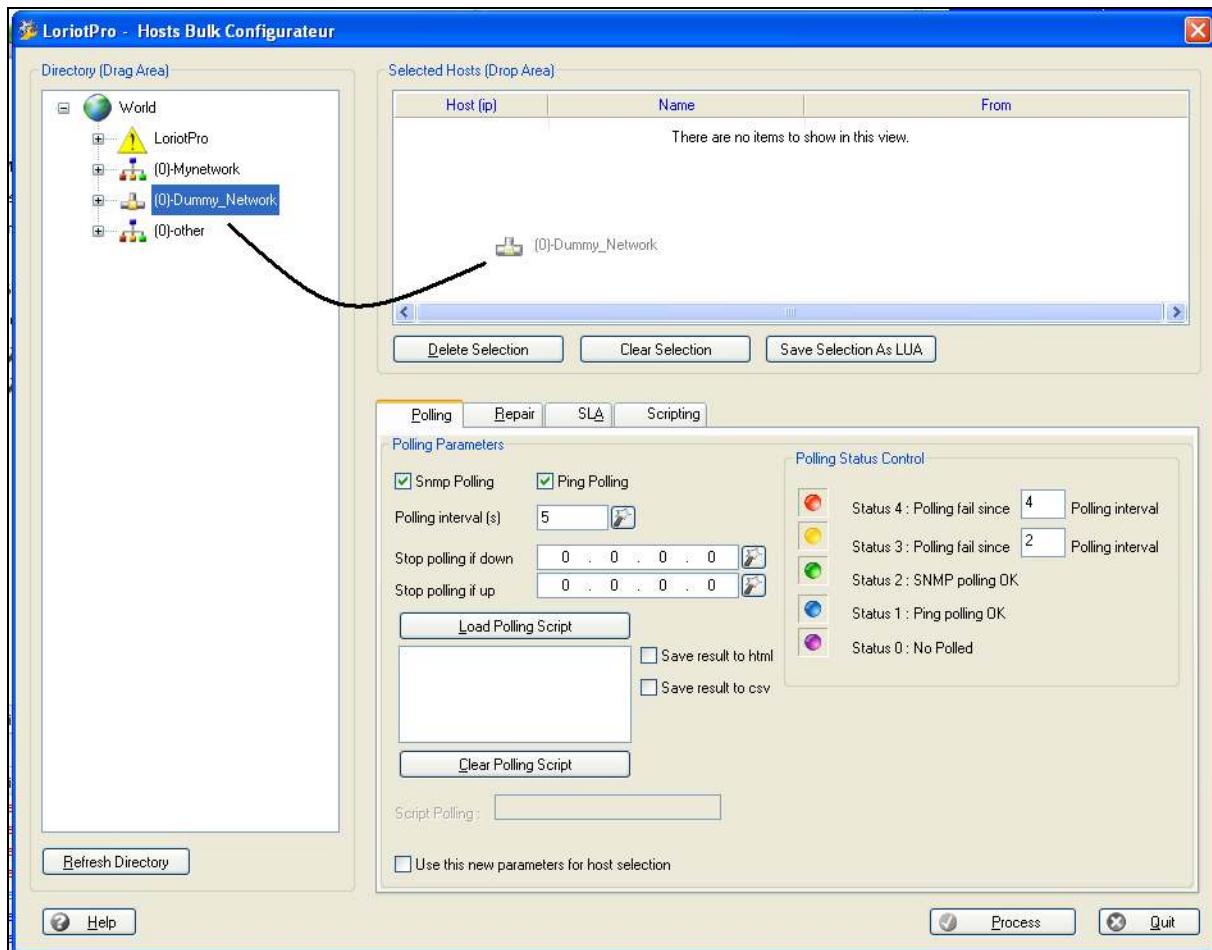
**Tools>Bulk Host Configuration**



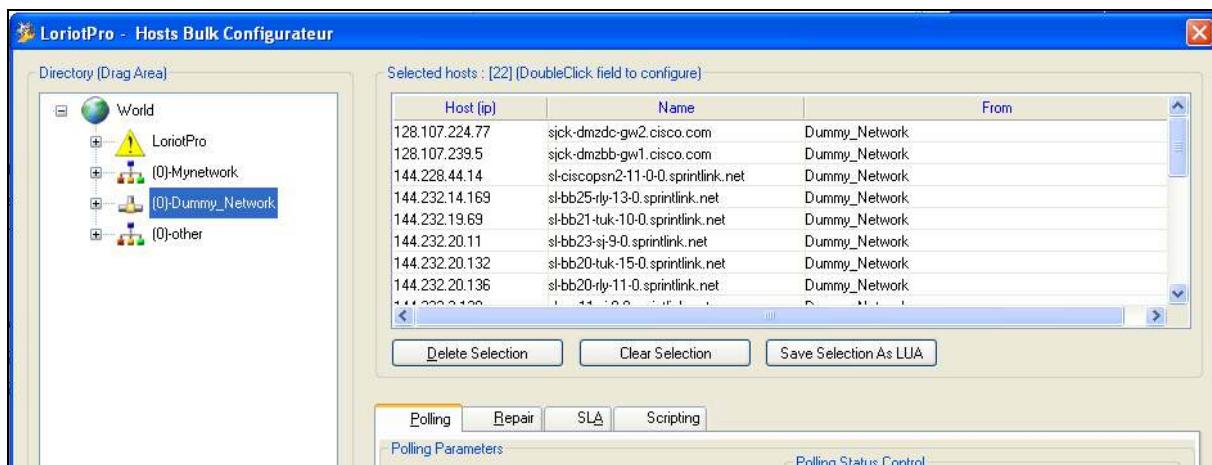
### 14.2 Sélection multiple d'équipement

Ce module permet de réaliser des configurations ou des tâches de management des équipements par lot.

## LoriotPro V5.00 Extended Edition – Supervision et gestion SNMP par scripts LUA Introduction au System Expert et aux scripts

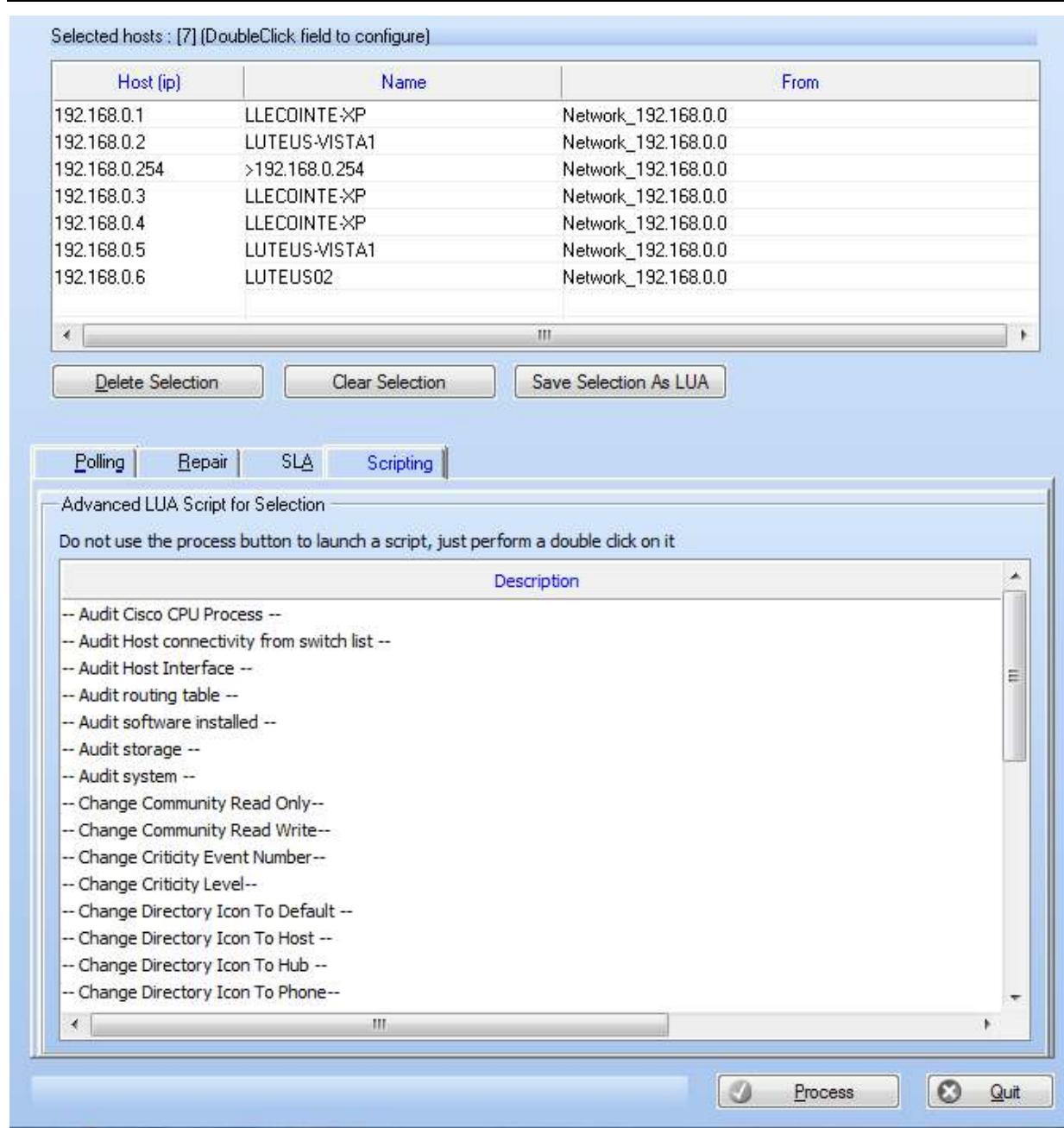


Il suffit de glisser les équipements de l'arbre vers la zone de sélection pour créer une liste.



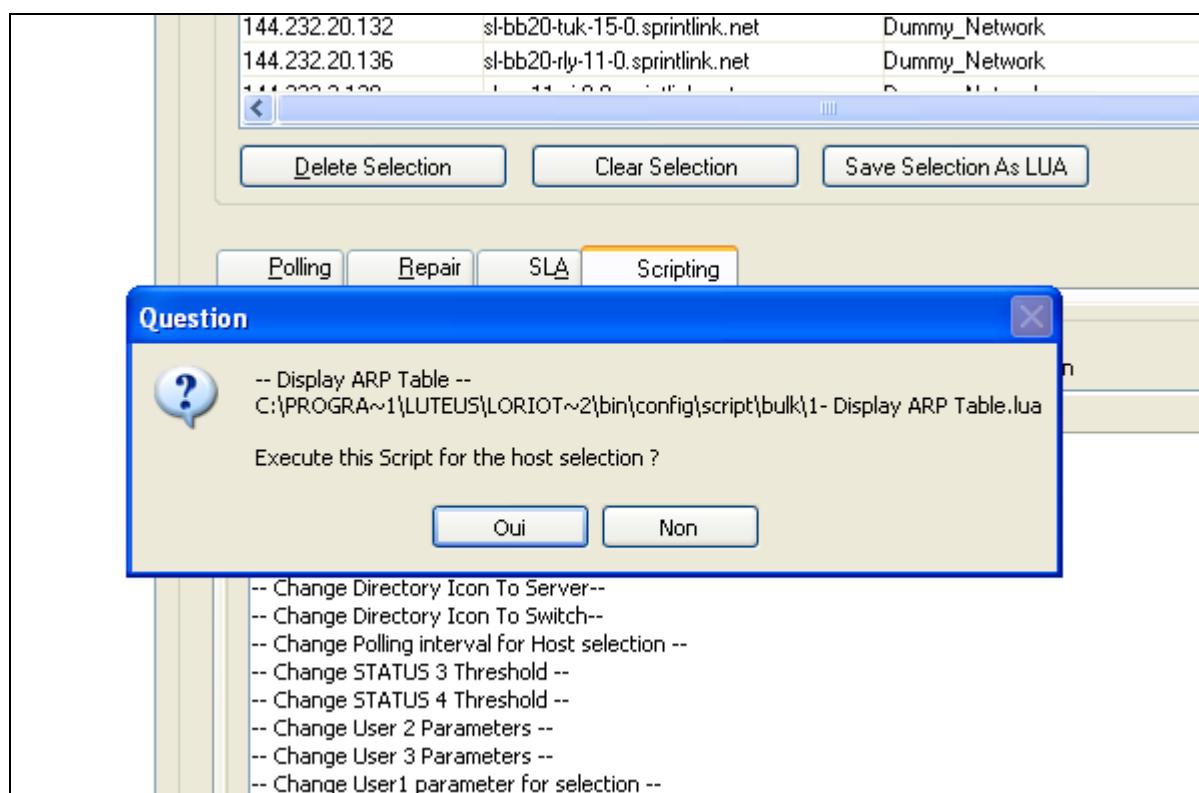
L'onglet Scripting permet d'avoir accès à une liste de scripts pré écrits permettant une gestion par lot.

## LoriotPro V5.00 Extended Edition – Supervision et gestion SNMP par scripts LUA Introduction au System Expert et aux scripts

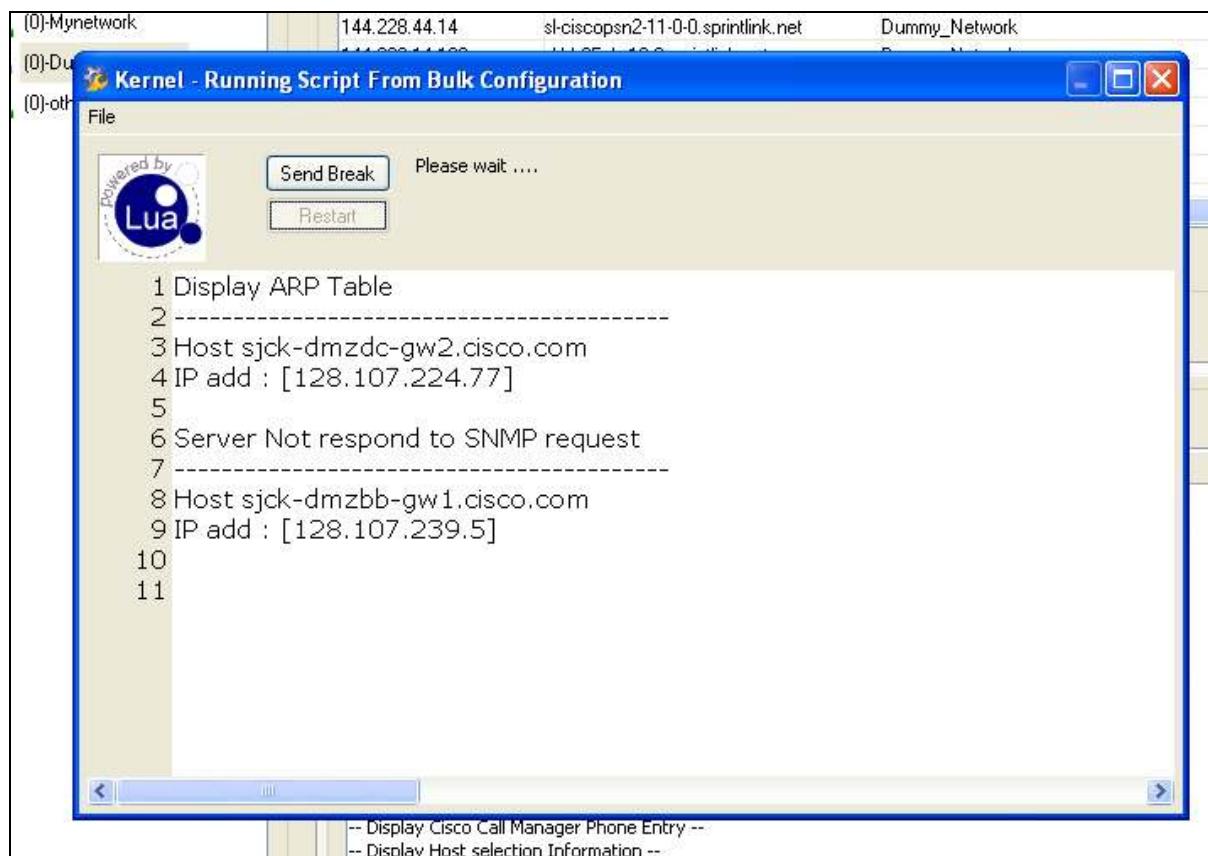


Un simple double clic sur une des lignes de la listbox lance le script en l'associant à la liste des équipements.

LoriotPro V5.00 Extended Edition – Supervision et gestion SNMP par scripts LUA  
Introduction au System Expert et aux scripts

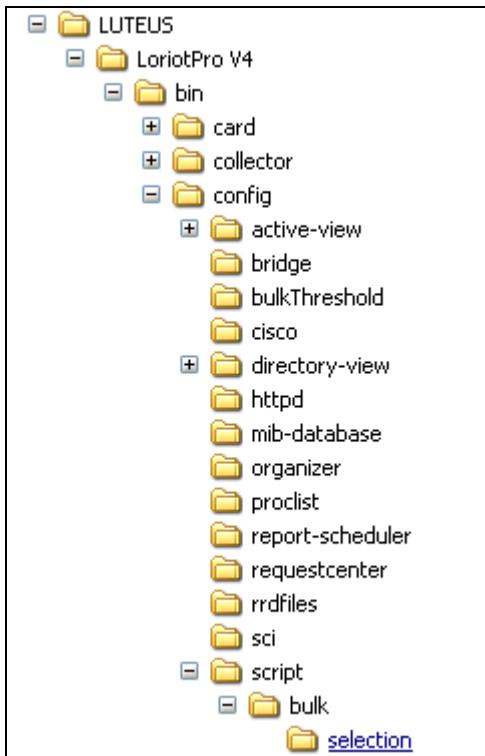


le module de lancement de script est appelé



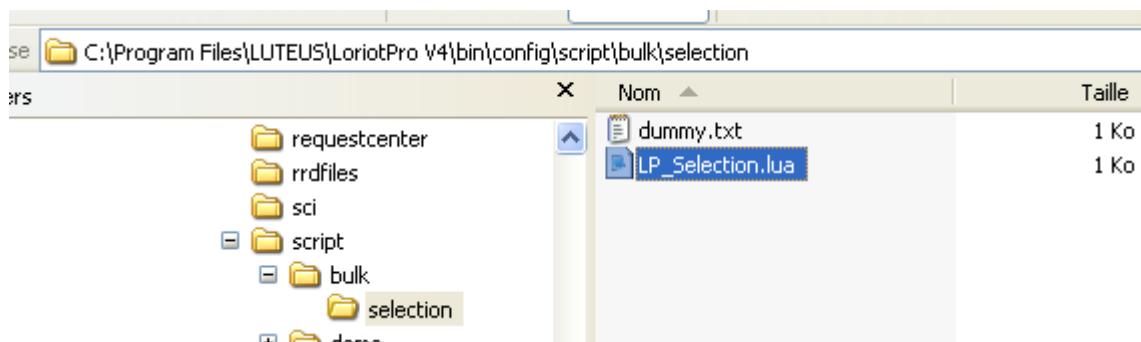
#### 14.2.1.1 Crédation de nouveau script de type « bulk »

Les scripts de type bulk sont placés dans le répertoire bin/config/scripts/bulk



Lors de l'appel du script, un fichier LP\_Selection.lua est créé dans le répertoire :

**\bin\config\script\bulk\selection**



Le fichier contient une fonction (LP\_HostsSelection(tab)) pré configurée qui contient la liste des équipements sélectionnés.

```
-- Loriotpro V5
function LP_HostsSelection(tab)
tab[0]="128.107.224.77"
tab[1]="128.107.239.5"
tab[2]="144.228.44.14"
```

```
tab[3] = "144.232.14.169"  
tab[4] = "144.232.19.69"  
tab[5] = "144.232.20.11"  
tab[6] = "144.232.20.132"  
tab[7] = "144.232.20.136"  
tab[8] = "144.232.3.138"  
tab[9] = "144.232.3.193"  
tab[10] = "144.232.7.250"  
tab[11] = "154.14.186.1"  
tab[12] = "154.14.65.3"  
tab[13] = "154.14.65.7"  
tab[14] = "154.14.71.29"  
tab[15] = "154.14.71.34"  
tab[16] = "212.81.82.97"  
tab[17] = "213.206.128.55"  
tab[18] = "213.206.129.143"  
tab[19] = "217.149.32.97"  
tab[20] = "217.149.33.149"  
tab[21] = "66.249.85.99"  
return 22;  
end
```

Cette fonction est utilisée par le script lancé à travers le module de configuration d'équipements par lot.

#### 14.2.1.1.1 Exemple d'un script de type bulk

Voici un exemple de script de type bulk utilisant les ressources du fichier dynamique LP\_Selection.lua.

La première ligne du script est utilisée pour définir l'intitulé du script dans la listbox du module.

```
-- Change Directory Icon To Router--  
  
-- Ce fichier doit impérativement se trouver dans bin/config/script/bulk  
  
lp_value = 0;  
lp_buffer="init";  
  
-- use this to initialise the host selection  
dofile(lp.GetPath().."/config/script/bulk/selection/LP_Selection.lua")  
dofile(lp.GetPath().."/config/script/loriotinit.lua");  
  
hosts={};  
hostnumber=LP_HostsSelection(hosts);  
  
if hostnumber==0 then error("Not host selected\n") end
```

```
lp.Print("Host Number : ",hostnumber,"\n");

for j=0,hostnumber-1 do

    lp_value=lp.GetIPInformation(hosts[j],"aa");
    if lp_value then
        lp.Print("Modify : ",hosts[j]," ",aa.name,"\n");
        lp.Print("\tOld type : ",aa["device_type"],"\n");

        --[[  
#define DSNMP_IS_IP_DEVICE 0  
#define DSNMP_IS_IP_HOST 1  
#define DSNMP_IS_IP_SERVER 2  
#define DSNMP_IS_IP_HUB 3  
#define DSNMP_IS_IP_SWITCH 4  
#define DSNMP_IS_IP_ROUTER 5  
#define DSNMP_IS_IP_PRINTER 6  
#define DSNMP_IS_IP_PHONE 7  
--]]  
  
        lp.SetIPOption(hosts[j],LP_LUA_ISDEVICE,5);
        lp_value=lp.GetIPInformation(hosts[j],"aaa");
        lp.Print("\tNew type router : ",aaa["device_type"],"\n");

    else
        lp_buffer="#ERROR";
        lp.Print("no exist : ",hosts[j],"\n");
    end

end
```

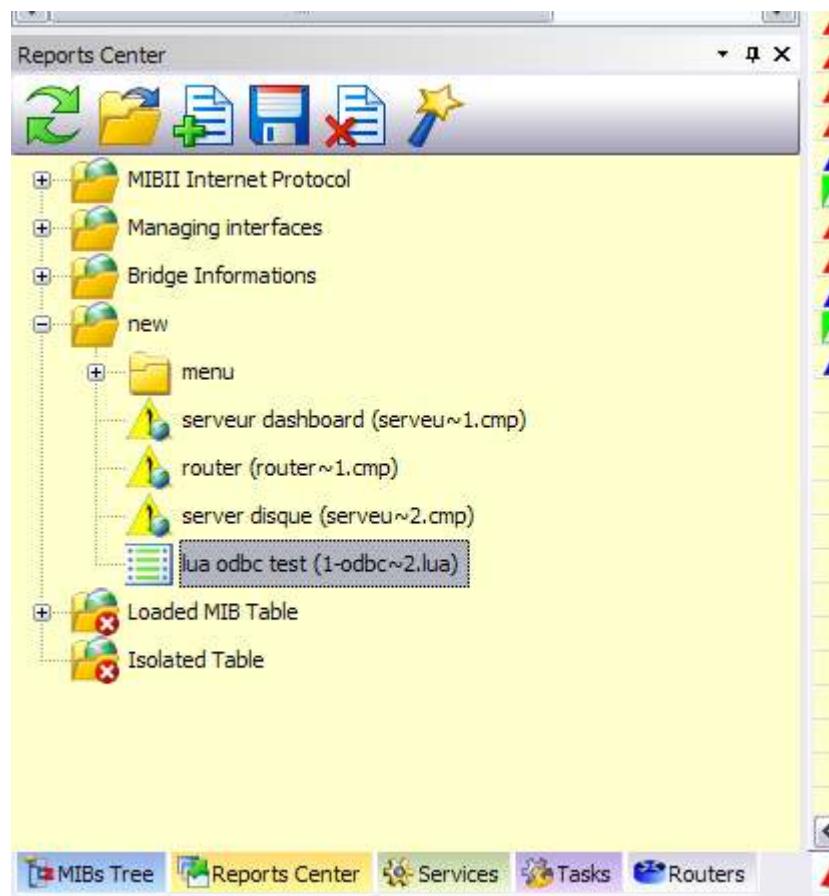
## 15 Report Center

### 15.1 Introduction

Le module ‘Report Center’ a pour vocation de vous permettre de classer et présenter vos scripts et/ou rapports dans un arbre structuré. Ces scripts ne seront pas associés à un équipement particulier mais pourront être exploités avec l’équipement sélectionné dans la Directory.

Le module Report Center est un « Workspace » de LoriotPro et se présente sous la forme d’une fenêtre repositionnable (‘Dockable’).





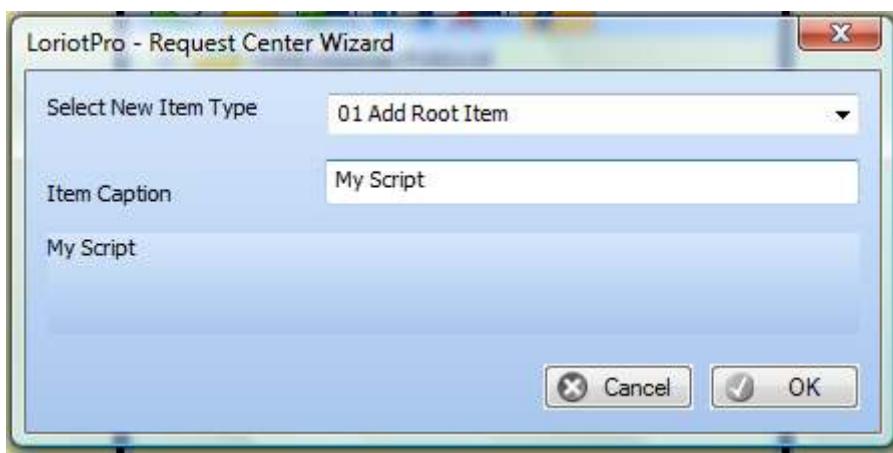
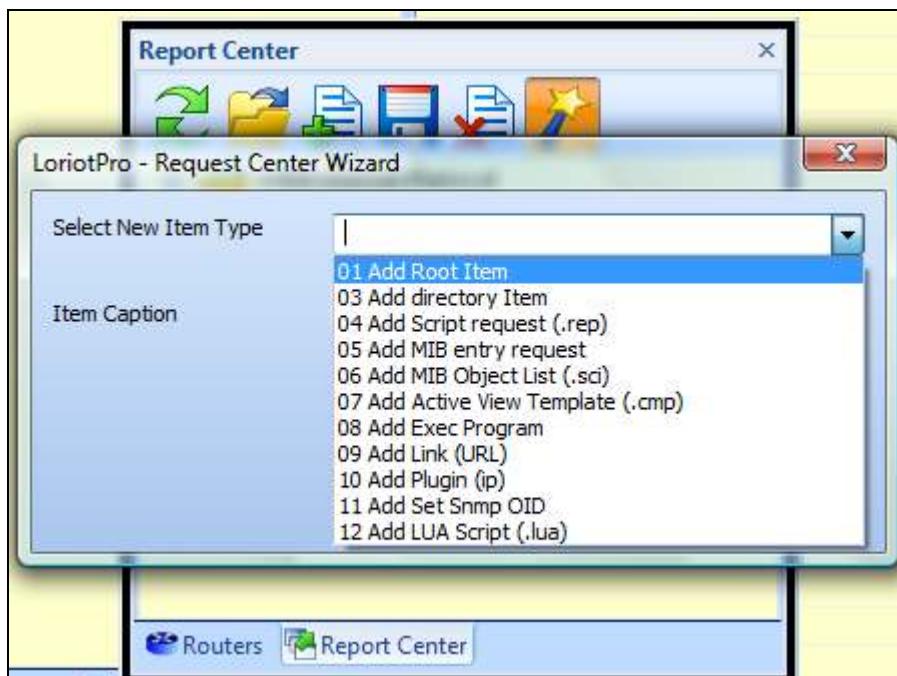
Les scripts mis en place dans le Report Center sont automatiquement associés à l'équipement par défaut sélectionné dans la Directory ou n'importe quel autre point du logiciel.

## 15.2 Configuration

Ce module possède un Wizard de création. L'exploitation de ce module est expliquée dans le documentation générale de LoriotPro, nous verrons ici uniquement l'exemple de l'insertion d'un script dans l'arbre.

### 15.2.1 Crédation d'une branche

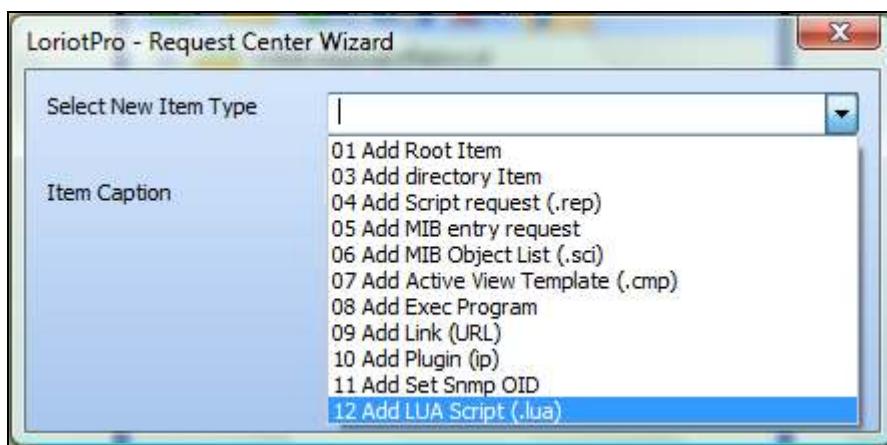
Il faut utiliser le Wizard et choisir l'insert d'un nouvel Item (**01 Add Root Item**)



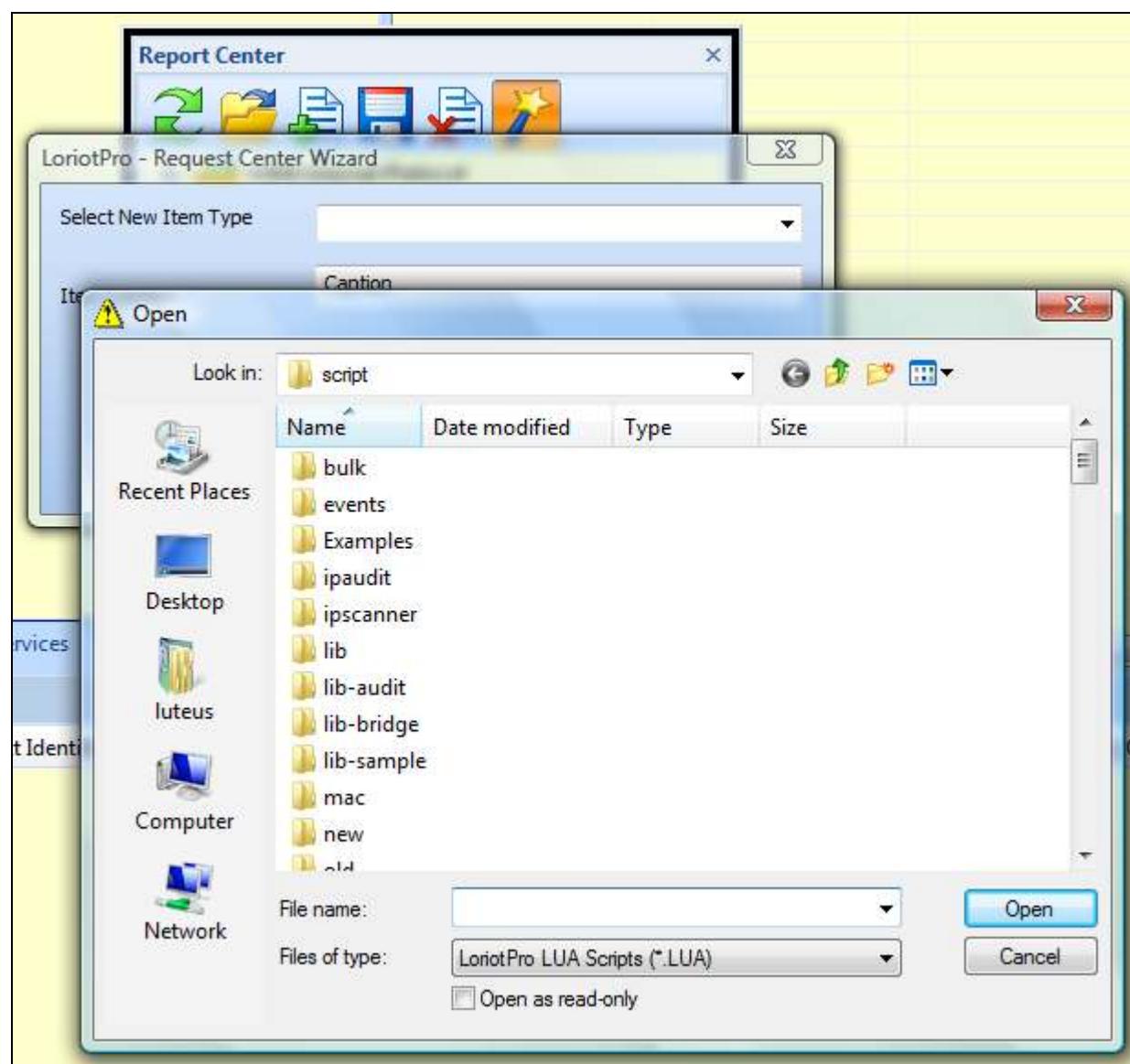
Faire 'OK'



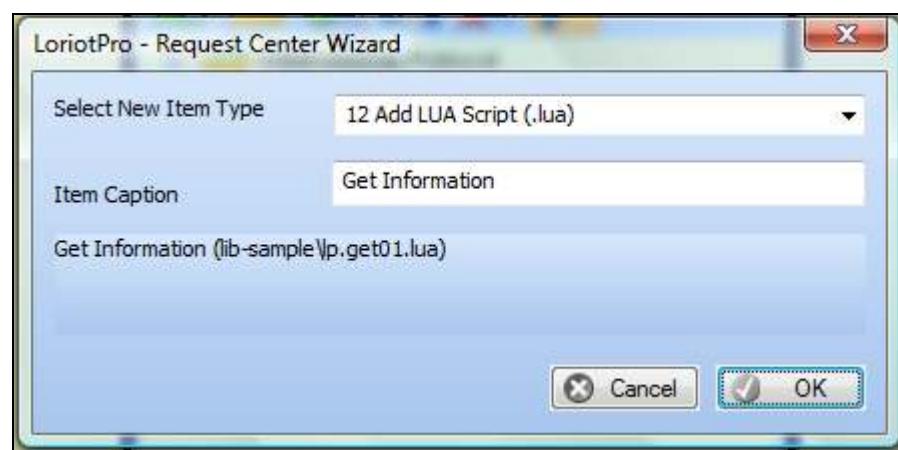
Nous insérons sous cet Item un '**12 Add LUA Script (.lua)**'

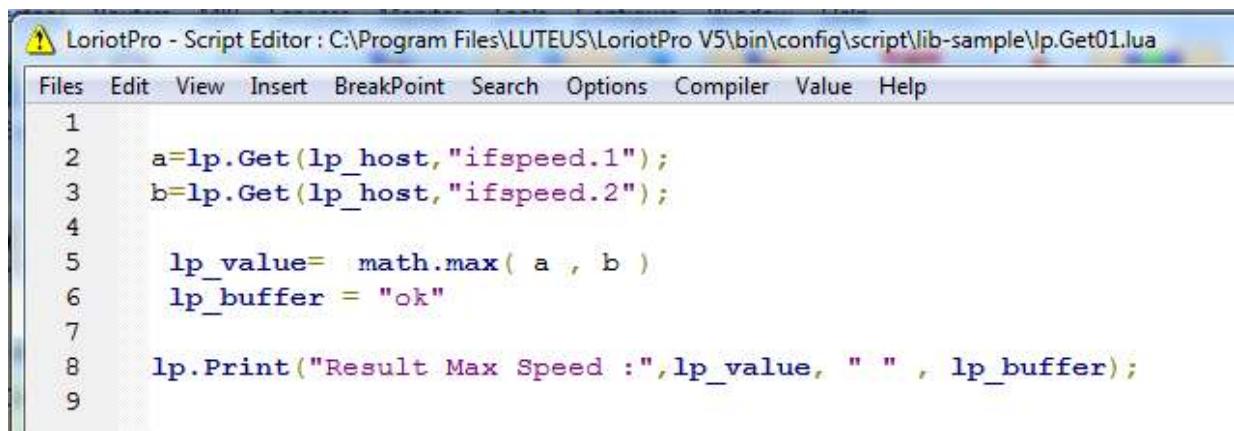


Une boîte de sélection de fichier est affiché.



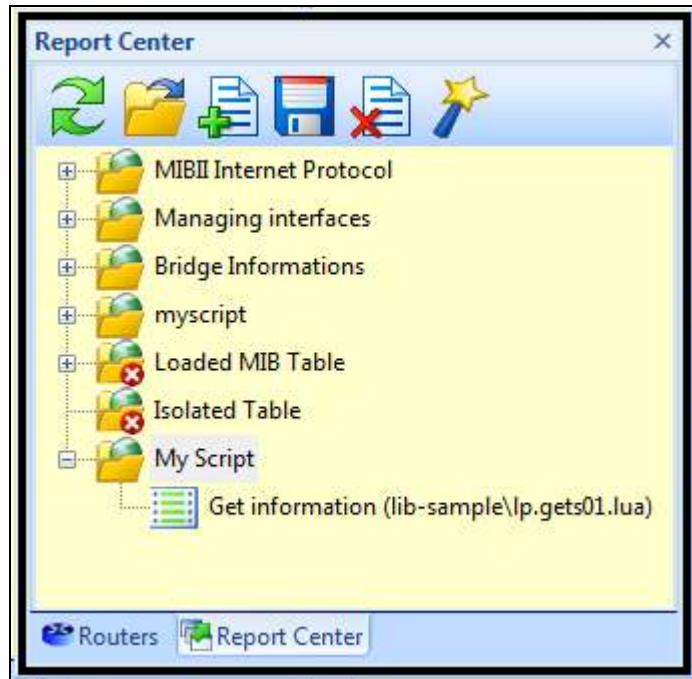
Un fichier LUA de script est choisi et un nom est donné au nouvel Item de l'arbre.





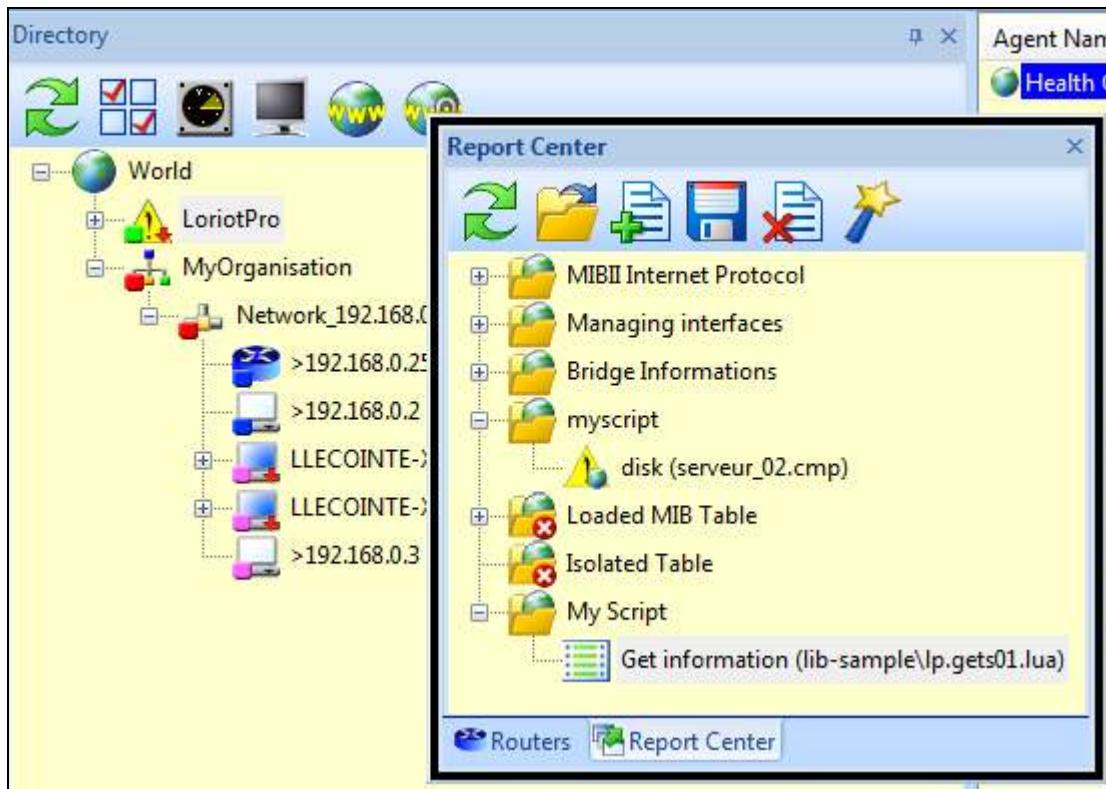
```
1
2     a=lp.Get(lp_host,"ifspeed.1");
3     b=lp.Get(lp_host,"ifspeed.2");
4
5     lp_value= math.max( a , b )
6     lp_buffer = "ok"
7
8     lp.Print("Result Max Speed : ",lp_value, " " , lp_buffer);
9
```

Le script sélectionné dans notre exemple collecte la vitesse de deux liens de l'équipement et affiche la vitesse la plus haute. Ce script n'est qu'un exemple simple de code pour illustrer notre propos.

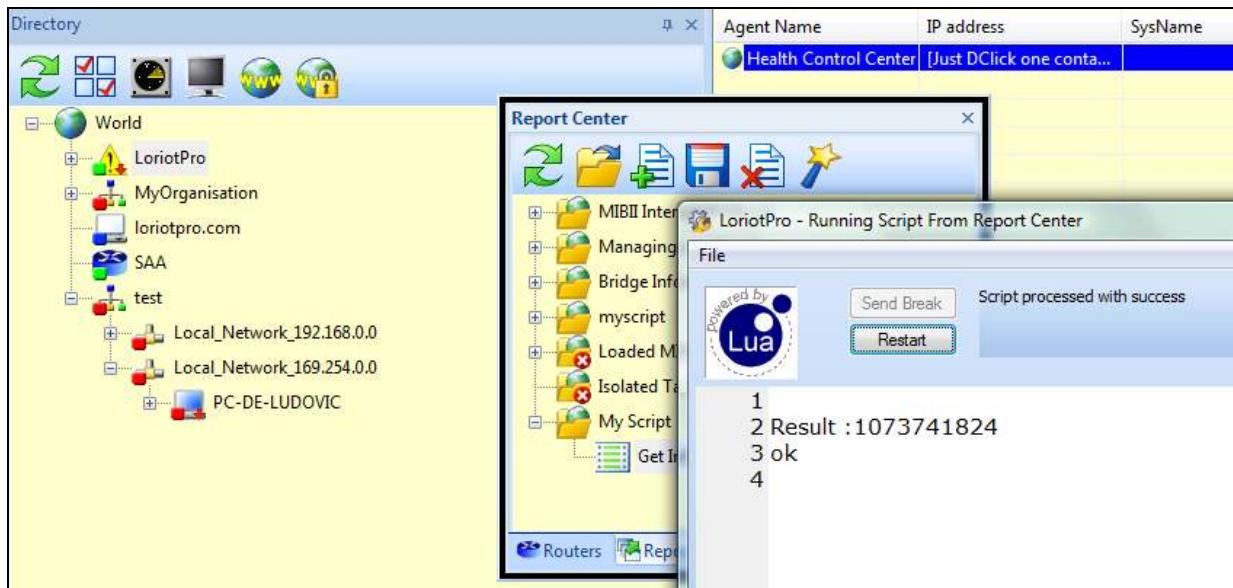


### 15.2.2 Exploitation du script d'exemple

Pour exploiter ce script, il suffit de sélectionner un équipement dans le logiciel (clic sur une représentation d'un équipement dans les arbres du logiciel).



Un double clic sur l'item créé dans l'arbre lancera le script en l'associant à l'équipement sélectionné.



Le module passe au script, comme variable globale, l'adresse IP de l'équipement dans la variable **ip\_host**.

Le « Report Center » peut aussi lancer des Activeviews modales qui ont la capacité de lancer des scripts au niveau de leurs objets graphiques.



## 16 Le Module IP Scanner

### 16.1 Introduction

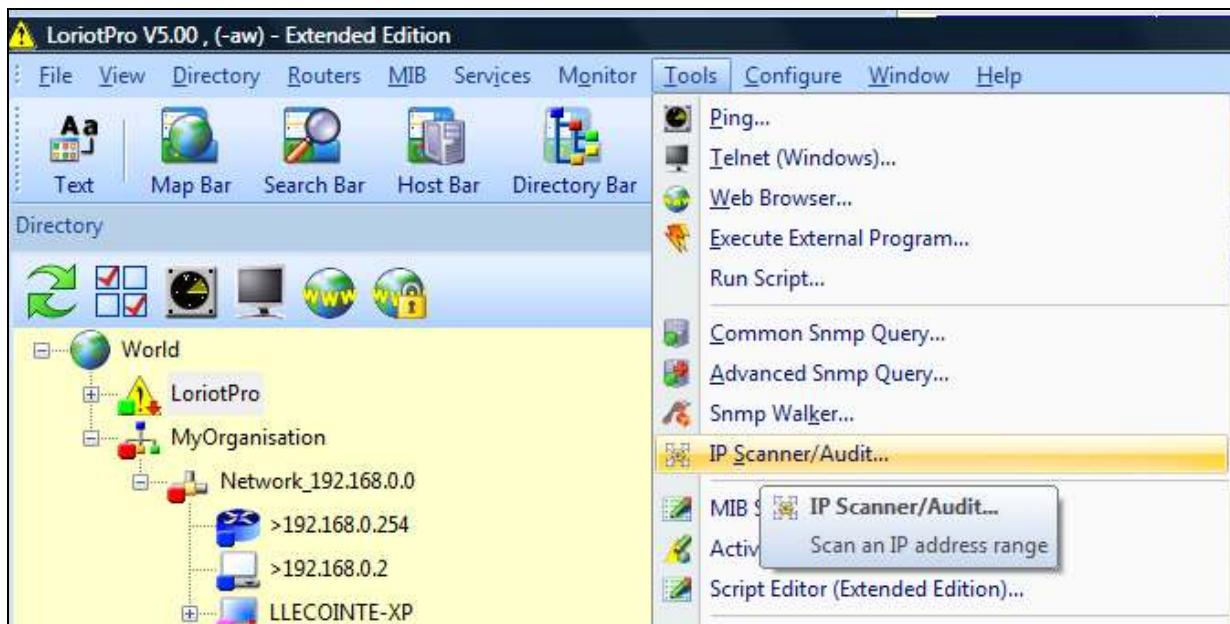
Le module IP Scanner permet de réaliser un scan ou interrogation d'un ou plusieurs équipements en réalisant des tests d'accès utilisant la couche réseaux. IP Scanner utilise différents protocoles et différentes méthodes de communication avec l'équipement en vue de réaliser un audit des flux, services ou fonctions supportés.

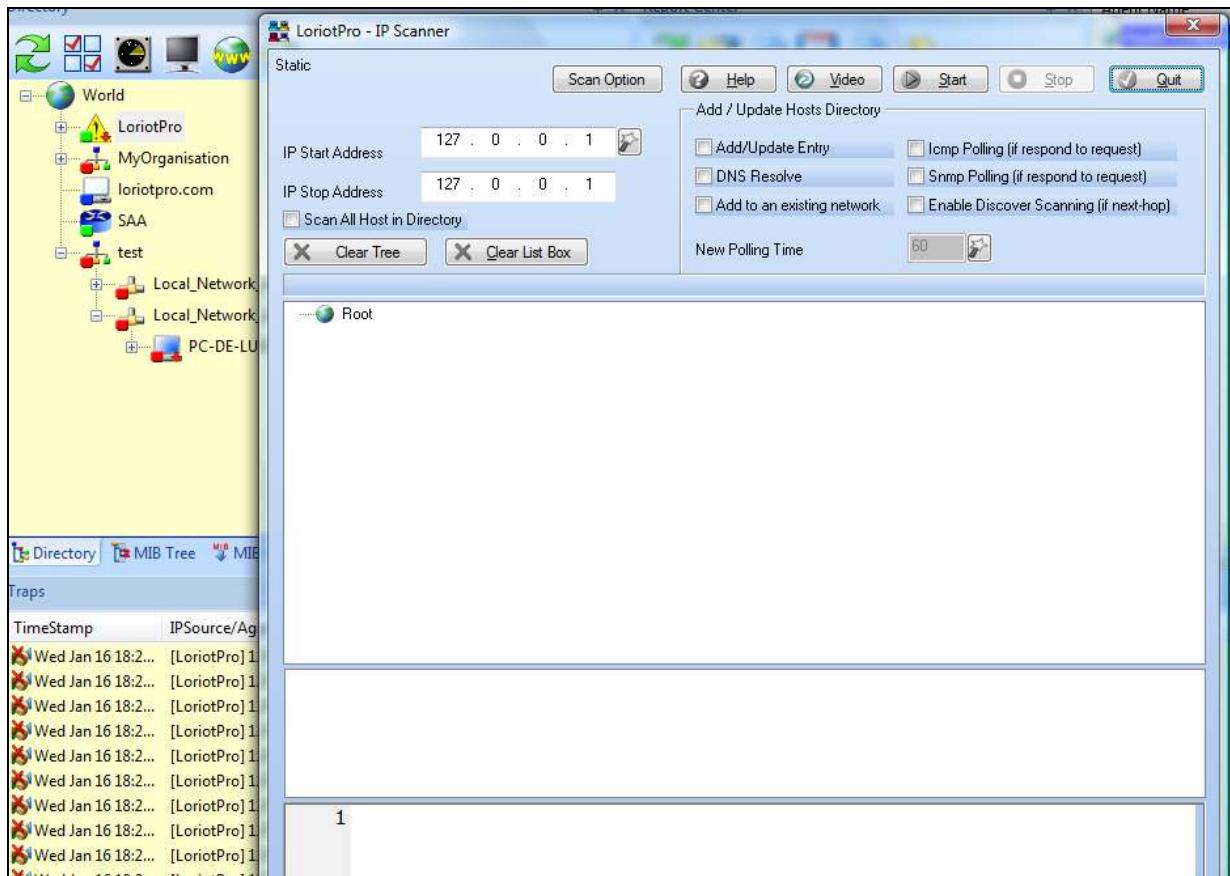
Le module IP Scanner utilise les mêmes ressources que le module IP Expert en ce qui concerne la partie Système Expert du module. Un script d'analyse et une stratégie écrite pour le module IP Expert pourront être réutilisés par ce module. La différence essentielle de ce module avec IP Expert est son utilisation manuelle, en fonction des besoins de l'administrateur pour scanner ou expertiser une portion du réseau ou un équipement unique.

### 16.2 Lancement d'IP Scanner associé à un équipement

Il faut sélectionner un équipement de la Directory et aller dans l'option du menu :

**Tools>IPScanner/Audit...**

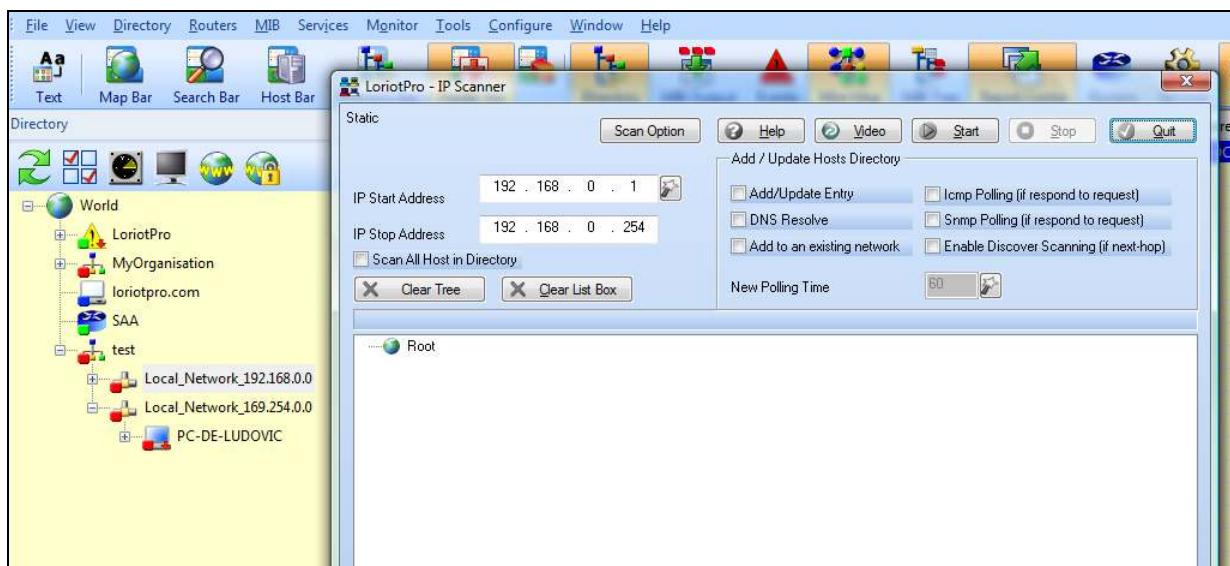




### 16.3 Lancement d'IP Scanner associé à un réseau

Il faut sélectionner le réseau dans la Directory et aller dans l'option du menu :

**Tools>IPScanner/Audit...**



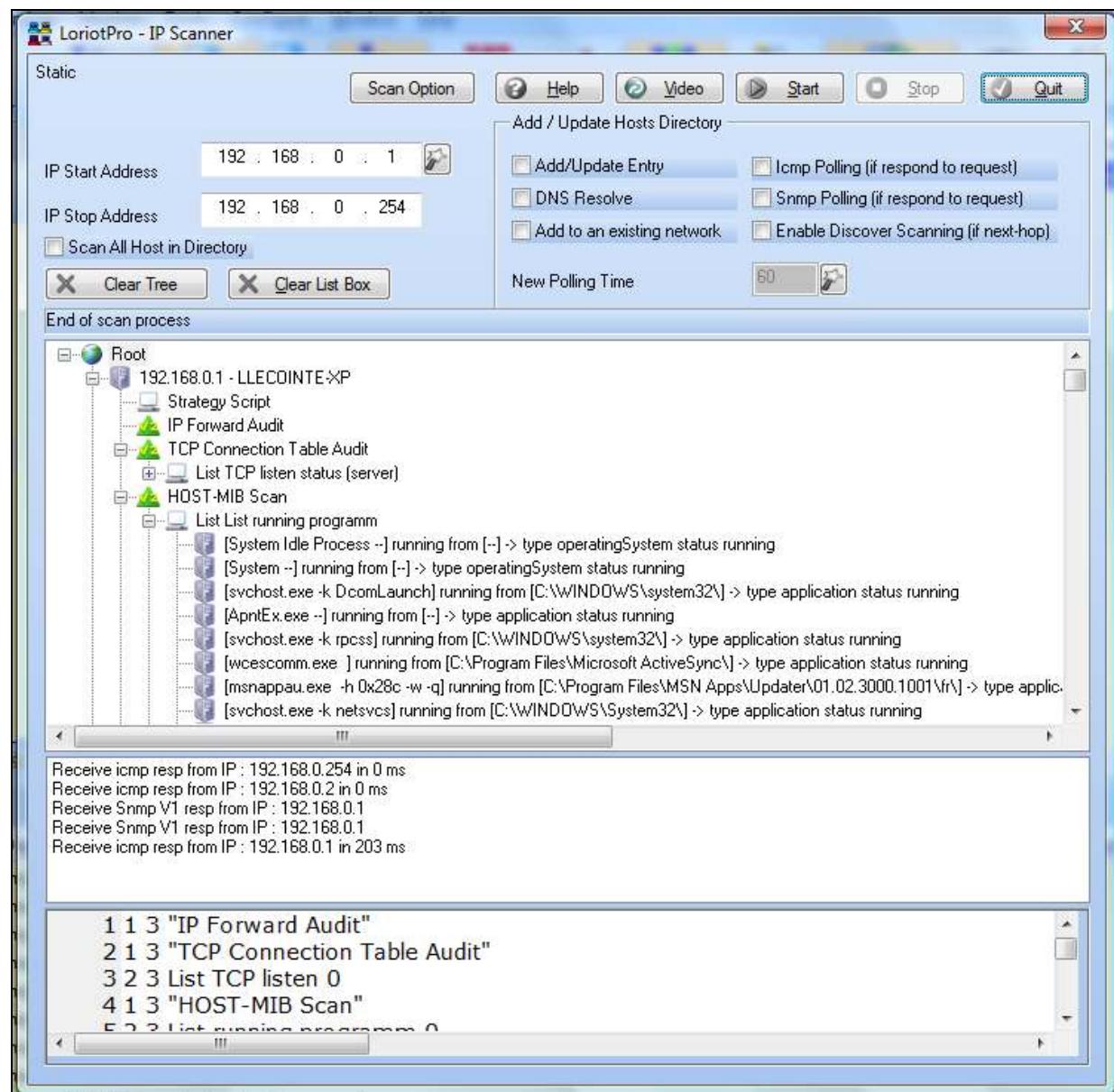
ou utiliser le menu contextuel du réseau dans la Directory.

La plage d'adresse IP du module IP Scanner est auto-configurée avec les paramètres du réseau sélectionné. On peut aussi utiliser les options de configuration du module pour sélectionner une plage adresses IP contigües d'un réseau.

### 16.3.1 Configuration du module IP Scanner

Le module IP Scanner comporte quatre zones :

1. La zone de configuration
2. La zone d'affichage des résultats dans un arbre structuré
3. La zone d'affichage des résultats des pollings basiques
4. La zone d'affichage des résultats liés aux scripts.



### 16.3.2 La zone de configuration

#### 16.3.2.1 Options - IP Start Address

L'adresse IP de départ du scan.

#### 16.3.2.2 Options - IP Stop Address

L'adresse IP de fin du scan

**Attention** 

L'adresse de fin doit être supérieure à l'adresse de départ.

L'adresse de départ et l'adresse de fin définissent une plage de scan, où chaque adresse IP sera scannée.

#### 16.3.2.3 Options - Scan All Host in Directory

Si vous cochez cette option, le module lancera un scan sur chaque équipement de la Directory (les options IP Start et IP Stop ne sont pas utilisées)

#### 16.3.2.4 Options - Clear Tree

Ce bouton permet d'effacer l'arbre des résultats.

#### 16.3.2.5 Options - Clear List Box

Ce bouton permet d'effacer la zone d'affichage des résultats de pollings basiques.

### 16.3.3 Section - Add / Update Hosts Directory

Cet ensemble d'options permet de demander au module d'utiliser les informations collectées pour modifier quelques paramètres des équipements analysés se trouvant dans la Directory ou d'insérer de nouveaux équipements. Ces options sont utilisées essentiellement avec la version standard de LoriotPro qui ne permet pas l'auto-configuration à travers le système expert.

#### 16.3.3.1 Options - AddUpdate Entry

Lors d'un scan d'une plage d'adresse IP, il est possible qu'un équipement ne faisant pas partie de la Directory soit détecté si cette option est cochée ; le nouvel équipement sera inséré dans la Directory dans le container par défaut. Si l'équipement existe déjà et répond au polling alors les paramètres de polling de cette zone seront utilisés pour reconfigurer l'équipement. Si cette option n'est pas cochée, l'ensemble des paramètres de la section 'Add / Update Hosts Directory' sont ignorés.

**Attention** 

Ces options ont été conservées pour rester compatibles avec des versions antérieures du logiciel. Il est préférable d'utiliser le nouveau module « Bulk configurator » pour reconfigurer en masse des équipements.

#### **16.3.3.2 Options - DNS Resolve**

Si un équipement répond, une requête DNS sera réalisée pour nommer l'équipement.

#### **16.3.3.3 Options - Add to existing network**

Si un nouvel équipement est détecté et inséré dans la Directory, le module essaie de l'insérer en dessous du container de type 'Network' correspondant à l'équipement, si il existe.

#### **16.3.3.4 Options - Icmp Polling (if respond to request)**

Modifie le paramètre de polling de l'équipement s'il répond au ping en validant le polling de type ping.

#### **16.3.3.5 Options - SNMP Polling (if respond to request)**

Modifie le paramètre de polling SNMP de l'équipement s'il répond aux requêtes SNMP en validant le polling de type SNMP.

#### **16.3.3.6 Options - Enable Discover Scanning (if next-hop)**

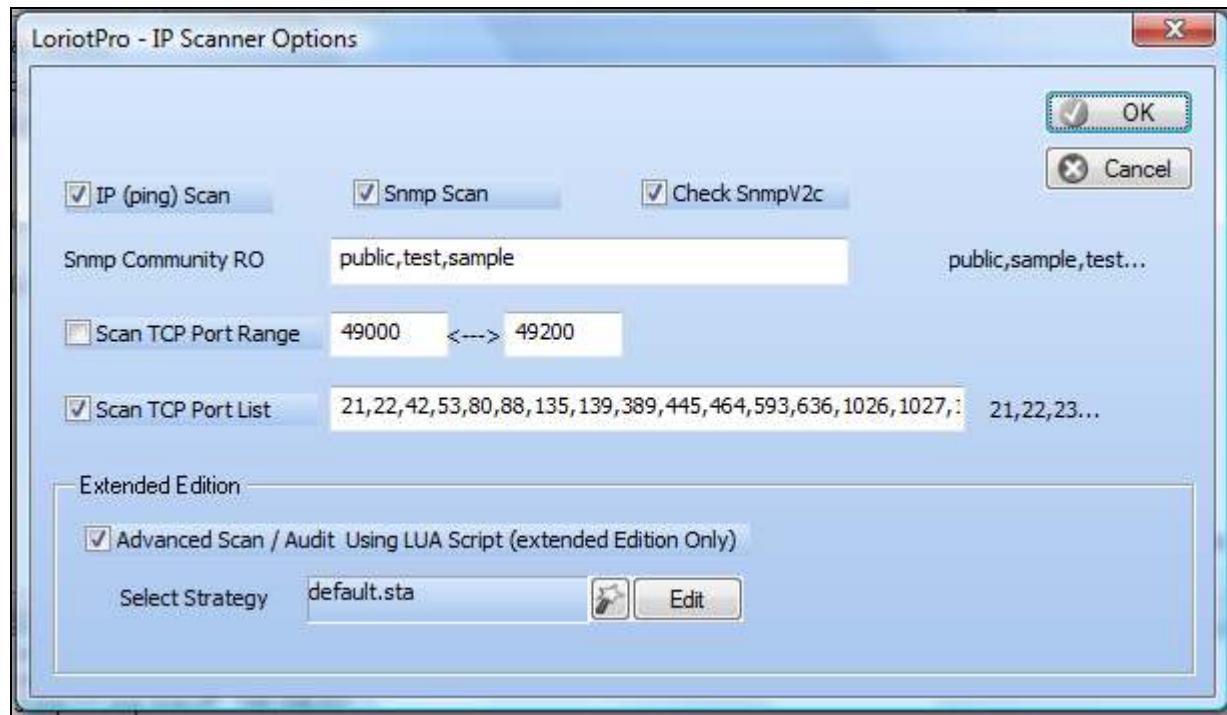
Si le module IP Scanner considère que cet équipement est un routeur alors il valide l'option 'Discover Scanning' permettant au module Discover de réaliser une analyse de l'équipement.

#### **16.3.3.7 Options - New Polling Time**

Il est possible de changer l'intervalle de polling de l'équipement.

### **16.3.4 Scan Option**

Le bouton 'Scan Option' ouvre la boîte de dialogue qui permet de configurer les paramètres de scans réalisés par le module.



#### 16.3.4.1 Option – IP (ping) Scan

Cette option réalise un pollig de type ping (ICMP) vers l'équipement et affiche le résultat dans l'arbre.

#### 16.3.4.2 Option – Snmp Scan

Cette option réalise un pollig de type ping (SNMP V1) vers l'équipement et affiche le résultat dans l'arbre.

#### 16.3.4.3 Option – Check SnmpV2c

L'option 'Snmp Scan' est cochée ; le module fera un polling de type SNMP V2c en plus pour tester si l'équipement supporte la version 2c de SNMP.

#### 16.3.4.4 Option – Snmp CommunityRO

L'option 'Snmp Scan' est cochée ; le module utilisera cet ensemble de communities pour contacter l'équipement.

Snmp Community RO	public,test,sample	public,sample,test...
-------------------	--------------------	-----------------------

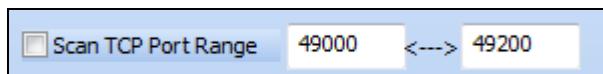
Il est possible de définir plusieurs communities ; dans ce cas, il faut écrire les communities à la suite dans le champ de saisie en les séparant avec une virgule.

#### Attention

Le caractère blanc n'est pas autorisé.

Il y aura autant de requêtes que de communities définies, le module s'arrête dans la liste lorsqu'une community est valide.

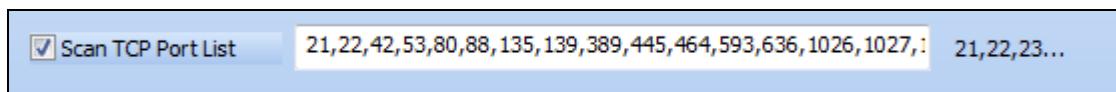
#### 16.3.4.5 Option – Scan TCP Port Range



On définit un scan de type TCP sur une plage de valeur 0 à 65535.

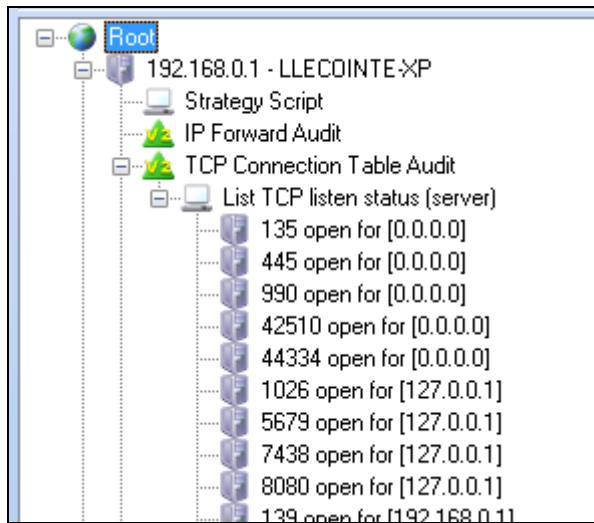
Cette option ne peut être exécutée que sur des équipements qui sont déjà dans l'arbre de résultats et il faut réaliser un polling ICMP ou SNMP positif pour que les scans 'TCP Range' soient déclenchés.

#### 16.3.4.6 Option – Scan TCP Port List



On définit une liste de port TCP à scanner ; dans ce cas, il faut écrire les numéros de port à la suite dans le champ de saisie en les séparant avec une virgule. Les valeurs possibles vont de 0 à 65535.

Cette option ne peut être exécutée que sur des équipements qui sont déjà dans l'arbre de résultats et il faut réaliser un polling ICMP ou SNMP positif pour que les scans 'TCP List' soient déclenchés.



#### Attention

L'usage de caractères blancs n'est pas autorisé.

#### 16.3.4.7      Option – Advanced Scan / Audit Using LUA Script

Cette option ne peut être exécutée que sur des équipements qui sont déjà dans la Directory et il faut réaliser un polling ICMP ou SNMP positif pour que les scans 'Advanced' soient déclenchés.

##### 16.3.4.7.1    Script IP Expert et stratégie associée

La case « Advanced Scan / Audit Using Script » permet de sélectionner une stratégie d'utilisation des scripts de type IP Expert.

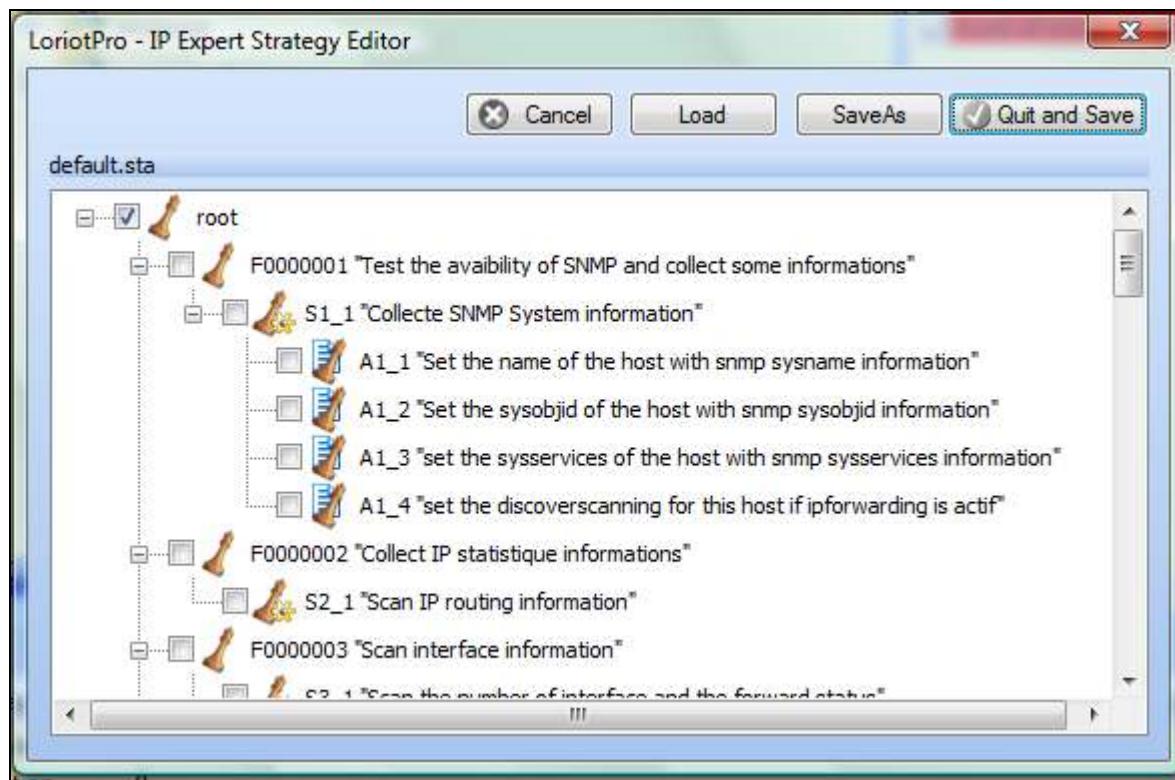
Les « scans » sont réalisés à partir de scripts situés dans le répertoire **bin/config/script/ipsscanner**. Les scripts sont « Open Source » et peuvent être modifiés avec l'éditeur de script fourni avec le logiciel. Chaque fichier de scripts contient des fonctions diverses d'audits, d'expertises, de configurations etc.. Les fonctions incluses dans les scripts sont structurées et auto répertoriées par le logiciel. LoriotPro fournit un module de gestion de ses fonctions et permet de définir des stratégies d'exploitations associées à un équipement ou un réseau. Basiquement le module « IP Scanner » utilise ces fonctions pour auto configurer les équipements et réaliser quelques audits. Il est possible d'utiliser les stratégies du module IP Expert et les fichiers de scrips du module. Le module « IP Scanner » utilise les mêmes scripts que le module « IP Expert ».

##### Attention

Le module « IP Scanner » utilise uniquement la stratégie définie dans le champ d'option (« default.sta » par défaut) que vous pouvez bien entendu modifier selon les besoins de votre entreprise. Si le paramètre de stratégie d'un équipement ou d'un réseau est validé, il ne sera pas utilisé par le module IP Scanner. Ceci permet de forcer l'exécution d'une stratégie de scan. A l'inverse le module IP Expert, lui, utilise la stratégie définie dans les paramètres de l'équipement ou du réseau associé en priorité.

Vous pouvez bien sûr créer une stratégie différente à partir des outils de gestion.

Le bouton « Edit » permet d'afficher la boîte de dialogue de sélection de votre stratégie « IP Scanner »



## 17 Filtre d'Events et de Trap scriptés

### 17.1 Introduction

LoriotPro Extended Edition offre plusieurs nouveaux types de filtres d'événements, dont un supporte l'exécution d'un script associé à un événement (Event LoriotPro) ou un TRAP (SNMP). L'exécution d'un script peut bloquer la réception des événements, le principe de fonctionnement de ce nouveau filtre est celui d'une queue de traitement classique.

- Un Event ou un Trap est collecté par le noyaux .
- Un filtre de traitement scripté est sélectionné en fonction des paramètres de l'Event ou du Trap SNMP.
- Un fichier de traitement est généré dans le répertoire **bin/EventScripter**
- Le service 'Event / Trap Scripter' consulte à intervalles réguliers les fichiers du répertoire **bin/EventScripter** et les exécute.

### 17.2 La gestion d'événements

Les gestionnaires d'événements de LoriotPro centralisent l'information et informe l'administrateur à travers des filtres configurés. Les principales tâches des modules de réception d'événements sont les suivantes :

- Alarms sur changement de statut, sur réception de TRAP, sur réception de log (Syslog), sur réception d'évent, sur déclenchement de seuil dans un script d'audit, sur expertise d'un module (plugin ou script).
- Il est possible de rediriger les alarmes vers d'autres superviseurs (LoriotPro Centralisé) ou produit tiers (Syslog, TRAP).
- Intègre une gestion de filtres avancée
- Les événements peuvent modifier l'état d'objets graphiques des vues (ActiveView), repositionnement, changement de couleur, changement de visuel (Clipart)...

Pour optimiser la gestion des événements, LoriotPro V5 dispose de plusieurs moteurs de réception spécialisés par protocole.

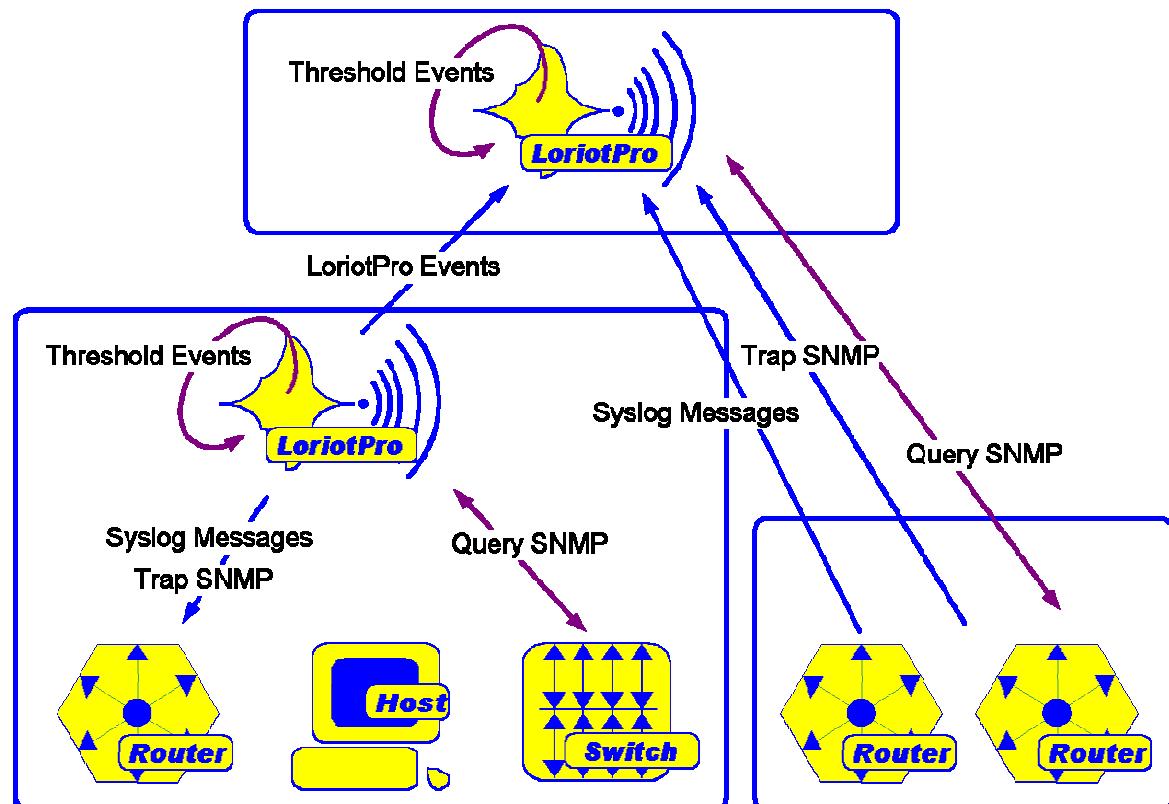
### 17.3 Types généraux d'événements

Il y a plusieurs types d'événements génériques

- Des événements envoyés par des équipements de façon aléatoire (pour LoriotPro)
- Des événements envoyés par les modules de LoriotPro vers le moteur de gestion des événements.
- Des événements envoyés par un autre LoriotPro vers le LoriotPro local.
- Des événements qui sont des réponses aux pollings (Asynchrone) générés par LoriotPro. (Traité dans un autre support)

Pour optimiser la gestion des événements, LoriotPro V5 dispose de plusieurs moteurs de réception spécialisés par protocole et d'une capacité de répartir la réception des événements sur plusieurs machines (plusieurs licences).

LoriotPro supporte un mode centralisé et distribué des événements symbolisés par le dessin suivant.



Les processus mis en œuvre par chaque moteur de réception suivent les logiques suivantes.

- *Réception / Collecte (temps réel)*
  - Chaque information est traitée dans une logique FIFO
  - Les queues de réception sont de 5Mo (+ 10000 paquets 512 octets)
- *Traitements des informations reçues*
  - Existence de filtres associés à l'événement ?
  - L'équipement concerné est-il dans la Directory ?
  - Compteur de Filtre, limitations ?
  - Si un filtre est trouvé, réalisation de la tâche définie
  - Regarde si un autre filtre existe pour l'événement ?
  - Si oui boucle, si non passe au paquet suivant.

**Attention** 

**Le traitement d'un événement est plus long que sa réception !!**

- Les queues de 5Mo sont là pour temporiser le traitement des événements et assurer le service.
- Cependant, sur de très grosses plateformes, il peut arriver que le flux des événements ne laisse pas de temps à un traitement en temps réel. Il y a alors un décalage temporel visible entre le moment de réception et le traitement. Pour optimiser ce problème commun à tous les logiciels de ce type, il faut utiliser des traitements le moins bloquant possible ou adopter une architecture répartie. Utiliser une machine plus puissante peut suffire dans certains cas.

## 17.4 Filtre d'événement LoriotPro Scripté

Le filtre d'événement scripté est configurable via un module Wizard pour faciliter la tâche de l'administrateur. Il s'inscrit dans la stratégie globale du logiciel et se configure comme les autres filtres.

**Attention** 

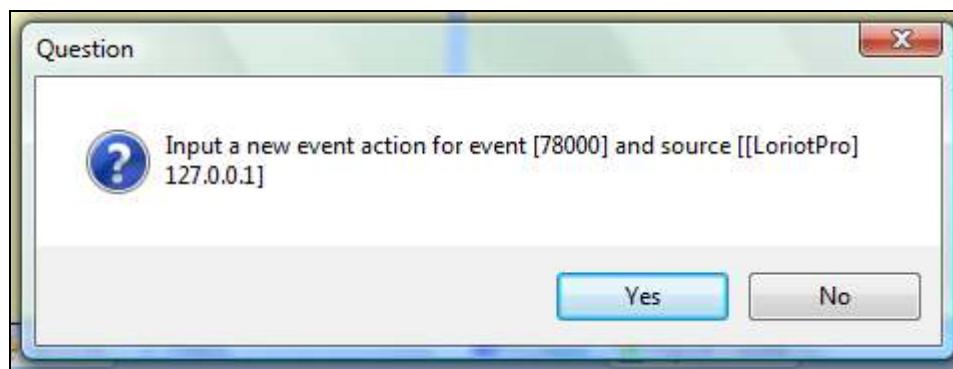
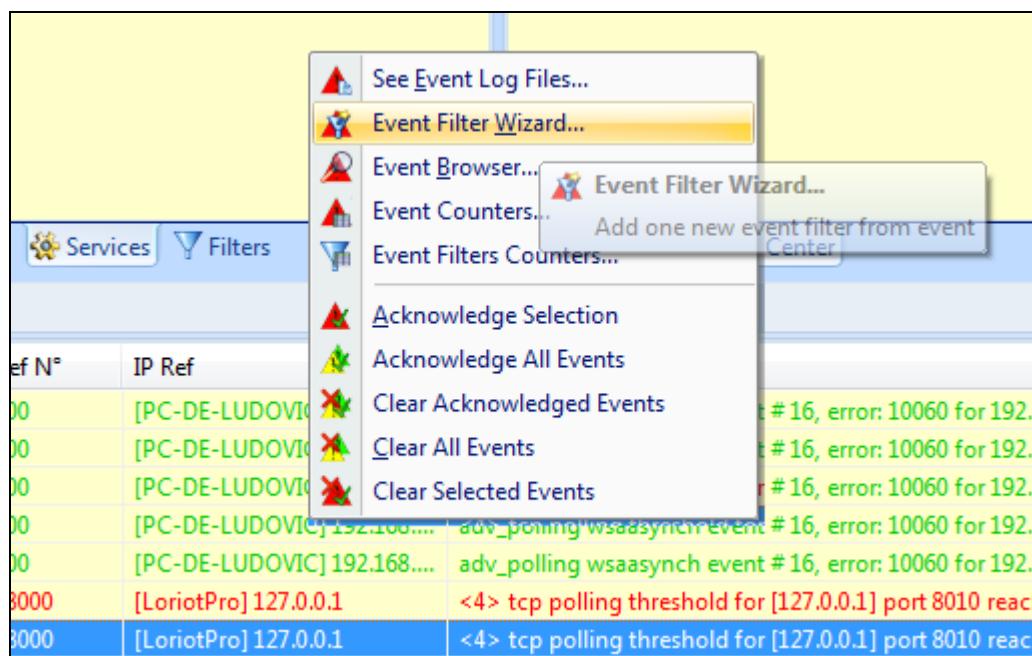
Il faut créer votre script avant de réaliser un filtre (voir le paragraphe sur la création d'un script de filtre).

### 17.4.1 Wizard d'insertion d'un Event LoriotPro

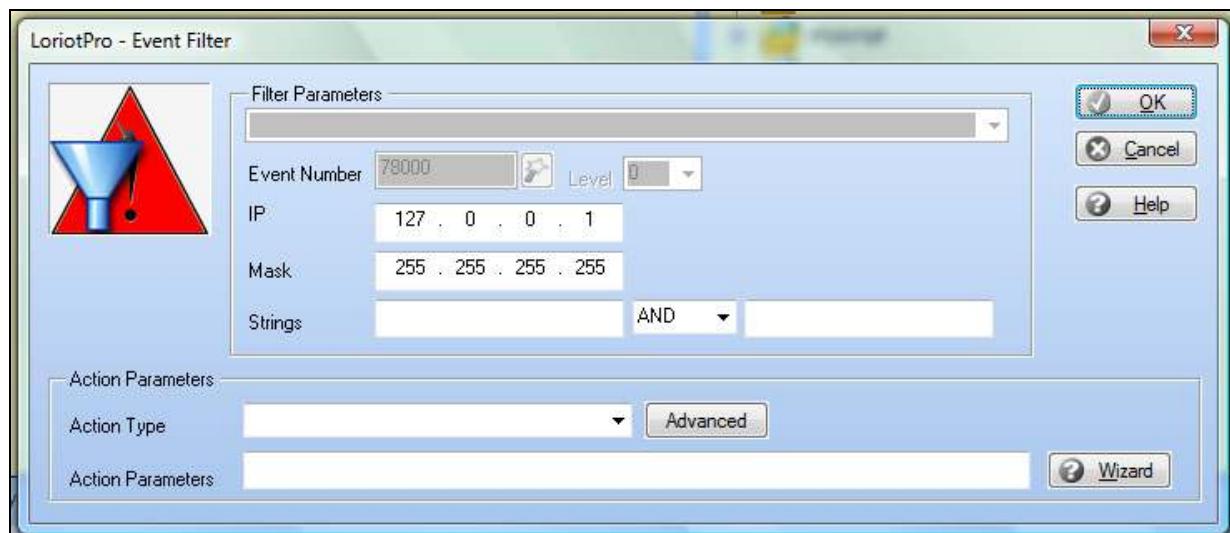
Nous allons prendre un exemple pour faciliter la compréhension de la configuration du filtre. Nous allons lancer un script sur réception d'un Event référencé 78000.

Events				
TimeStamp	LoriotPro ...	Ref N°	IP Ref	Alert
▲ Fri Jan 18 15:18:46 2...	Local	600	[LoriotPro] 127.0.0.1	adv_audit: 127.0.0.1 100(0) audit the storage unit of the host is down
▲ Fri Jan 18 15:18:44 2...	Local	600	[LoriotPro] 127.0.0.1	warning host [127.0.0.1] as low disk c:\ label: serial number ecc14f60 : reached 83%
▲ Fri Jan 18 15:18:43 2...	Local	78000	[LoriotPro] 127.0.0.1	<4> tcp polling threshold for [127.0.0.1] port 8010 reached down
▲ Fri Jan 18 15:18:42 2...	Local	600	[LUTEUS02] 192.168.0.6	adv_polling wsaasynch event # 16, error: 10060 for 192.168.0.6:21
▲ Fri Jan 18 15:18:40 2...	Local	78000	[LoriotPro] 127.0.0.1	<4> tcp polling threshold for [127.0.0.1] port 8010 reached down
▲ Fri Jan 18 15:18:39 2...	Local	78000	[LoriotPro] 127.0.0.1	<4> tcp polling threshold for [127.0.0.1] port 8010 reached down
▲ Fri Jan 18 15:18:38 2...	Local	600	[DC-DE-LUDOVIC1102] 168	adv_polling wsaasynch event # 16, error 10038 for 192.168.0.5:135

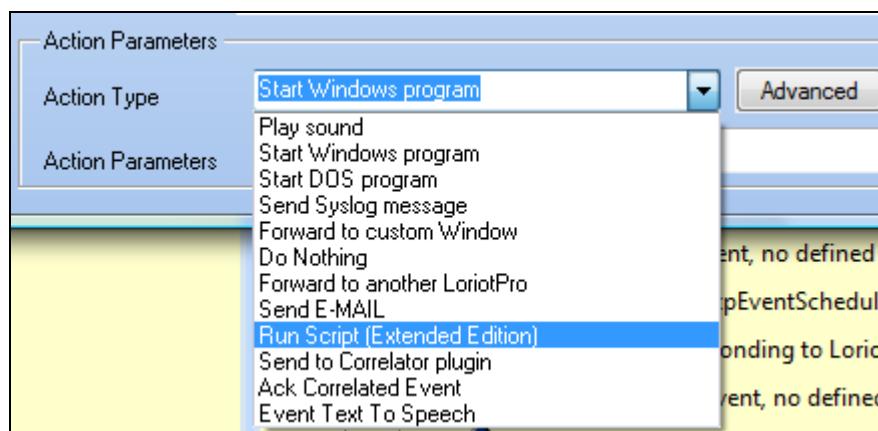
Nous utilisons le menu contextuel sur la ligne de l'événement reçu dans la listbox de réception des Events. (Pour simplifier la configuration)



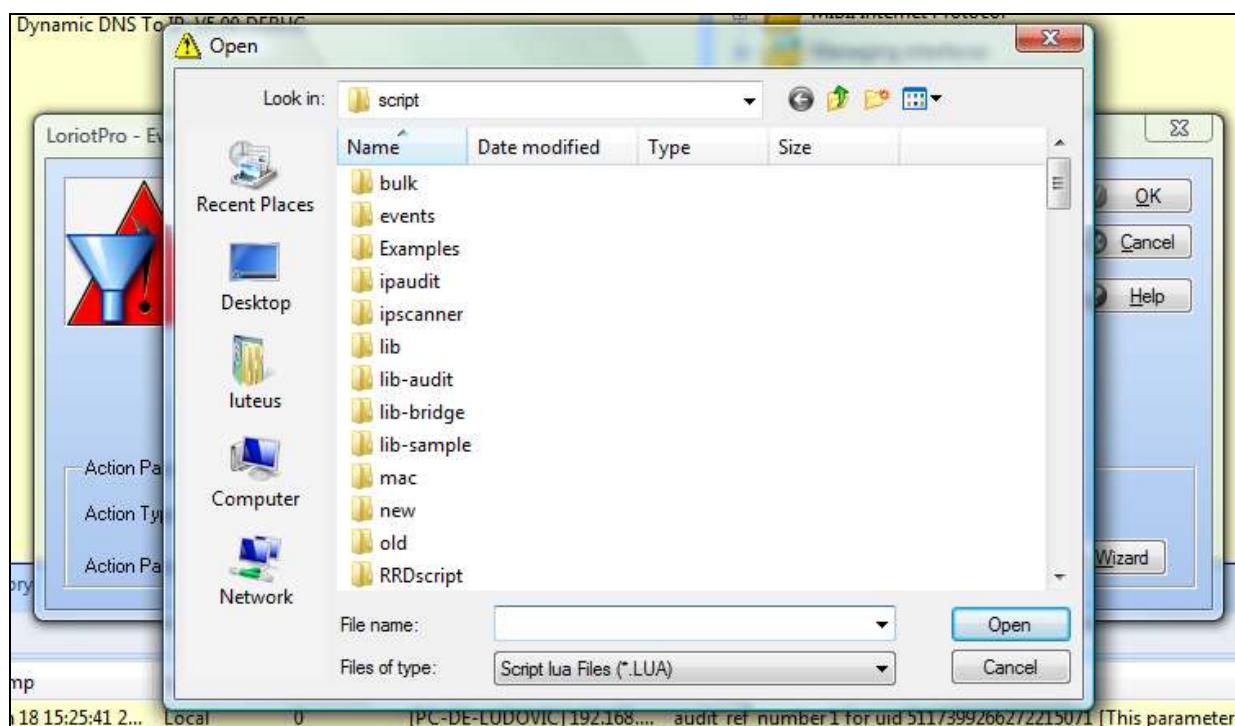
Faire 'Yes'



Le Wizard de création de filtre d'Event est auto-configuré. Dans le ComboBox 'Action Type' on sélectionne l'option '**'Run Script (Extended Edition)**'.



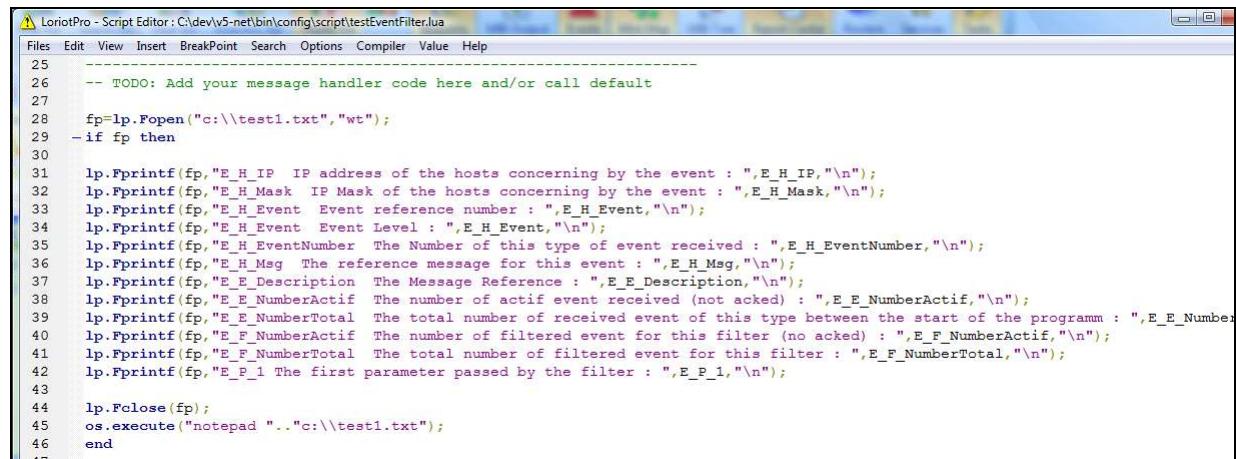
Le Wizard vous demande de sélectionner le fichier de script qui sera lancé avec le filtre, une boîte de dialogue apparaît.



Dans notre exemple, nous allons choisir le fichier **testEventFilter.lua**. qui sauve dans un fichier les paramètres et les affiche dans notepad.

# LoriotPro V5.00 Extended Edition – Supervision et gestion SNMP par scripts LUA

## Introduction au System Expert et aux scripts

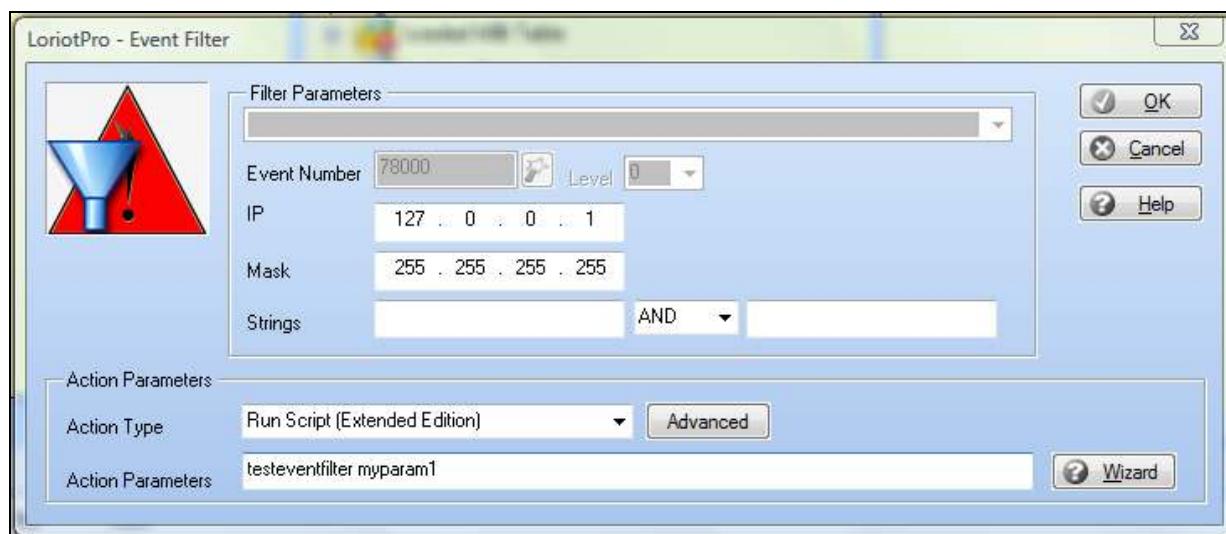


```
25 -----
26 -- TODO: Add your message handler code here and/or call default
27
28 fp=lp.Fopen("c:\\test1.txt","wt");
29 -if fp then
30
31 lp.Fprintf(fp,"E_H_IP IP address of the hosts concerning by the event : ",E_H_IP,"\n");
32 lp.Fprintf(fp,"E_H_Mask IP Mask of the hosts concerning by the event : ",E_H_Mask,"\n");
33 lp.Fprintf(fp,"E_H_Event Event reference number : ",E_H_Event,"\n");
34 lp.Fprintf(fp,"E_H_Event Event Level : ",E_H_Event,"\n");
35 lp.Fprintf(fp,"E_H_EventNumber The Number of this type of event received : ",E_H_EventNumber,"\n");
36 lp.Fprintf(fp,"E_H_Msg The reference message for this event : ",E_H_Msg,"\n");
37 lp.Fprintf(fp,"E_E_Description The Message Reference : ",E_E_Description,"\n");
38 lp.Fprintf(fp,"E_E_NumberActif The number of actif event received (not acked) : ",E_E_NumberActif,"\n");
39 lp.Fprintf(fp,"E_E_NumberTotal The total number of received event of this type between the start of the programm : ",E_E_NumberTotal,"\n");
40 lp.Fprintf(fp,"E_F_NumberActif The number of filtered event for this filter (no acked) : ",E_F_NumberActif,"\n");
41 lp.Fprintf(fp,"E_F_NumberTotal The total number of filtered event for this filter : ",E_F_NumberTotal,"\n");
42 lp.Fprintf(fp,"E_P_1 The first parameter passed by the filter : ",E_P_1,"\n");
43
44 lp.Fclose(fp);
45 os.execute("notepad ..\"c:\\test1.txt\"");
46 end
47
```

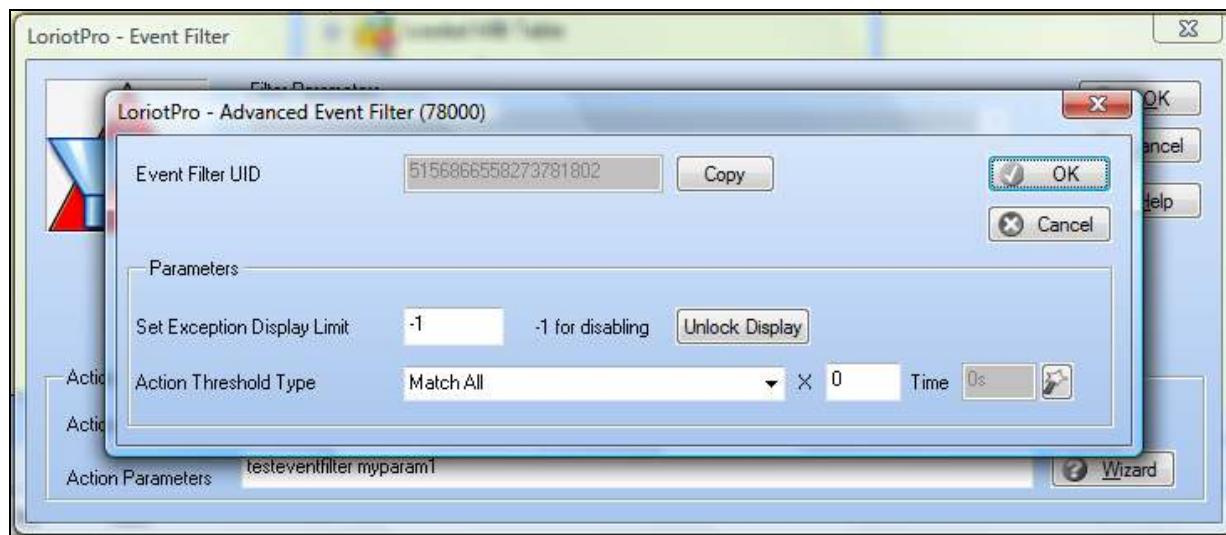
```
-----  
-- Start of your program -----  
-----  
-- TODO: Add your message handler code here and/or call default  
  
fp=lp.Fopen("c:\\test1.txt","wt");
if fp then
    lp.Fprintf(fp,"E_H_IP IP address of the hosts concerning by the event : ",E_H_IP,"\n");
    lp.Fprintf(fp,"E_H_Mask IP Mask of the hosts concerning by the event : ",E_H_Mask,"\n");
    lp.Fprintf(fp,"E_H_Event Event reference number : ",E_H_Event,"\n");
    lp.Fprintf(fp,"E_H_Event Event Level : ",E_H_Event,"\n");
    lp.Fprintf(fp,"E_H_EventNumber The Number of this type of event received : ",E_H_EventNumber,"\n");
    lp.Fprintf(fp,"E_H_Msg The reference message for this event : ",E_H_Msg,"\n");
    lp.Fprintf(fp,"E_E_Description The Message Reference : ",E_E_Description,"\n");
    lp.Fprintf(fp,"E_E_NumberActif The number of actif event received (not acked) : ",E_E_NumberActif,"\n");
    lp.Fprintf(fp,"E_E_NumberTotal The total number of received event of this type between the start of the programm : ",E_E_NumberTotal,"\n");
    lp.Fprintf(fp,"E_F_NumberActif The number of filtered event for this filter (no acked) : ",E_F_NumberActif,"\n");
    lp.Fprintf(fp,"E_F_NumberTotal The total number of filtered event for this filter : ",E_F_NumberTotal,"\n");
    lp.Fprintf(fp,"E_P_1 The first parameter passed by the filter : ",E_P_1,"\n");
    lp.Fclose(fp);
    os.execute("notepad ..\"c:\\test1.txt\"");
end
```

### Attention

Ce script est juste un exemple de gestion des variables. Il n'est pas souhaitable de lancer un notepad sur chaque réception d'un événement car cela provoquerait un 'Deni Of Service' du logiciel. En effet, les boîtes de dialogue non fermées satureraient la mémoire de la machine. De plus, le programme notepad.exe est lancé en mode Modal, bloquant ainsi l'exécution du fichier suivant et déclenchant le timeout de 2 secondes, ce qui provoque le 'kill' du script.

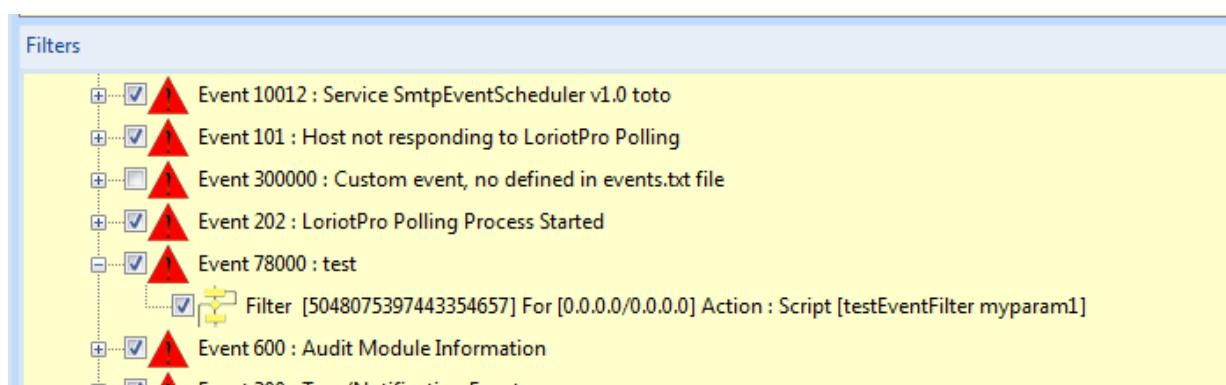


Le nom du fichier de script apparaît dans le champ ‘Action Parameters’, nous avons ajouté une variable ‘myparam1’.



Nous finalisons les paramètres d’occurrence du filtre.

Le filtre est maintenant en place dans l’arbre des filtres d’Event.



#### 17.4.1.1 Exécution du filtre

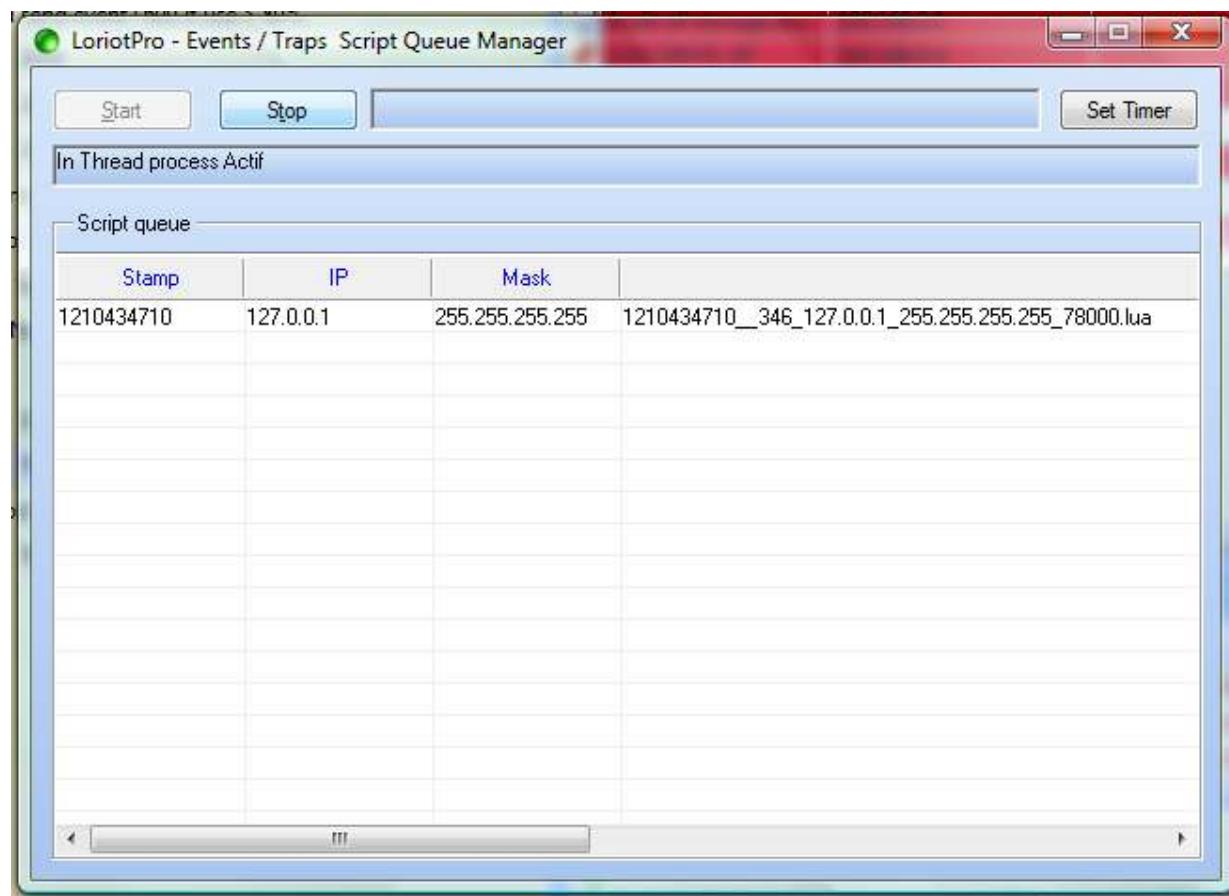
Un événement 78000 correspondant au filtre est reçu par le module d'événements.

Events				
TimeStamp	LoriotPro ...	Ref N°	IP Ref	Alert
▲ Fri Jan 18 18:26:14 2...	Local	0	[PC-DE-LUDOVIC]192.168.0.5	audit_ref_number 1 for uid 5117399266272215071 [This parameter is se
▲ Fri Jan 18 18:26:11 2...	Local	0	[PC-DE-LUDOVIC]192.168.0.5	audit_ref_number 1 for uid 5117399343581626400 [This parameter is se
▲ Fri Jan 18 18:26:06 2...	Local	78000	[LoriotPro]127.0.0.1	<4> tcp polling threshold for [127.0.0.1] port 8010 reached down
▲ Fri Jan 18 18:25:58 2...	Local	78000	[LoriotPro]127.0.0.1	<4> tcp polling threshold for [127.0.0.1] port 8010 reached down
▲ Fri Jan 18 18:25:54 2...	Local	78000	[LoriotPro]127.0.0.1	<4> tcp polling threshold for [127.0.0.1] port 8010 reached down

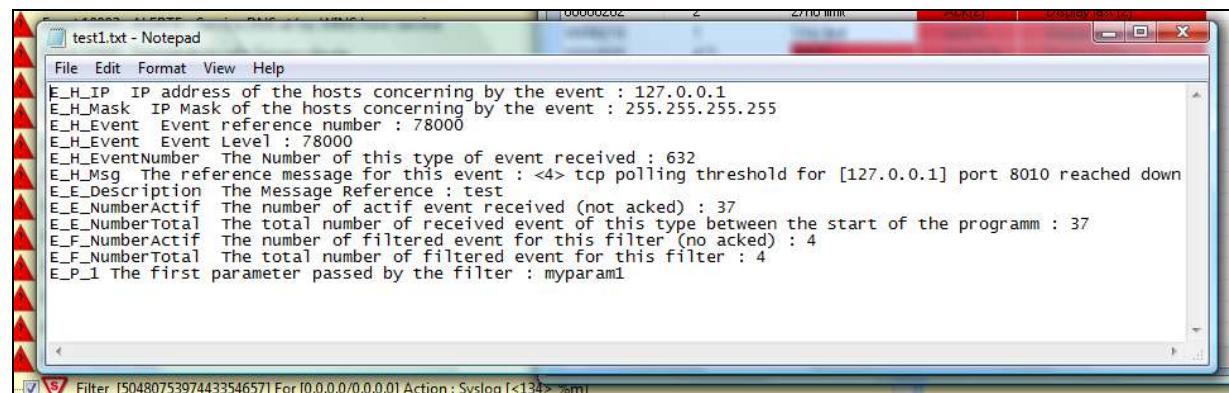
Un fichier dynamique est créé par le noyau de LoriotPro dans le répertoire **bin/EventScripter**

isk (C:) ▶ dev ▶ v5-net ▶ bin ▶ EventScripter ▶				Search
en ▶ Burn				
Name	Date modified	Type	Size	
1200677166_373_127.0.0.1_255.255.255.255_78000.lua	18/01/2008 18:26	LUA File	1 KB	
1200677158_372_127.0.0.1_255.255.255.255_78000.lua	18/01/2008 18:26	LUA File	1 KB	
1200677154_371_127.0.0.1_255.255.255.255_78000.lua	18/01/2008 18:25	LUA File	1 KB	
1200676864_370_127.0.0.1_255.255.255.255_78000.lua	18/01/2008 18:21	LUA File	1 KB	
1200676859_369_127.0.0.1_255.255.255.255_78000.lua	18/01/2008 18:21	LUA File	1 KB	
1200676853_368_127.0.0.1_255.255.255.255_78000.lua	18/01/2008 18:20	LUA File	1 KB	
error	18/01/2008 18:26	File Folder		

Le service détecte et lance le fichier (script).



Le script est exécuté, un fichier est créé par le script puis lancé dans notepad.exe.



**Attention**

Dans ce cas, le programme notepad.exe est exécuté en mode Modal, il reste plus de deux secondes, ce qui provoque le 'Kill' du thread associé au script.

Warning Service Lock, Kill Thread

Le module ‘Event Filter Counter’ montre que le filtre a été réalisé 12 fois.

Event Filter Counters				Display Options
For Event Number	Total Matched	Matched Since Last Clear Counter	Last Filtered At	Filter Parameters
00000300				Event 300 : Trap/Notification Event
00000600	445	445	Fri Jan 18 18:32:48 2008	Filter [5139352858461732904] For [0.0.0.0/0.0.0.0] Action : Delay [100 555 %i 255.255.255.255 6000 4 toto va bien] Filter [51177913633901437967] For [127.0.0.1/255.255.255.255] Action : Syslog [127.0.0.1 %m]
00000800	114	114	Fri Jan 18 18:32:27 2008	Event 78000 : test
00078000	12	12	Fri Jan 18 18:29:57 2008	Filter [5156866558273781802] For [127.0.0.1/255.255.255.255] Action : Script [testeventfilter myparam1] Filter [504807539744354657] For [0.0.0.0/0.0.0.0] Action : Syslog [<134> %m]
00000202	45	45	Fri Jan 18 18:29:57 2008	Event 202 : LoriotPro Policy Process Started

#### 17.4.2 Wizard d'insertion d'un Filtre de Trap SNMP

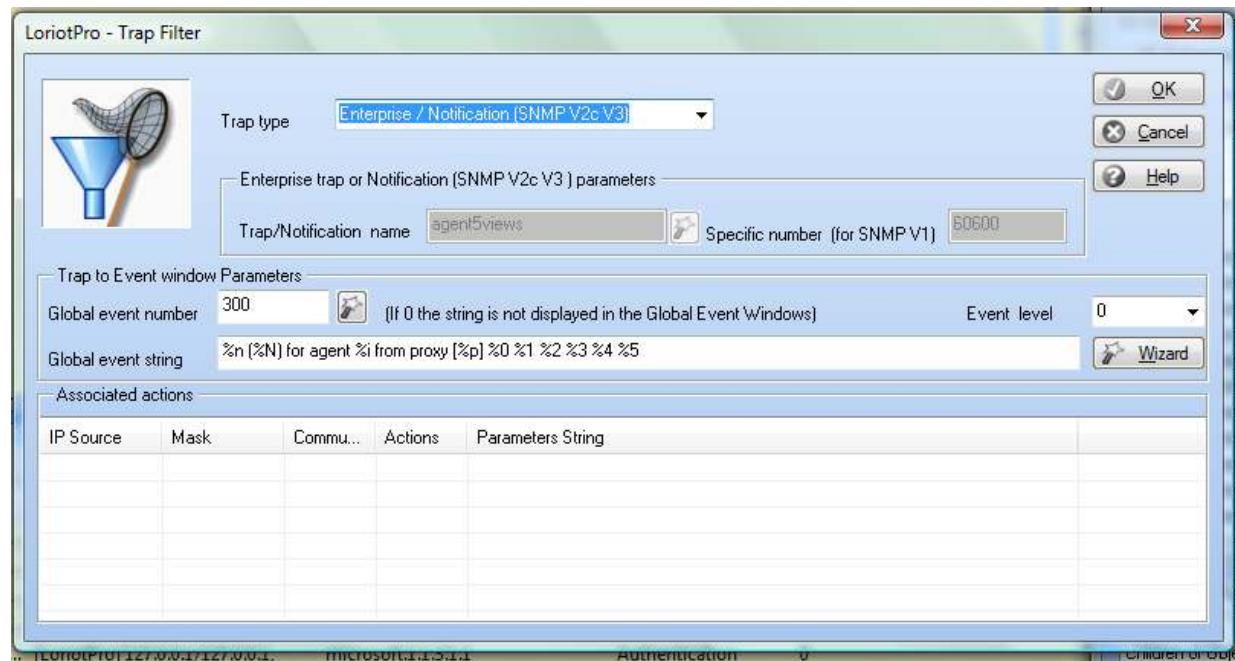
De la même façon, on lance le Wizard associé à la réception d'un Trap dans le ‘Workspace des Traps’ .

TimeStamp	IPSource/Agent	Object Identifier	Generic Trap	Enterprise Specific	Value0,OID	Value1,OID	Value2,OID	Value3,OID
Fri Jan 18 18:46:4...	[LoriotPro]127.0.0.1/127.0.0.1,	agent5views	Enterprise	60600 ApplipAgen...	string,applipagent...	string,applipagent...	string,applipagent...	string,applipagent...
Fri Jan 18 18:46:4...	[LoriotPro]127.0.0.1/127.0.0.1,	agent5views	Enterprise	60600 ApplipAgen...	string,applipagent...	string,applipagent...	string,applipagent...	string,applipagent...
X Fri Jan 18 18:46:4...	[LoriotPro]127.0.0.1/127.0.0.1	microsoft.1.1.3.1	Authentication	0				

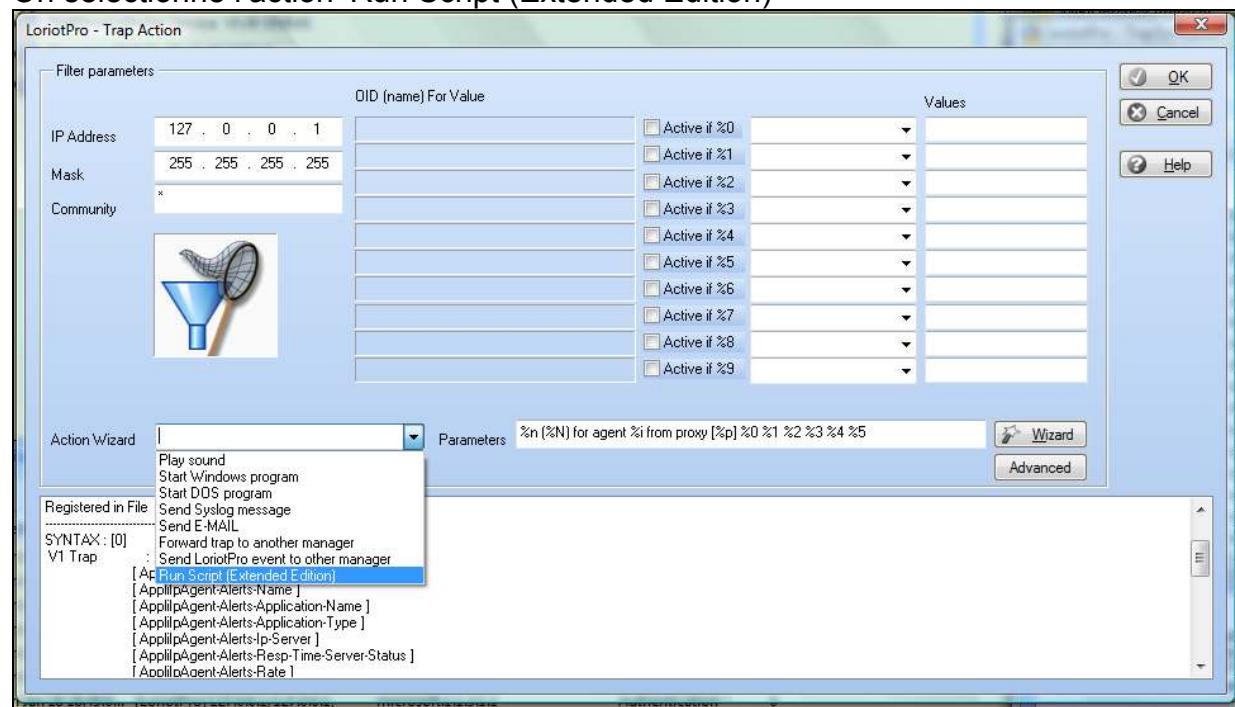


Ce trap n'est pas encore filtré, on l'ajoute à l'arbre des traps.

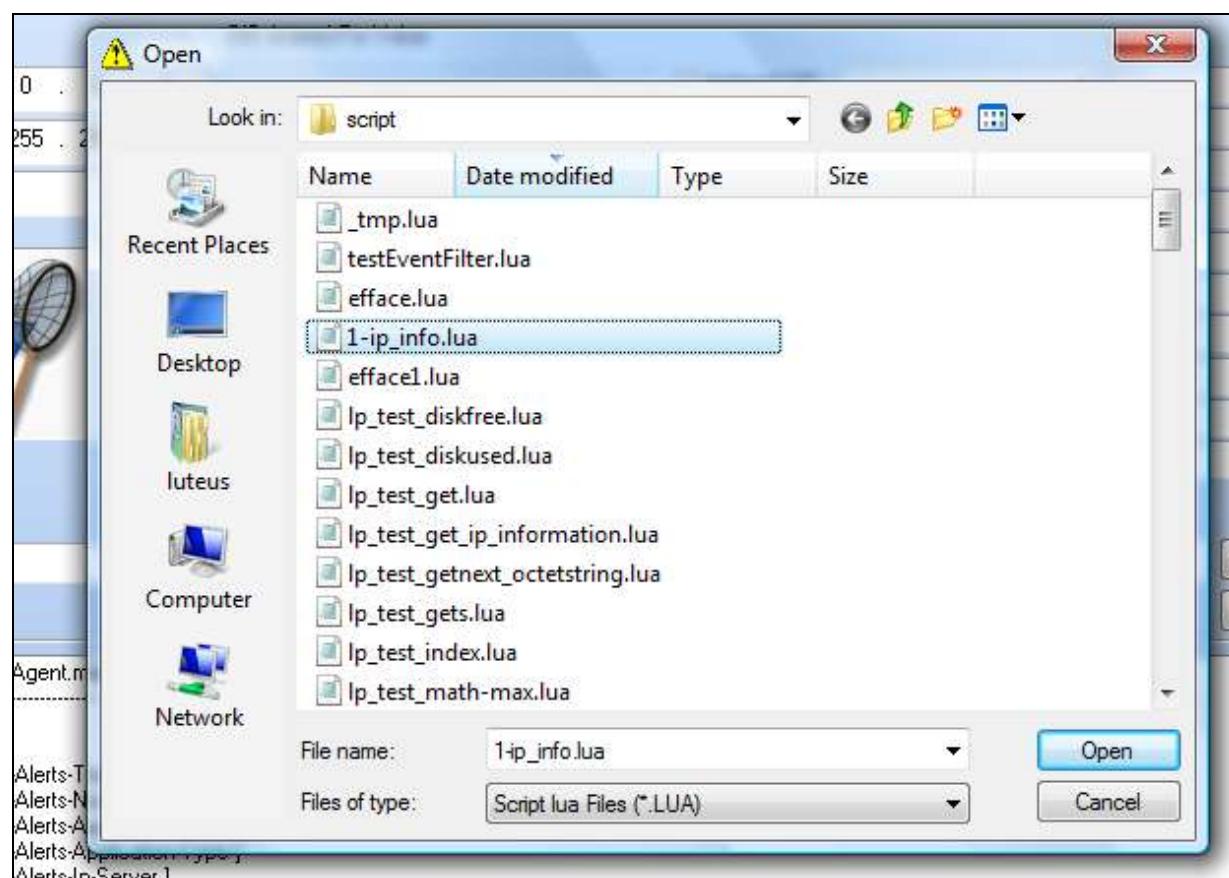
## LoriotPro V5.00 Extended Edition – Supervision et gestion SNMP par scripts LUA Introduction au System Expert et aux scripts



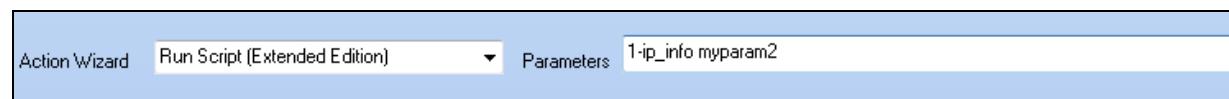
On sélectionne l'action ‘Run Script (Extended Edition)’



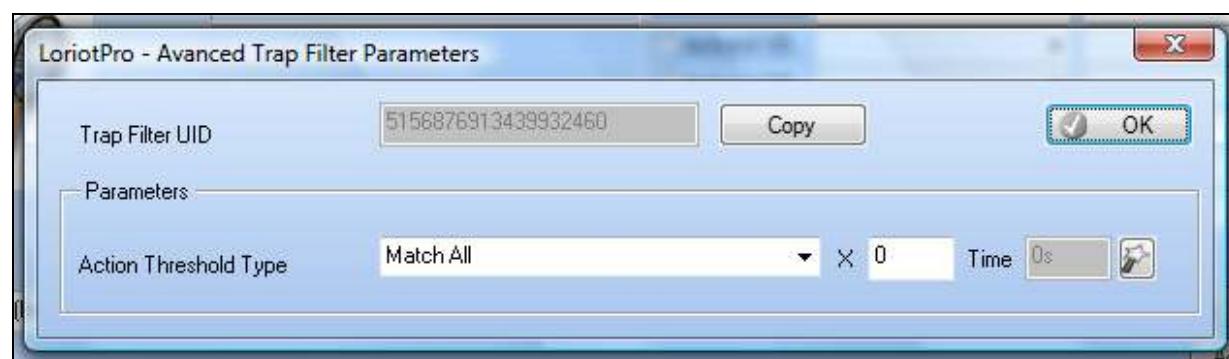
Un wizard nous propose de sélectionner notre fichier de script. La procédure restante est la même que celle du wizard d’Event.



Un fichier de script est sélectionné et le champ ‘ Parameters’ est renseigné, nous avons ajouté une variable.



L'occurrence d'exécution du filtre est validée.





Le nouveau filtre pour le TRAP agent5views est ajouté à l'arbre des filtres de Traps.

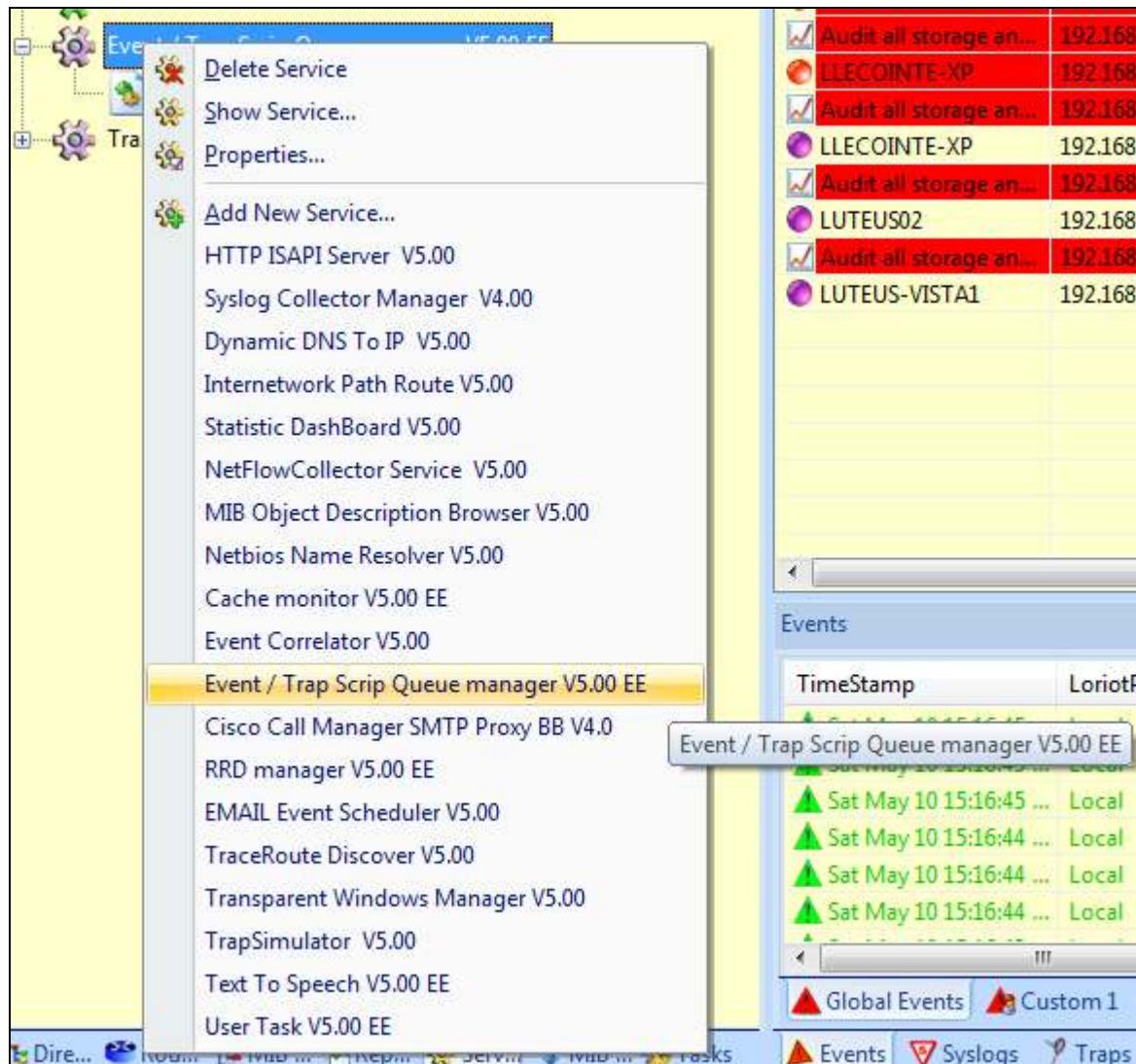
## 17.5 Lancement du service ‘Event / Trap Script Queue manager’

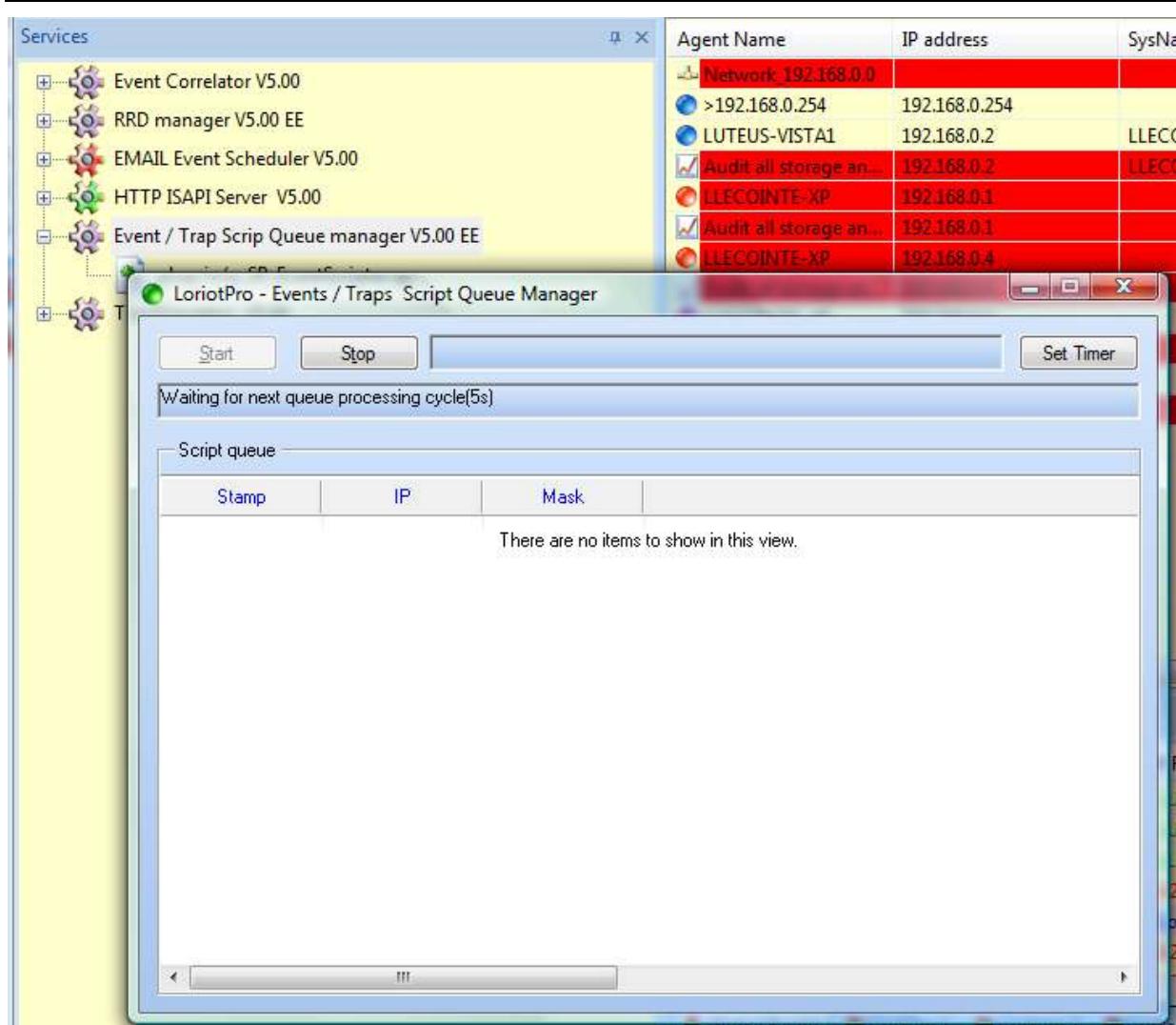
Le service ‘Event / Trap Script Queue manager’ est indispensable pour traiter les fichiers dynamiques. Il s’installe comme tous les plugins de service dans le ‘Workspace Service’.

### Attention

Les fichiers sont traités dans l’ordre de génération par le module. Si un script bloque, il arrête le traitement des autres fichiers. Le service intègre une sécurité qui stoppe l’exécution de tous les scripts qui ont une durée d’exécution supérieure à 2 secondes.

LoriotPro V5.00 Extended Edition – Supervision et gestion SNMP par scripts LUA  
Introduction au System Expert et aux scripts





#### 17.5.1.1 Option – Set Timer

Par défaut le service ne se fait pas en temps réel mais scanne le répertoire toutes les 20 secondes, il est possible de réduire ou augmenter cet intervalle avec l'option Set Timer.



#### 17.5.1.2 Option – Stop

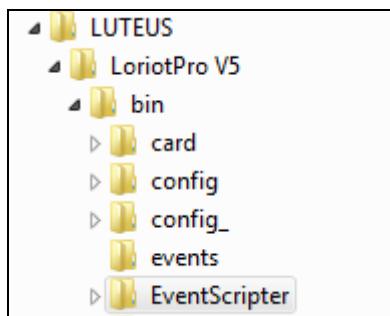
Stoppe le service.

#### 17.5.1.3 Option – Start

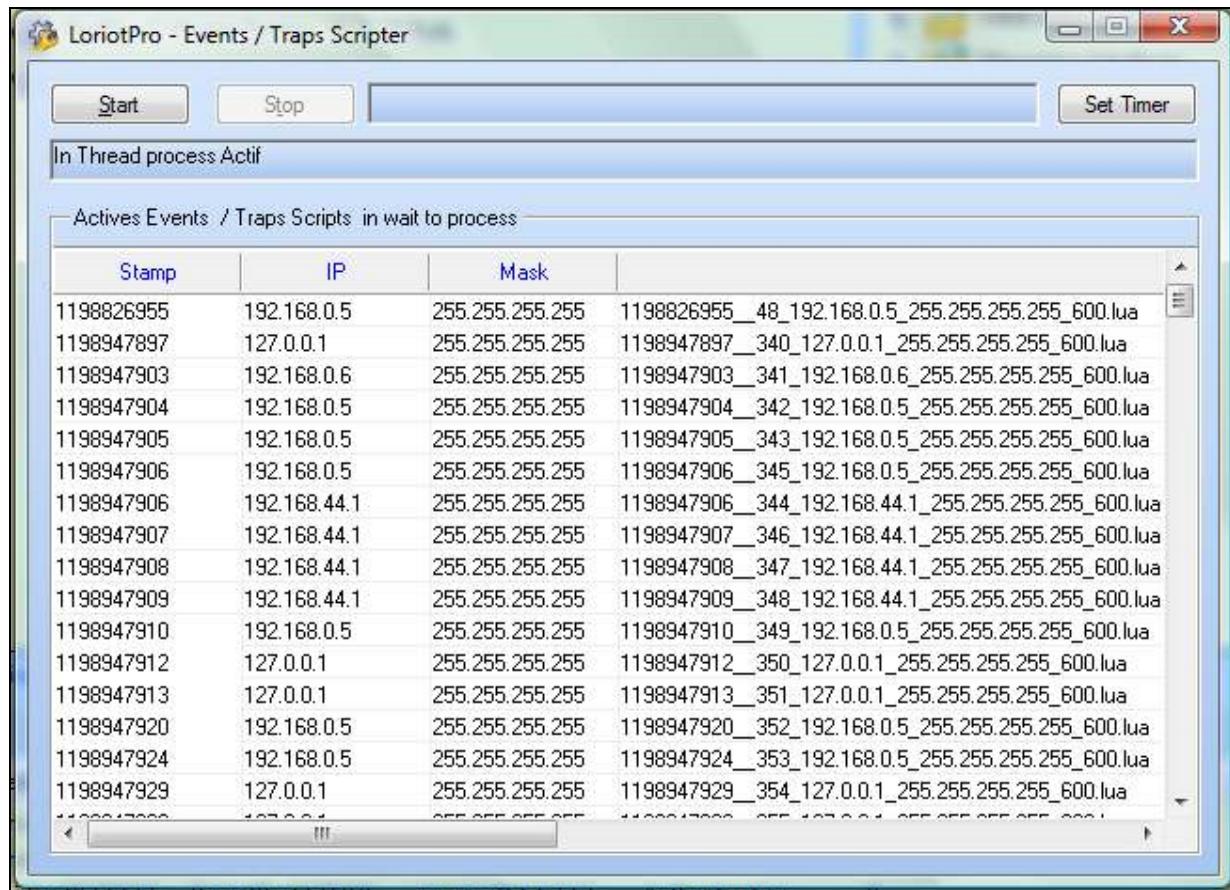
Lance le service. Le service est lancé automatiquement à l'initialisation du logiciel.

### 17.6 Génération de Script pour le module Event / Trap Script Queue manager'

Les scripts utilisés par le module ‘Event / Trap Script Queue manager’ sont des scripts classiques, le module lance simplement le script en lui fournissant un ensemble de variables globales correspondant aux paramètres de l’Event ou du Trap. Le script peut ou non exploiter ces variables lors de son exécution. Pour réaliser cette encapsulation de variable de façon dynamique, le noyau du logiciel génère un fichier structuré unique contenant les variables et le lien vers le script qui les exploite. Ce fichier unique est placé dans le répertoire **bin/EventScripter** .



Le Service ‘Event / Trap Script Queue manager’ scrute à intervalles réguliers ce répertoire et exécute les fichiers de script présents.

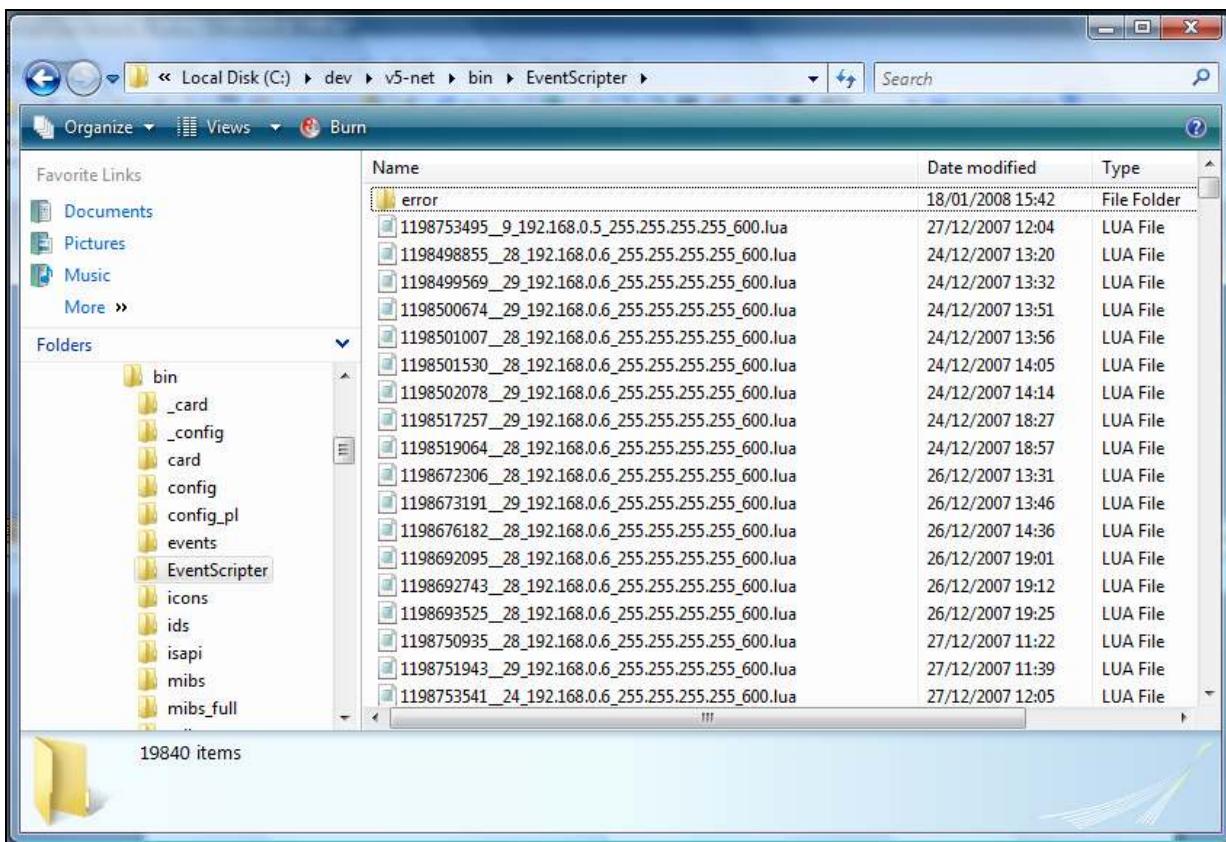


Les scripts lancés par le module ‘Event / Trap Script Queue manager’ sont situés dans l’arborescence

***Bin/config/script/*** du logiciel.

### 17.6.1 Structure des fichiers dynamiques

Ces fichiers sont placés par le noyau du logiciel dans le répertoire ***bin/EventScripter***.



Le répertoire **error** contient les fichiers qui n'ont pas pu être traité par le module 'Event / Trap Script Queue manager'.

Le nom du fichier est structuré pour permettre son analyse par le module 'Event / Trap Script Queue manager'.

Pour un Event :

**TimeStamp\_Numerod-ordre\_IP\_Mask\_EventNumber.lua**

Pour un Trap :

**TimeStamp\_Numerod-ordre\_IP\_Mask.lua**

## 17.6.2 Variables associées aux fichiers d'Event LoriotPro

### 17.6.2.1 Exemple de fichier d'Event généré

```
E_H_IP='192.168.0.6';
E_H_Mask='255.255.255.255';
E_H_Event=600;
E_H_Level=2;
E_H_EventNumber=1762;
E_H_Msg='adv_audit: 192.168.0.6 100(0) audit the storage unit of the host is
down';
```

```
E_E_Description='Audit Module Information';
E_E_NumberActif=1015;
E_E_NumberTotal=1015;
E_F_NumberActif=1015;
E_F_NumberTotal=1015;
E_P_1='param1';
dofile(lp.GetPath() .. '/config/script/events/event01.lua');
```

La première partie contient les variables globales et la dernière ligne le lancement du script sélectionné dans le filtre.

La variable **Ip\_host** est passée au script pour lui fournir l'adresse IP de l'équipement concerné par le filtre.

**Attention** 

Il faut que l'équipement concerné par l'événement se trouve dans la Directory pour que le script puisse s'exécuter dans de bonnes conditions.

### 17.6.3 Variables associées aux fichiers de TRAP SNMP

#### 17.6.3.1 Exemple d'un fichier de Trap SNMP généré

```
T_H_IP='127.0.0.1';
T_H_Version=0;
T_H_Generic=6;
T_H_Specific=22008;
T_H_Community='public';
T_H_Timestamp=1197551083;
T_F_Name='cpqRackEnclosureFanFailed_v1';
T_F_Trap='compaq';
T_F_String='1-ip_info';
T_F_NumberActif=982;
T_F_NumberTotal=982;
T_F_EventNumber=300;
T_F_EventLevel=0;
T_A_Name='1-ip_info';
T_A_NumberActif=982;
T_A_NumberTotal=982;
T_P_Number=9;
T_P_TableBuffer={};
T_P_TableBuffer[0]='145';
T_P_TableBuffer[1]='string';
T_P_TableBuffer[2]='string';
T_P_TableBuffer[3]='string';
T_P_TableBuffer[4]='string';
T_P_TableBuffer[5]='string';
T_P_TableBuffer[6]='string';
T_P_TableBuffer[7]='145';
T_P_TableBuffer[8]='string';
T_P_TableValue={};
```

```
T_P_TableValue[0]='145';
T_P_TableValue[1]='0';
T_P_TableValue[2]='0';
T_P_TableValue[3]='0';
T_P_TableValue[4]='0';
T_P_TableValue[5]='0';
T_P_TableValue[6]='0';
T_P_TableValue[7]='145';
T_P_TableValue[8]='0';
T_P_TableName={ };
T_P_TableName[0]='cpqrackcommonenclosuretrapsequencenum.0';
T_P_TableName[1]='cpqrackcommonenclosurefansparepartnumber.0';
T_P_TableName[2]='cpqrackcommonenclosurefanlocation.0';
T_P_TableName[3]='cpqrackcommonenclosureserialnum.0';
T_P_TableName[4]='cpqrackcommonenclosurename.0';
T_P_TableName[5]='cpqrackuid.0';
T_P_TableName[6]='cpqrackname.0';
T_P_TableName[7]='cpqhotrapflags.0';
T_P_TableName[8]='sysname.0';
dofile(lp.GetPath() .. '/config/script/1-ip_info.lua');
```

La première partie contient les variables globales, et la dernière ligne le lancement du script sélectionné dans le filtre.

La variable **Ip\_host** est passée au script pour lui fournir l'adresse IP de l'équipement concerné par le filtre.

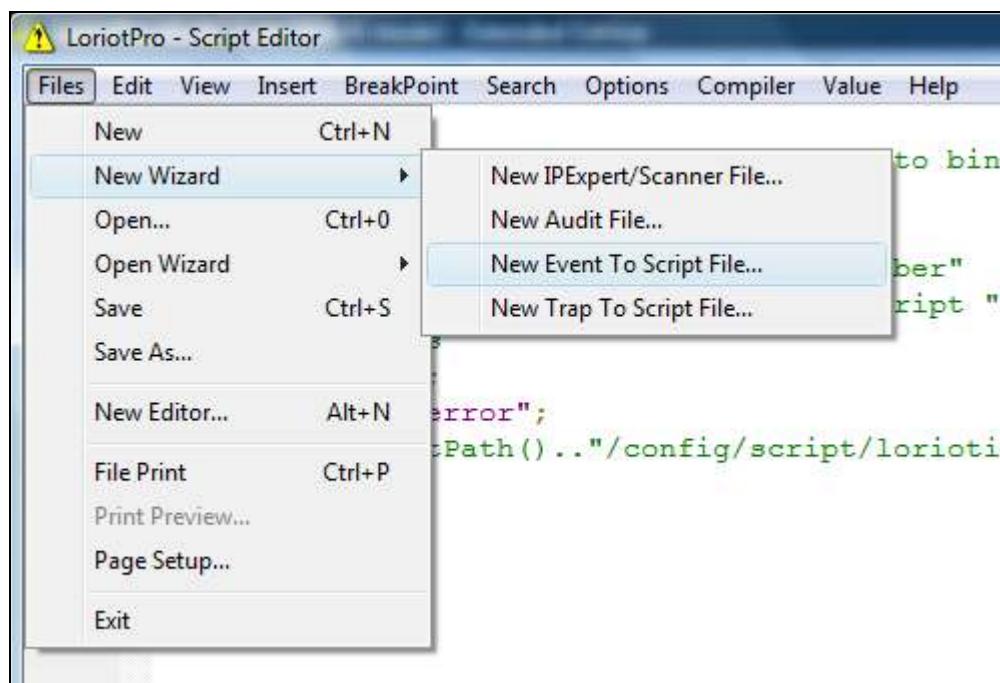
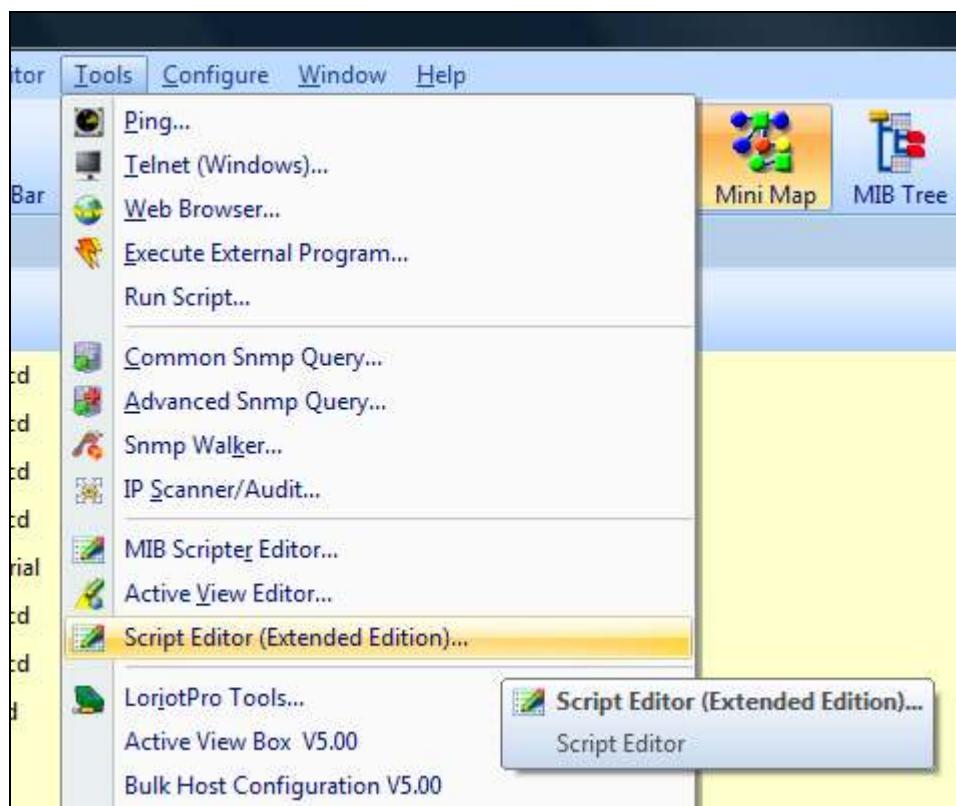
**Attention** 

Il faut que l'équipement concerné par l'événement se trouve dans la Directory pour que le script puisse s'exécuter dans de bonnes conditions.

#### 17.6.4 Wizard de creation d'un fichier de script de filtre d'Event

L'environnement de développement vous propose un Wizard de génération de script pour le module 'Event / Trap Script Queue manager'.

Le lancement de l'éditeur se fait à partir du menu principal du logiciel :



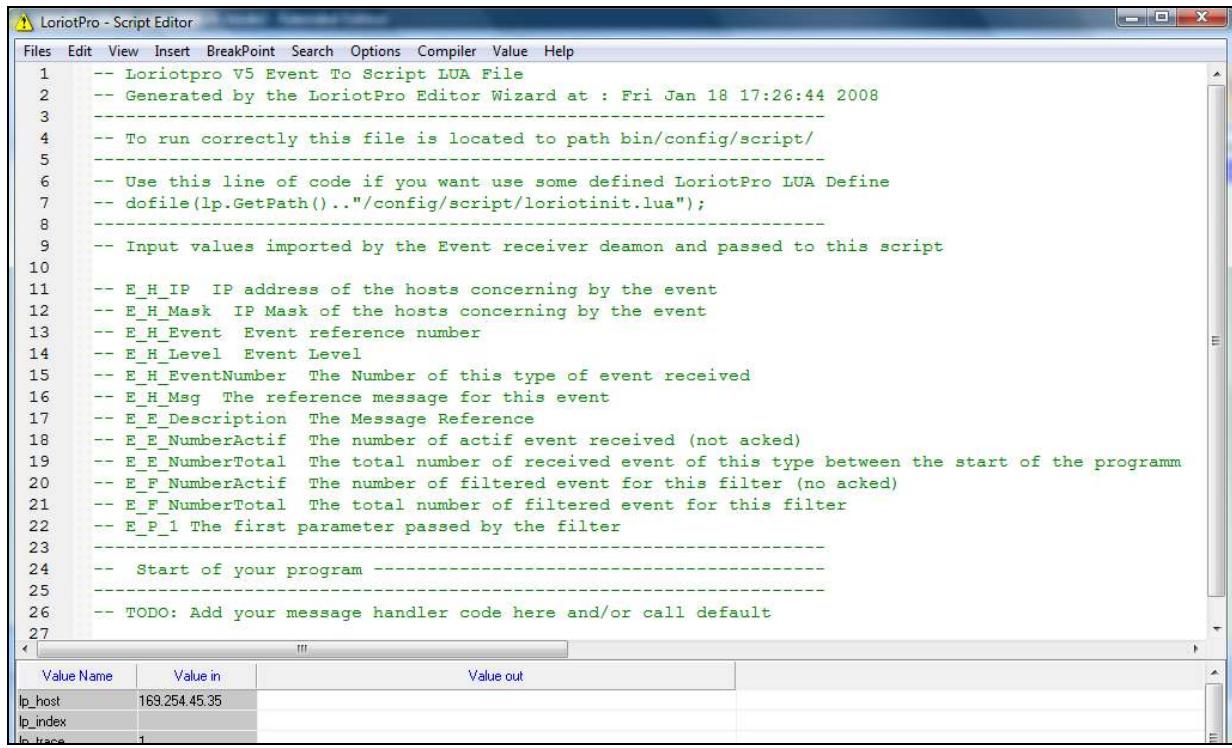
Il faut choisir l'option :

**Files/New Wizard/New IP Expert/New Event To Script File...**

Pour un script d'Event et :

#### 17.6.4.1 Filtre d'Event LoriotPro

Un fichier standard est créé pour vous aider à gérer les variables passées par le noyau du logiciel et placées dans le fichier de traitement.



Le fichier ne contient que des commentaires pour vous rappeler le nom et l'objet des variables passées au script.

```
-- Loriotpro V5 Event To Script LUA File
-- Generated by the LoriotPro Editor Wizard at : Fri Jan 18 17:26:44 2008
-----
-- To run correctly this file is located to path bin/config/script/
-----
-- Use this line of code if you want use some defined LoriotPro LUA Define
-- dofile(lp.GetPath().."/config/script/lriotinit.lua");
-----
-- Input values imported by the Event receiver deamon and passed to this
script

-- E_H_IP IP address of the hosts concerning by the event
-- E_H_Mask IP Mask of the hosts concerning by the event
-- E_H_Event Event reference number
-- E_H_Level Event Level
-- E_H_EventNumber The Number of this type of event received
-- E_H_Msg The reference message for this event
-- E_E_Description The Message Reference
-- E_E_NumberActif The number of actif event received (not acked)
-- E_E_NumberTotal The total number of received event of this type between
the start of the programm
-- E_F_NumberActif The number of filtered event for this filter (no acked)
-- E_F_NumberTotal The total number of filtered event for this filter
-- E_P_1 The first parameter passed by the filter
```

```
-- Start of your program -----
-----
TODO: Add your message handler code here and/or call default
```

#### 17.6.4.2 Variables d'Event

Variables	Commentaires
E_H_IP	IP address of the hosts concerning by the event
E_H_Mask	IP Mask of the hosts concerning by the event
E_H_Event	Event reference number
E_H_Level	Event Level
E_H_EventNumber	The Number of this type of event received
E_H_Msg	The reference message for this event
E_E_Description	The Message Reference
E_E_NumberActif	The number of actif event received (not acked)
E_E_NumberTotal	The total number of received event of this type between the start of the programm
E_F_NumberActif	The number of filtered event for this filter (no acked)
E_F_NumberTotal	The total number of filtered event for this filter
E_P_1	The first parameter passed by the filter

#### 17.6.5 Wizard de création d'un fichier de script de filtre de TRAP SNMP

Il faut choisir l'option :

**Files/New Wizard/New IP Expert/New Trap To Script File...**

Pour un filtre de Trap SNMP.

#### 17.6.5.1 Filtre de Trap SNMP

Le fichier ne contient que des commentaires pour vous rappeler le nom et l'objet des variables passées au script.

```
-- Loriotpro V5 Event To Script LUA File
-- Generated by the LoriotPro Editor Wizard at : Fri Jan 18 17:31:14 2008
-----
-- To run correctly this file is located to path bin/config/script/
-----
-- Use this line of code if you want use some defined LoriotPro LUA Define
-- dofile(lp.GetPath().."/config/script/loriotinit.lua");
-----
-- Input values imported by the Event receiver deamon and passed to this
script

-- T_H_IP  IP address of the agent concerning by the Trap
-- T_H_Version  The version of SNMP used by the Trap
-- T_H_Generic  The SNMP generic value (if SNMP V1 Trap)
```

```
-- T_H_Specific The SNMP specific value (if SNMP V1 Trap)
-- T_H_Community The SNMP community used by the Trap (V1 v2c)
-- T_H_Timestamp The SNMP timestamp used by the Trap (V1)
-- T_T_Name The Trap name
-- T_T_Trap The Trap reference
-- T_T_String The string associed by LoriotPro for this Trap
-- T_T_NumberActif The number of this Trap received (no acked)
-- T_T_NumberTotal The total number of this Trap received
-- T_T_EventNumber The Event Number rerouted for this Trap
-- T_T_EventNumber The Event Number rerouted for this Trap
-- T_T_EventLevel The Event level rerouted for this Trap
-- T_A_Name The filter name for this Trap
-- T_A_NumberActif The filtered trap number (no acked)
-- T_A_NumberTotal The total number for this Trap number
-- T_P_Number The Number of parameter received with this Trap
-- T_P_TableBuffer The Table for collect the parameters
-- T_P_TableBuffer[0] The first parameter received for this trap
-- T_P_TableValue The Table for collect the parameters
-- T_P_TableValue[0] The first parameter received for this trap
-- T_P_TableName The Table for collect the parameters
-- T_P_TableName[0] The first parameter received for this trap
-- T_P_1 The first Parameter passed with then trap filter
-- use this part to simulate an Event to debug your script
---[[[
T_H_IP='';           -- IP address of the agent concerning by the Trap
T_H_Version='';    -- The version of SNMP used by the Trap
T_H_Generic='';    -- The SNMP generic value (if SNMP V1 Trap)
T_H_Specific='';   -- The SNMP specific value (if SNMP V1 Trap)
T_H_Community='';   -- The SNMP community used by the Trap (V1 v2c)
T_H_Timestamp='';   -- The SNMP timestamp used by the Trap (V1)
T_T_Name='';        -- The Trap name
T_T_Trap='';        -- The Trap reference
T_T_String='';      -- The string associed by LoriotPro for this Trap
T_T_NumberActif=''; -- The number of this Trap received (no acked)
T_T_NumberTotal=''; -- The total number of this Trap received
T_T_EventNumber=''; -- The Event Number rerouted for this Trap
T_T_EventNumber=''; -- The Event Number rerouted for this Trap
T_T_EventLevel='';  -- The Event level rerouted for this Trap
T_A_Name='';        -- The filter name for this Trap
T_A_NumberActif=''; -- The filtered trap number (no acked)
T_A_NumberTotal=''; -- The total number for this Trap number
T_P_Number=1;       -- The Number of parameter received with this Trap
T_P_TableBuffer={}; -- The Table for collect the parameters
T_P_TableBuffer[1]=''; -- The first parameter received for this trap
T_P_1='param1';     -- The first Parameter passed with then trap filter
-- use this part to simulate an Event to debug your script
]]--
-----
-- Start of your program -----
-----
-- TODO: Add your message handler code here and/or call default
-- [[ -- sample
lp.Print('Trap IP Add : ',T_H_IP,'\n');
]]--
```

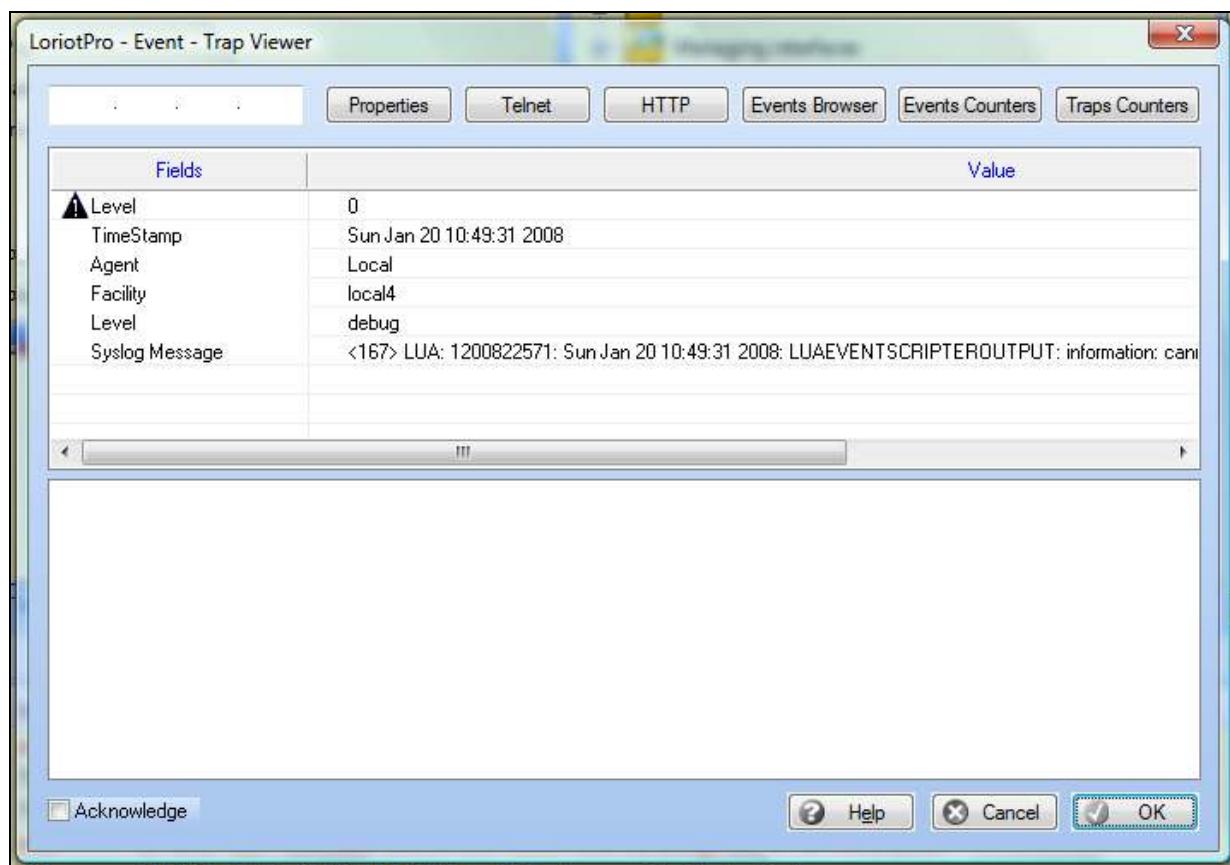
### 17.6.5.2 Variables de Trap SNMP

Variables	Commentaires
T_H_IP	IP address of the agent concerning by the Trap
T_H_Version	The version of SNMP used by the Trap
T_H_Generic	The SNMP generic value (if SNMP V1 Trap)
T_H_Specific	The SNMP specific value (if SNMP V1 Trap)
T_H_Community	The SNMP community used by the Trap (V1 v2c)
T_H_Timestamp	The SNMP timestamp used by the Trap (V1)
T_T_Name	The Trap name
T_T_Trap	The Trap reference
T_T_String	The string associed by LoriotPro for this Trap
T_T_NumberActif	The number of this Trap received (no acked)
T_T_NumberTotal	The total number of this Trap received
T_T_EventNumber	The Event Number rerouted for this Trap
T_T_EventLevel	The Event level rerouted for this Trap
T_A_Name	The filter name for this Trap
T_A_NumberActif	The filtered trap number (no acked)
T_A_NumberTotal	The total number for this Trap number
T_P_Number	The Number of parameter received with this Trap
T_P_TableBuffer	The Table for collect the parameters
T_P_TableBuffer[0]	The first parameter received for this trap
T_P_TableValue	The Table for collect the parameters
T_P_TableValue[0]	The first parameter received for this trap
T_P_TableName	The Table for collect the parameters
T_P_TableName[0]	The first parameter received for this trap
T_P_1	The first Parameter passed with then trap filter

### 17.6.6 Mise au point des scripts

Les scripts étant lancés en tâche de fond par le service ‘Event / Trap Scripter’, il n'est pas toujours aisé de détecter une erreur d'exécution. Pour vous aider dans cette tâche, le module envoie un syslog vous indiquant l'erreur rencontrée.

Syslogs			
t	Facility	Level	Alert
????	????	127.0.0.1	Adv_Audit: 127.0.0.1 100(3) Audit the storage unit of the host is down
????	????	127.0.0.1	WARNING Host [127.0.0.1] AS Low disk C:\Label: Serial Number ecc14f60 : Reached 82% 63263/76317 Mo
local4	debug	<167>	LUA: 1200822274: Sun Jan 20 10:44:34 2008: LUAEVENTSCRIPTEROUTPUT: information: cannot open c:/dev/v5-net/bin/config/script/testeventfilter.lua: No such file or directory
local4	debug	<167>	LUA: 1200822274: Sun Jan 20 10:44:34 2008: LUAEVENTSCRIPTEROUTPUT: information: cannot open c:/dev/v5-net/bin/config/script/testeventfilter.lua: No such file or directory
local4	debug	<167>	LUA: 1200822273: Sun Jan 20 10:44:33 2008: LUAEVENTSCRIPTEROUTPUT: information: cannot open c:/dev/v5-net/bin/config/script/testeventfilter.lua: No such file or directory



```
<167> LUA: 1200822571: Sun Jan 20 10:49:31 2008: LUAEVENTSCRIPTEROUTPUT:  
information: cannot open c:/dev/v5-  
net/bin/config/script/testeventfilter.lua: No such file or Directory
```

La référence utilisée par le module dans le message syslog est :

**LUAEVENTSCRIPTEROUTPUT:**

## 18 Utilisation direct des scripts

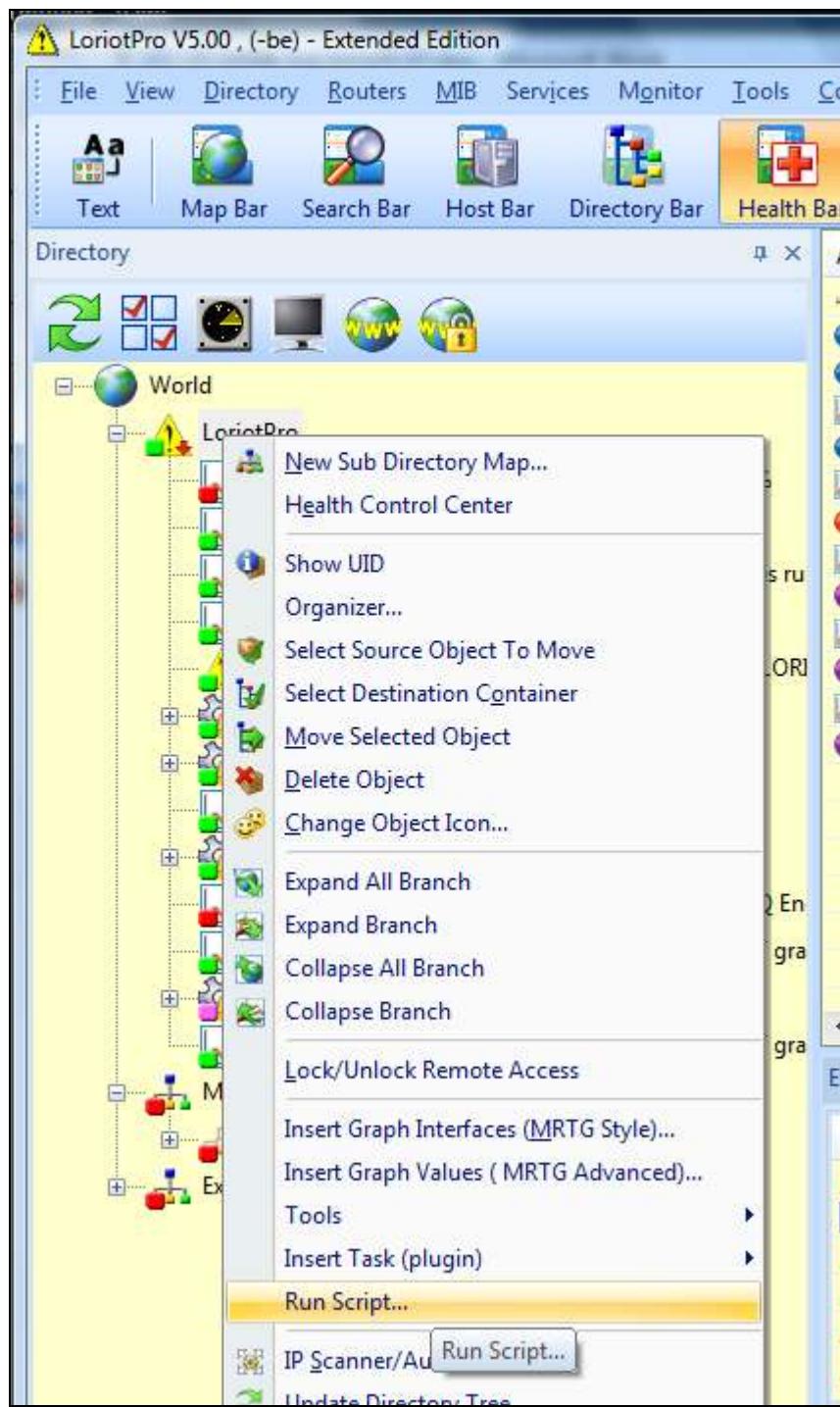
### 18.1 Introduction

LoriotPro vous permet de lancer directement des scripts à la demande en fonction de vos besoins.

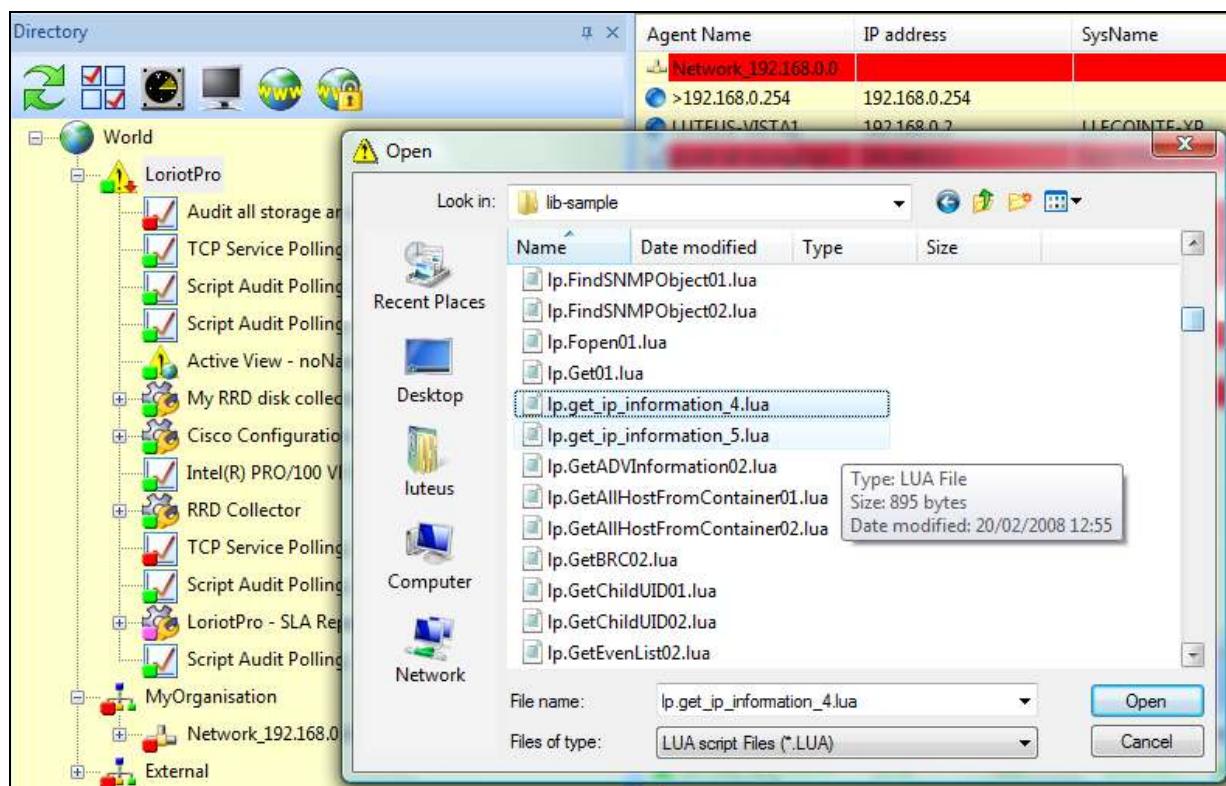
### 18.2 A partir des menus du logiciel

Il est possible de lancer un script à partir des menus du logiciel. Le logiciel passe dans la variable **Ip\_host** l'adresse IP du Host par défaut sélectionnée dans la Directory.

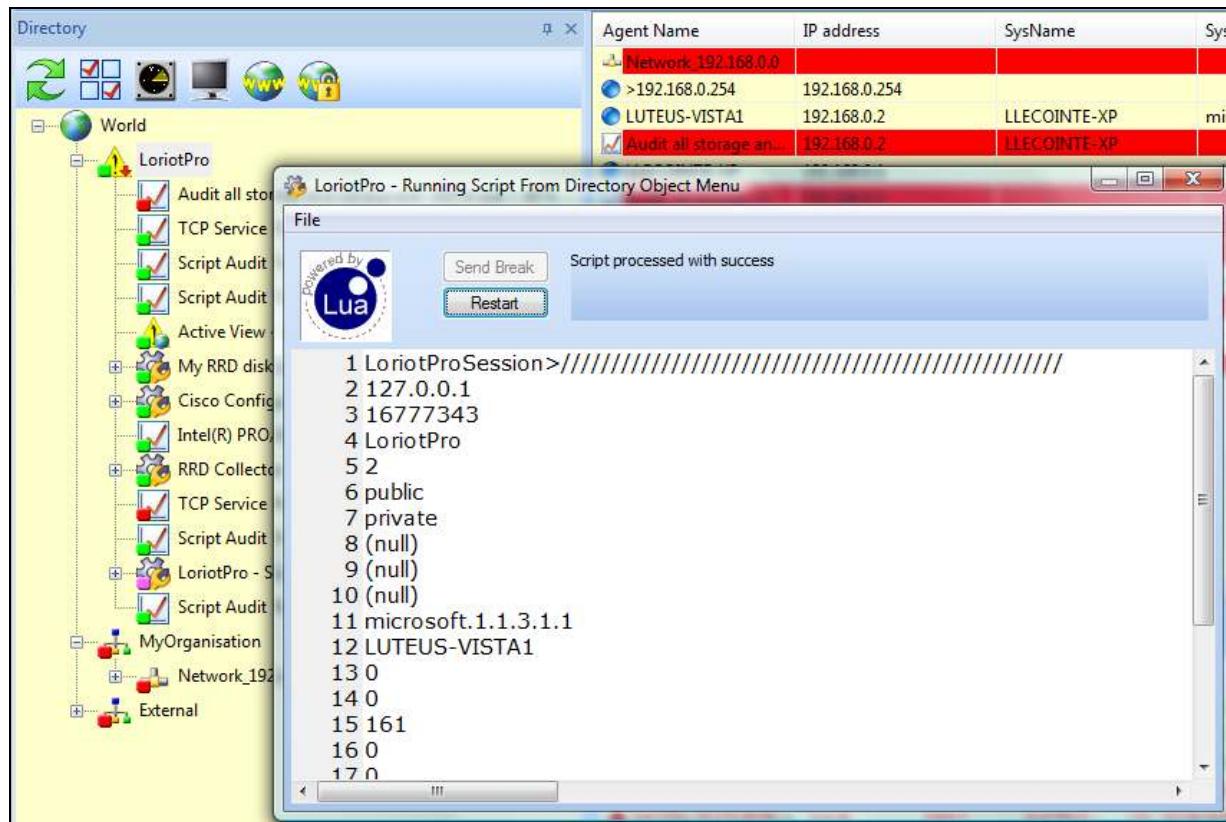
#### 18.2.1.1 Menu contextuel d'un équipement



## LoriotPro V5.00 Extended Edition – Supervision et gestion SNMP par scripts LUA Introduction au System Expert et aux scripts

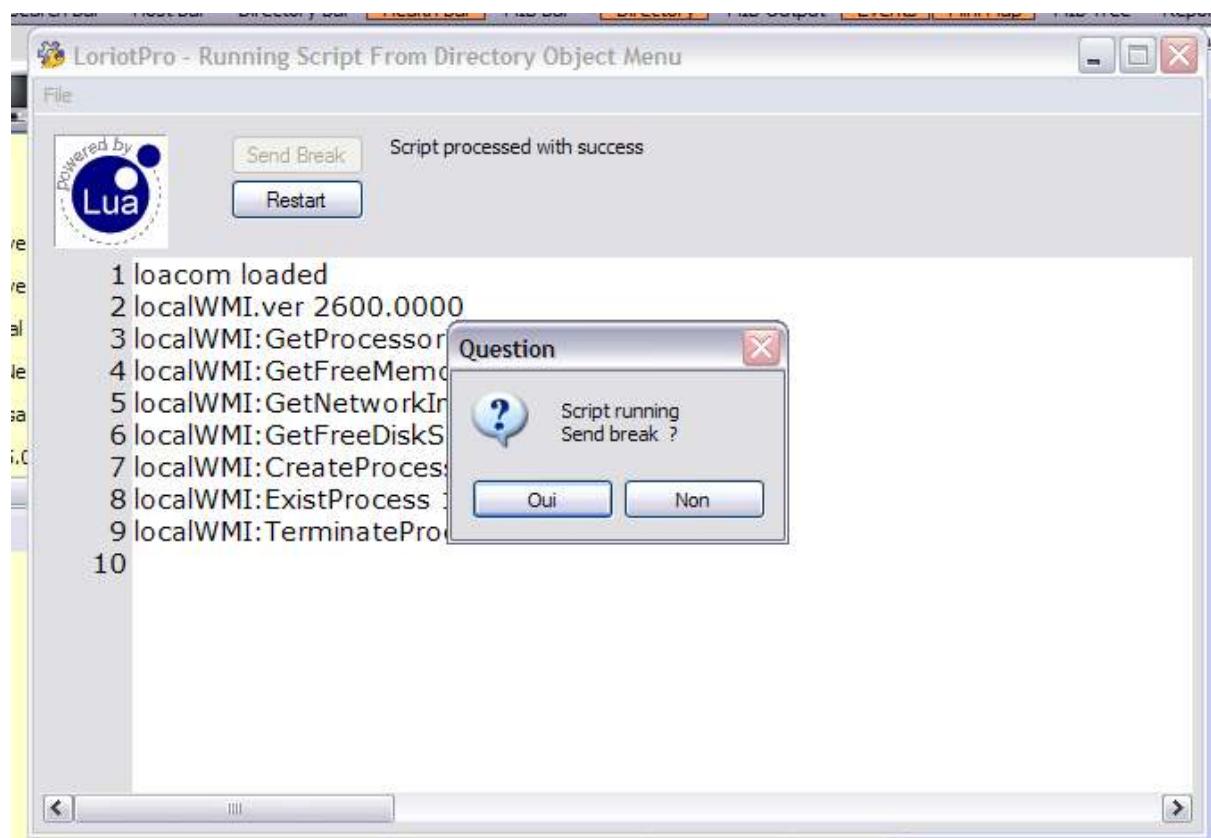


Sélectionnez votre script.



La fenêtre d'exécution des scripts est chargée et le script exécuté directement. Il est possible de relancer le script avec le bouton « Restart » ; un bouton « Send Break »

permet d'envoyer vers le noyau LUA une demande d'arrêt du script. En fonction de l'état de travail du noyau de LoriotPro, cette fonction peut être sans effet et vous devez alors attendre la fin de l'exécution du script.

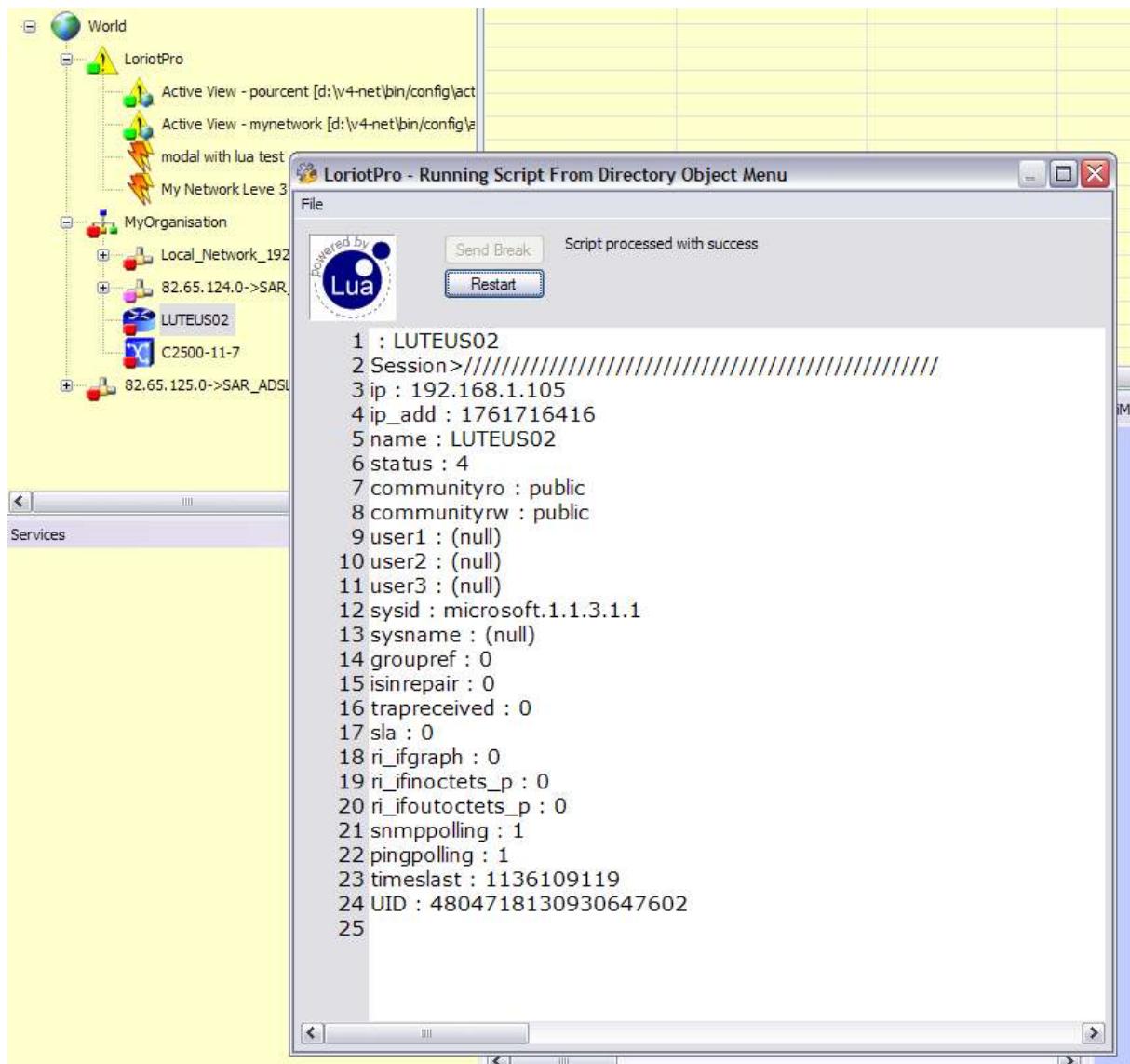


Dans cet exemple, on vérifie que la variable **Ip\_host** correspondant à l'adresse IP de l'équipement sélectionné dans la Directory est bien passée au script :

The screenshot shows the LoriotPro Script Editor interface. The title bar reads "LoriotPro - Script Editor : D:\v4-net\bin\config\script\1-ip\_info.lua". The menu bar includes Files, Edit, View, Insert, Search, Options, Compiler, and Value. The main code area contains the following LUA script:

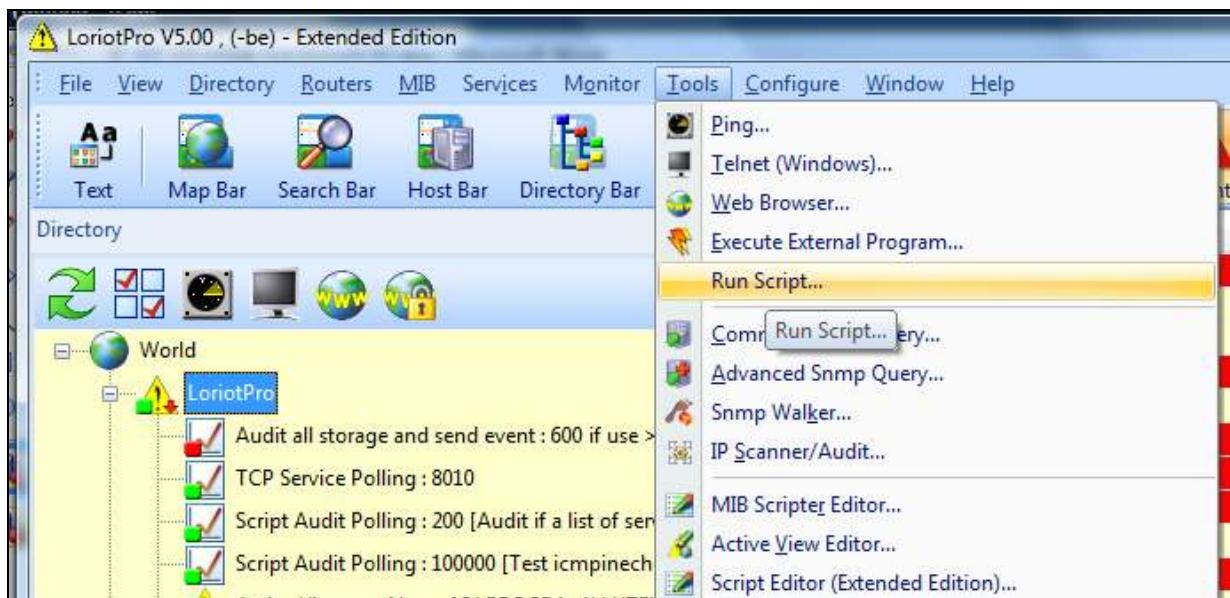
```
11 lp_value=lp.GetIPInformation(ip_host,"aa");
12
13
14
15 -- if lp_value then
16
17     lp_buffer=aa["name"];
18     lp.Print(" : ",aa.name,"\n");
19     lp.Print(Session>//////////,"\\n");
20     lp.Print("ip : ",aa["ip"],"\n");
21     lp.Print("ip_add : ",aa["ip_add"],"\n");
22     lp.Print("name : ",aa["name"],"\n");
23     lp.Print("status : ",aa["status"],"\n");
24     lp.Print("communityro : ",aa["communityro"],"\n");
25     lp.Print("communityrw : ",aa["communityrw"],"\n");
26     lp.Print("user1 : ",aa["user1"],"\n");
27     lp.Print("user2 : ",aa["user2"],"\n");
28     lp.Print("user3 : ",aa["user3"],"\n");
29     lp.Print("sysid : ",aa["sysid"],"\n");
30     lp.Print("sysname : ",aa["sysname"],"\n");
31     lp.Print("groupref : ",aa["groupref"],"\n");
32     lp.Print("isinrepair : ",aa["isinrepair"],"\n");
33     lp.Print("trapreceived : ",aa["trapreceived"],"\n");
34     lp.Print("sla : ",aa["sla"],"\n");
35     lp.Print("ri_ifgraph : ",aa["ri_ifgraph"],"\n");
36     lp.Print("ri_ifinoctets_p : ",aa["ri_ifinoctets_p"],"\n");
37     lp.Print("ri_ifoutoctets_p : ",aa["ri_ifoutoctets_p"],"\n");
38     lp.Print("snmppolling : ",aa["snmppolling"],"\n");
39     lp.Print("pingpolling : ",aa["pingpolling"],"\n");
40     lp.Print("timeslast : ",aa["timeslast"],"\n");
41     lp.Print("UID : ",aa["UID"],"\n");
42 else
43     lp_buffer="#ERROR";
44 end
```

Below the code editor is a table with three columns: "Value Name", "Value in", and "Value out".



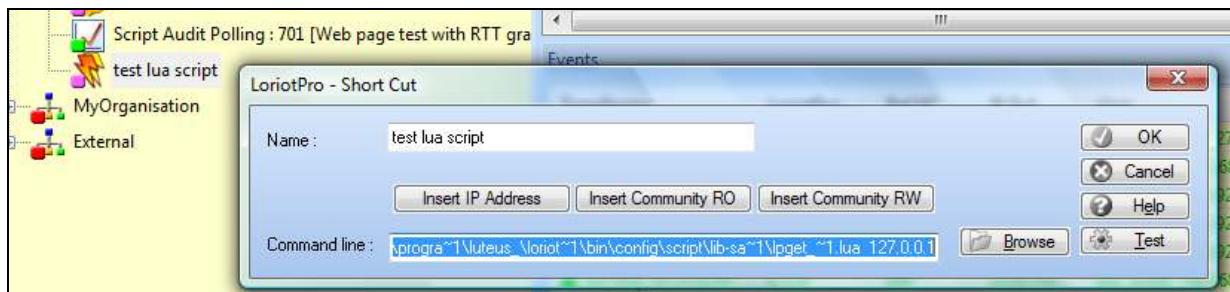
### 18.2.1.2      **Menu principal**

Il est aussi possible de lancer un script à partir du menu principal.

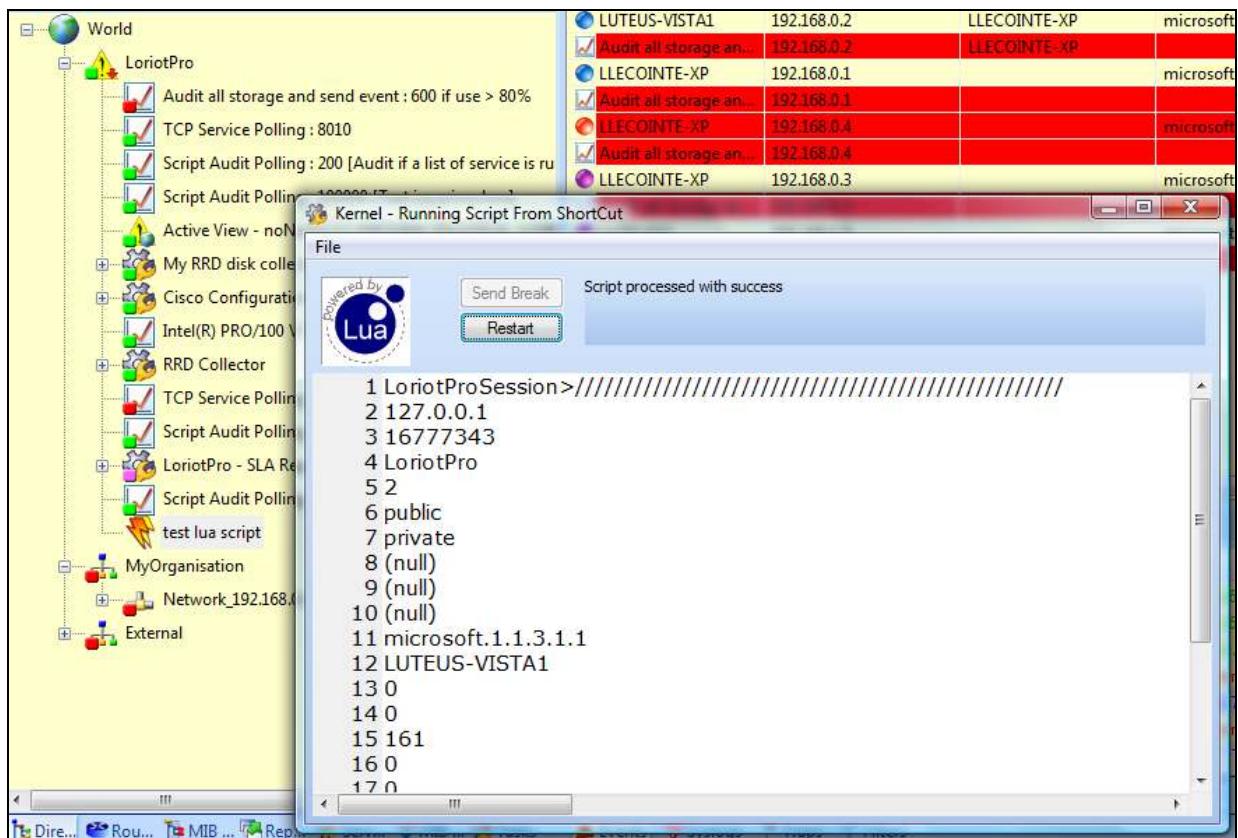


### 18.3 A partir d'un Shortcut d'équipement

On peut définir un “shortcut” raccourci qui permet de lancer un script associé à un équipement de la Directory.



## LoriotPro V5.00 Extended Edition – Supervision et gestion SNMP par scripts LUA Introduction au System Expert et aux scripts



### Attention !

Pour des raisons de portabilité (configuration d'un équipement sur la plateforme de développement vers la plateforme de production si l'arborescence des répertoires est différente) vous pouvez supprimer une partie du nom de fichier (le path) pour travailler avec des emplacements relatifs de fichiers.

C:\PROGRA~1\LUTEUS\LORIOT~2\bin\config\script\1-ODBC~1.LUA 127.0.0.1  
Devient  
config\script\1-ODBC~1.LUA 127.0.0.1

la partie **C:\PROGRA~1\LUTEUS\LORIOT~2\bin\** correspondant au path du logiciel peut être supprimée.

## 19 ANNEXES

### 19.1 Intégration du SLA dans LUA

#### 19.1.1 Introduction

De base, le module de polling (ICMP/SNMP) intégré à LoriotPro permet de sauvegarder des informations concernant les résultats de polling des équipements. Ces résultats sont sauvegardés dans des fichiers texte journaliers, les noms des fichiers sont codés avec un "timestamp" pour une recherche rapide d'une période. Ces fichiers texte sont placés dans des répertoires au nom de l'adresse IP de l'équipement interrogé « Poller ». Ces résultats permettent de calculer un SLA sur une période donnée.

### 19.2 LoriotPro (Ipsla) Librairie

Cette librairie donne accès à des fonctions permettant d'exploiter les données SLA associées aux équipements et modules de la Directory. Consultez la documentation du logiciel pour mieux comprendre comment valider des collectes de SLA d'équipements d'Audit ou de polling.

La librairie **lua\_lp\_sla.dll** permet de réaliser le calcul du SLA pour une période donnée.

Initialisation de la librairie dans un script

```
if (lp.IsDebugMode() == 1) then
lib,init=lp.LoadLibrary(lp.GetPath() .. "/lua_lp_slad.dll", "libinit");
else
lib,init=lp.LoadLibrary(lp.GetPath() .. "/lua_lp_sla.dll", "libinit");
end

if (lib) then
init();
-- votre code
end
```

### 19.3 Tableau de définition de la librairie

Voici un résumé de l'ensemble des fonctions de cette librairie.

Fonctions	Syntaxe simplifié
lpsla.Compute	lpsla.Compute('loriotpro_id', 'sla_rep', Syear, Smonth, Sday, Eyear, Emonth, Eday, STime, ETime, RTT_Threshold, Availability, Performance, 'array');

lpsla.GetLoriotProIDList	lpsla.GetLoriotProIDList('array');
lpsla.GetSLAList	lpsla.GetSLAList('ID', 'array');

**Attention** 

Les fonctions standard de LoriotPro retournent des tableaux indexés numériquement en utilisant l'index 0 en première position de tableau. Classiquement les tableaux indexés numériquement dans LUA débutent leurs indexation avec la valeur 1. Cette particularité provoque quelques erreurs dans l'utilisation de certaines fonctions de gestion des tables LUA.

La fonction table.getn() peut être utilisée de la façon suivante :

```
j=lp.FindFile(lp.GetPath(),"*.dll","a");
if j then
lp.Print('Find ',j,' Files\n');
    for i=0,table.getn(a) do
        lp.Print(a[i],"\n");
    end
end
```

par contre la fonction ipairs() qui permet de browser un tableau indexé numériquement n'est pas utilisable car elle n'affiche pas le premier élément du tableau.

```
for key,value in ipairs(t) do lp.Print("[",key,"] [",value,"]\n") end
```

Les fonctions concernées par cette limitation sont :

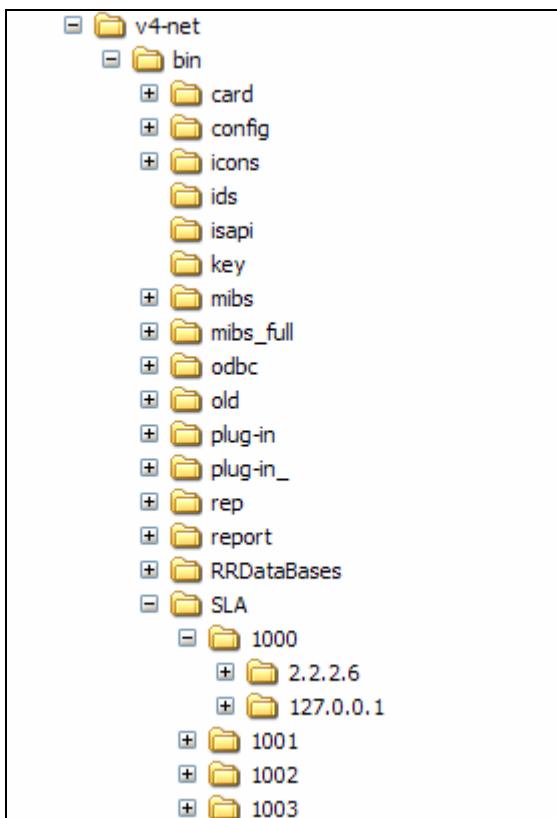
lp.FindFile  
lp.GetAllHostFromContainer  
lp.GetEvenList  
lp.GetEventActionList  
lp.GetTableEntryList  
lp.GetTrapFilterActionList  
lp.GetTrapFilterList  
lp.GetRows  
lp.Parse  
lp.Print

lpav.FindRef

lpsla.GetSLAList  
lpsla.GetLoriotProIDList

## 19.4 Architecture des répertoires de collectes des SLA

Le logiciel met en oeuvre, pour sauvegarder ces résultats, une logique de structure de répertoire utilisant comme racine le répertoire SLA. Les fichiers sauvegardés dans l'arborescence sont des fichiers banalisés avec un timestamp.



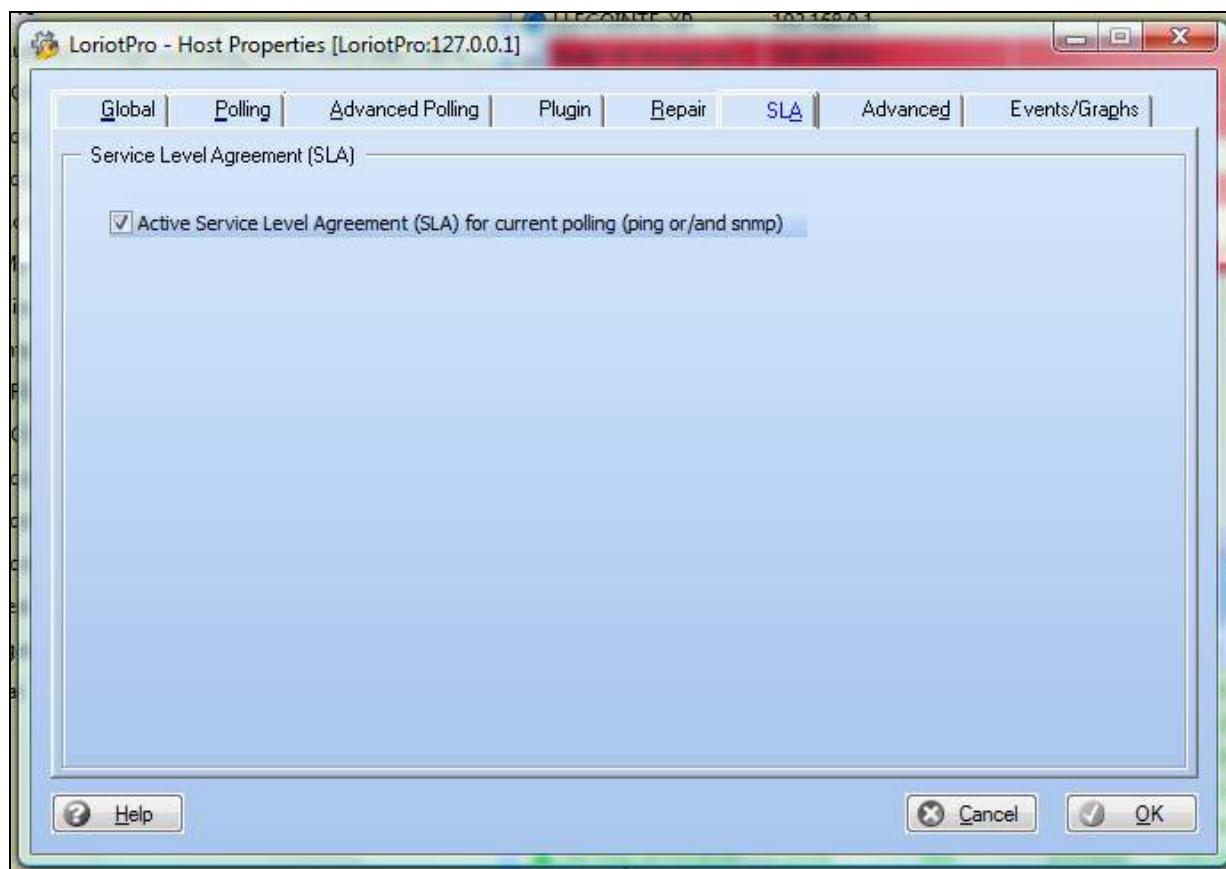
Dans notre capture, *V4-net* est le répertoire d'installation du logiciel. Par défaut le logiciel s'installe dans :

C:\Program Files\LUTEUS\LoriotPro V4\

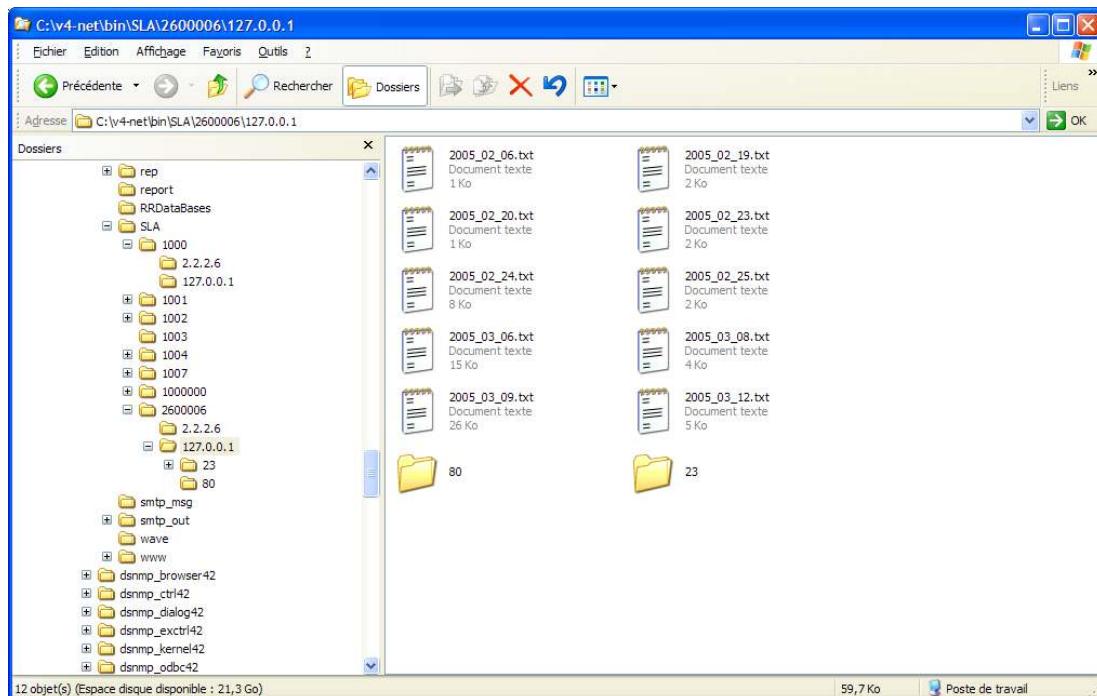
Le répertoire réservé au SLA est dans \bin\SLA soit par défaut :

C:\Program Files\LUTEUS\LoriotPro V4\bin\SLA.

Dans ce répertoire, on trouve un premier niveau correspondant à l'identifiant (ID) de la licence d'utilisation, ceci nous permet d'avoir plusieurs collecteurs. Dans notre exemple, nous avons les identifiants **1000 1001 1002** et **1003** qui correspondent à plusieurs LoriotPro. Après l'identifiant de licence, on trouve un répertoire par adresse IP configurée pour supporter la collecte SLA. En standard, le logiciel ne collecte pas le résultat du polling. Cette option doit être validée dans les propriétés d'un équipement dans le signet SLA.



Les fichiers placés directement dans le répertoire avec l'adresse IP correspondent à la collecte des informations de polling de l'équipement. Les répertoires avec l'adresse IP peuvent contenir d'autres répertoires qui correspondent à des niveaux de SLA plus précis concernant un port UDP TCP ou des URL. La logique de noms de ces répertoires n'est pas encore arrêtée.



Dans tous les cas, les fichiers se trouvant dans les répertoires ont une structure identique pour permettre une analyse simple des valeurs. Le fichier ne contient pas de référentiel de l'équipement mais uniquement des résultats. Le respect de l'architecture des répertoires est donc très important car un fichier mal placé ne fournit plus aucune information valide.

Les modules d'audit et de polling avancé TCP peuvent aussi générer des fichiers qui permettent le calcul d'un SLA.

La logique est la suivante :

Pour TCP :

Le répertoire de collecte répond à la logique suivante :

### **IP-TCP-port-UID**

**IP**      adresse IP de l'équipement concerné  
**TCP**     information statique  
**Port**    le numéro de port TCP pollé  
**UID**     l'UID 64bits de l'item dans la Directory.

Pour un module d'audit :

Le répertoire de collecte répond à la logique suivante :

### **IP-AUDIT-Numero-UID**

**IP**      adresse IP de l'équipement concerné  
**AUDIT**    information statique  
**Numero**   le numéro de référence de l'audit  
**UID**     l'UID 64bits de l'item dans la Directory.

Name	Date modified	Type	Size
212.81.82.126	10/05/2008 15:07	File Folder	
192.168.0.3-AUDIT-100-5162826864779067411	10/05/2008 15:06	File Folder	
192.168.0.4-AUDIT-100-5156029546457202700	10/05/2008 15:06	File Folder	
192.168.0.6-AUDIT-100-5175017588282884118	10/05/2008 15:06	File Folder	
127.0.0.1-AUDIT-700-5194984126737809440	10/05/2008 15:06	File Folder	
127.0.0.1-AUDIT-100-5154912137700705772	10/05/2008 15:06	File Folder	
127.0.0.1-AUDIT-200-5154941979133478384	10/05/2008 15:06	File Folder	
127.0.0.1-AUDIT-701-5195015402689658914	10/05/2008 15:06	File Folder	
127.0.0.1-AUDIT-100000-5155024803282812938	10/05/2008 15:06	File Folder	
192.168.0.1-AUDIT-100-5154914087615858157	10/05/2008 15:06	File Folder	
192.168.0.2-AUDIT-100-5165418838887497748	10/05/2008 15:06	File Folder	
127.0.0.1-TCP-8010-5154921170016929262	10/05/2008 15:06	File Folder	
127.0.0.1-TCP-10113-5194966156594642974	10/05/2008 15:06	File Folder	
212.81.82.126-TCP-23-5194970447266971679	10/05/2008 15:06	File Folder	
127.0.0.1-TCP-0-5195015402689658914	30/04/2008 14:34	File Folder	
127.0.0.1-TCP-0-5194984126737809440	30/04/2008 13:49	File Folder	
127.0.0.1-TCP-23-5154921170016929262	29/04/2008 19:01	File Folder	
127.0.0.1-TCP-0-5155024803282812938	07/03/2008 16:32	File Folder	
127.0.0.1-AUDIT-100-515492270840677861	13/01/2008 12:44	File Folder	
192.168.0.2-AUDIT-100-5144875486324719624	10/01/2008 10:58	File Folder	
127.0.0.1-AUDIT-4-514192051446533254	10/01/2008 10:58	File Folder	
192.168.0.1-AUDIT-100-5153512321959591948	10/01/2008 10:58	File Folder	
192.168.0.6-AUDIT-100-5144879016787836938	10/01/2008 10:58	File Folder	
127.0.0.1-AUDIT-100-5144876006015762441	10/01/2008 10:58	File Folder	

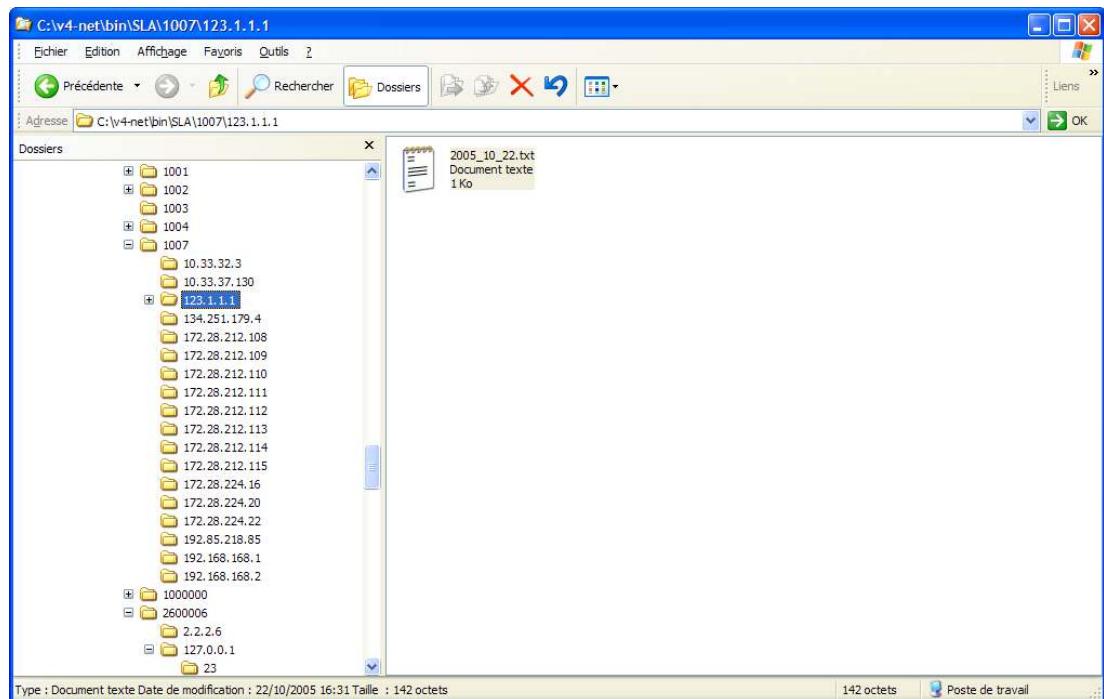
## 19.4.1 Analyse des fichiers

### 19.4.1.1 Codage du nom des fichiers

Le nom des fichiers est codé pour une recherche rapide de période. Il y a un fichier généré par jour (365 par an). Le fichier est au format texte et est lisible par un éditeur standard.

**Année\_Mois\_Jour.txt**

<b>Année</b>	sur 4 digits
<b>Mois</b>	sur 2 digits
<b>Jour</b>	sur 2 digits



Dans cet exemple, nous avons le fichier de collecte du polling (ICMP/SNMP) du 22 octobre 2005 pour l'équipement 123.1.1.1 du « LoriotPro » d'identifiant (ID) 1007.

#### 19.4.1.2 Contenus des fichiers

Un fichier contient une information générique qui est toujours la même quel que soit le type de SLA collecté.

Une ligne correspond à un résultat de collecte avec trois champs

***timestamp;type de polling;(temps de réponse ou information system)***

```
1110629680;1;start
1110629680;2;start
1110629697;2;16
1110629714;2;15
1110629733;2;0
1110629750;2;0
1110629767;2;0
1110629783;2;0
1110629800;2;0
1110629817;2;15
1110629833;2;0
```

#### ***Champ 1***

Le premier champ nous informe sur le timestamp de polling ; pour connaître l'intervalle de polling, il faut analyser le fichier sur plusieurs lignes pour en déduire l'intervalle.

#### ***Champ 2***

Le champ 2 nous informe sur le type de polling :

Numéro	Type de polling
1	ICMP
2	SNMP
3	TCP
4	UDP
5	URL
..	

Dans le cas d'un double polling ICMP / SNMP des 1 et des 2 peuvent apparaître dans le fichier.

### ***Champ 3***

Le champ 3 nous fournit des informations système ou des temps de réponse en millisecondes.

Label	interprétation
<b>start</b>	indique que le SLA a été activé (ou programme démarré 'mini -bk')
<b>stop</b>	indique que le SLA a été interrompu
<b>stop_polling</b>	le polling est globalement arrêté (valable aussi pour un plugin)
<b>start_polling</b>	le polling est globalement démarré (valable aussi pour un plugin)
<b>Stop_repair</b>	<b>(mini -bk) fin de la période de maintenance</b>
<b>Start_repair</b>	<b>(mini -bk) début de la période de maintenance</b>
<b>Stop_loriot</b>	<b>(mini -bk) le logiciel est stoppé</b>
<b>Start_loriot</b>	<b>(mini -bk) le logiciel a démarré</b>
<b>-1</b>	il n'y a pas eu de réponse de la part de l'équipement
<b>35</b>	Une valeur numérique : temps de réponse en millisecondes.

#### **19.4.1.3 Détermination de l'intervalle de polling**

```
1129991441;1;start
1129991441;2;start
1129991450;1;-1
1129991450;2;-1
1129991466;1;-1
1129991466;2;-1
1129991482;1;-1
1129991482;2;-1
1129991498;1;-1
1129991498;2;-1
1129991514;1;-1
```

Dans cet exemple, nous voyons que le SLA a démarré pour cet équipement en ICMP et SNMP le 1129991441, l'intervalle de polling est :

$$1129991466 - 1129991450 = 16 \text{ secondes}$$

Dans ce cas de double polling, il faut prendre le timestamp de la ligne 5 - 4 pour trouver la valeur. En effet, le démarrage du SLA est un processus de collecte qui ne correspond pas à la logique d'intervalles de polling. A chaque polling, il y a une collecte mais le SLA peut être activé entre deux collectes.

Dans le cas d'un simple polling ICMP ou SNMP, il faut analyser le fichier.

Dans cet exemple, l'équipement ne répond pas et dans ce cas, il y a un double polling ICMP (1) et SNMP (2).

Si l'équipement répond aux requêtes SNMP, il n'y aura pas de réponse de type ICMP.

```
1110103926;1;start
1110103926;2;start
1110103943;2;0
1110103960;2;15
1110103979;2;32
1110103996;2;0
1110104013;2;0
1110104029;2;0
1110104046;2;15
1110104062;2;0
1110104079;2;0
```

Si des lignes avec des ICMP(1) apparaissent alors que l'équipement est en double polling, cela veut dire que cet équipement ne répondait plus aux requêtes SNMP(2) et que le poller a basculé pour cet équipement en mode ICMP.

Si l'administrateur stoppe le SLA ou le polling, l'information system **stop** est sauvegardée.

```
1129991866;2;-1
1129991882;1;-1
1129991882;2;-1
1129991895;1;stop
1129991895;2;stop
1129991920;1;start
1129991920;2;start
```

#### 19.4.1.4 Trou de collecte

Un trou dans l'intervalle de polling indique que LoriotPro s'est arrêté anormalement, « crash ».

```
1110103926;1;start
1110103926;2;start
1110103943;2;0
```

```
1110103960;2;15
1110103979;2;32
1110103996;2;0
1110104013;2;0
1110104029;2;0
1110104046;2;15
1110104062;2;0
1110104079;2;0
```

Dans cet exemple, le SLA a démarré à 1110103926

L'intervalle de polling est  $1110103960 - 1110103943 = 17$  secondes, on remarque que l'intervalle varie en fonction de la charge du noyau du logiciel. Il faut donc collecter les informations système (interval de polling) de l'équipement pour connaître les vraies valeurs d'intervalles de temps.

Dans le cas du polling d'un équipement, cela est simple, on peut utiliser dans un module en C++ les fonctions api suivantes :

```
Csnmp m_Csnmp;
snmp_session *ss=NULL;
snmp_session_ext *sse=NULL;
time_t send_at=0;
time_t last_receive_at=0;
time_t polling_interval=0;
clock_t last_icmp_rcv=0;
clock_t last_icmp_send=0;
clock_t last_snmp_rcv=0;
clock_t last_snmp_send=0;

ss=m_Csnmp.FoundIPSession(m_Csnmp.atoi("127.0.0.1"), 0, GLOBALSESSION);
if (ss)
{
    send_at=ss->timesfirst;
    last_receive_at=ss->timeslast;
    polling_interval=ss->timeout;
    sse=(snmp_session_ext *)ss->fonct;

    if (sse)
    {
        last_icmp_rcv = sse->last_icmp_polling_rcv
        last_icmp_send = sse->last_icmp_polling_sent
        last_snmp_rcv = sse->last_snmp_polling_rcv
        last_snmp_send = sse->last_snmp_polling_sent
    }
}
```

ou ss->timesfirst correspond au timestamp du dernier envoi  
ou ss->timeslast correspond au timestamp du dernier reçut  
ou ss->timeout correspond a l'intervalle de polling configuré  
...

Dans le cas d'un polling (TCP UDP ...) généré par un plugin, il faut analyser le fichier (ou utiliser le fichier *description.txt* à définir).

#### 19.4.1.5      Arrêt du logiciel

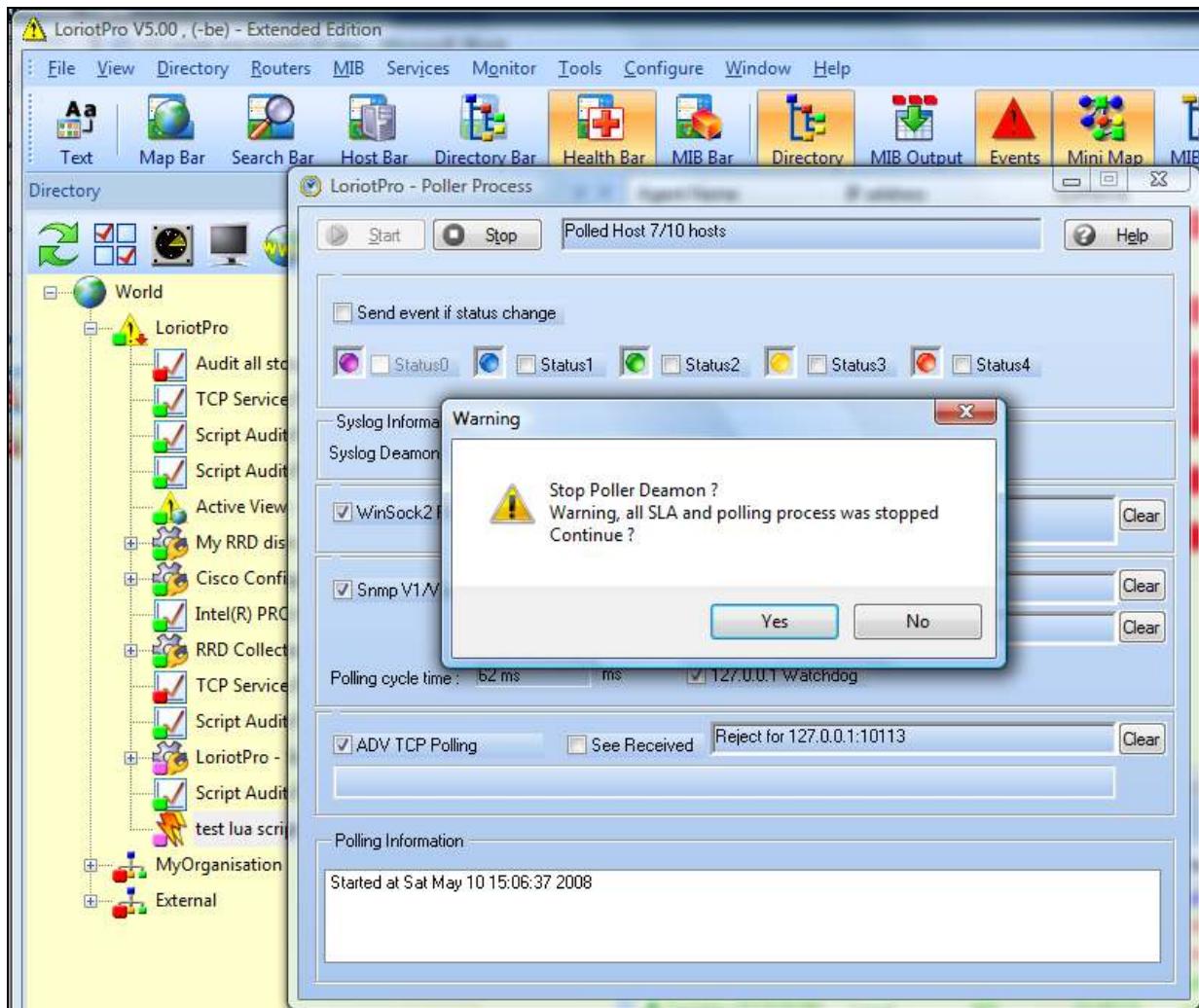
Dans le cas d'un arrêt inopiné du logiciel, il y aura un vide dans le fichier.

```
1110103926;1;start
1110103926;2;start
1110103943;2;0
1110103960;2;15
1110103979;2;32
1110103996;2;0
1110104013;2;0
1110104029;2;0
1110104046;2;15
1110104330;2;0
1110104348;2;0
1110104365;2;0
1110104381;2;0
1110104397;2;0
1110104415;2;15
1110104432;2;0
1110105908;1;start
1110105908;2;start
1110105925;2;16
```

Dans cet exemple, il y a un vide entre 1110105908 (**start**) et 1110104432, on en déduit que LoriotPro a été arrêté.

#### 19.4.1.6      Arrêt global du polling des équipements

Il est possible de stopper manuellement le polling des équipements.

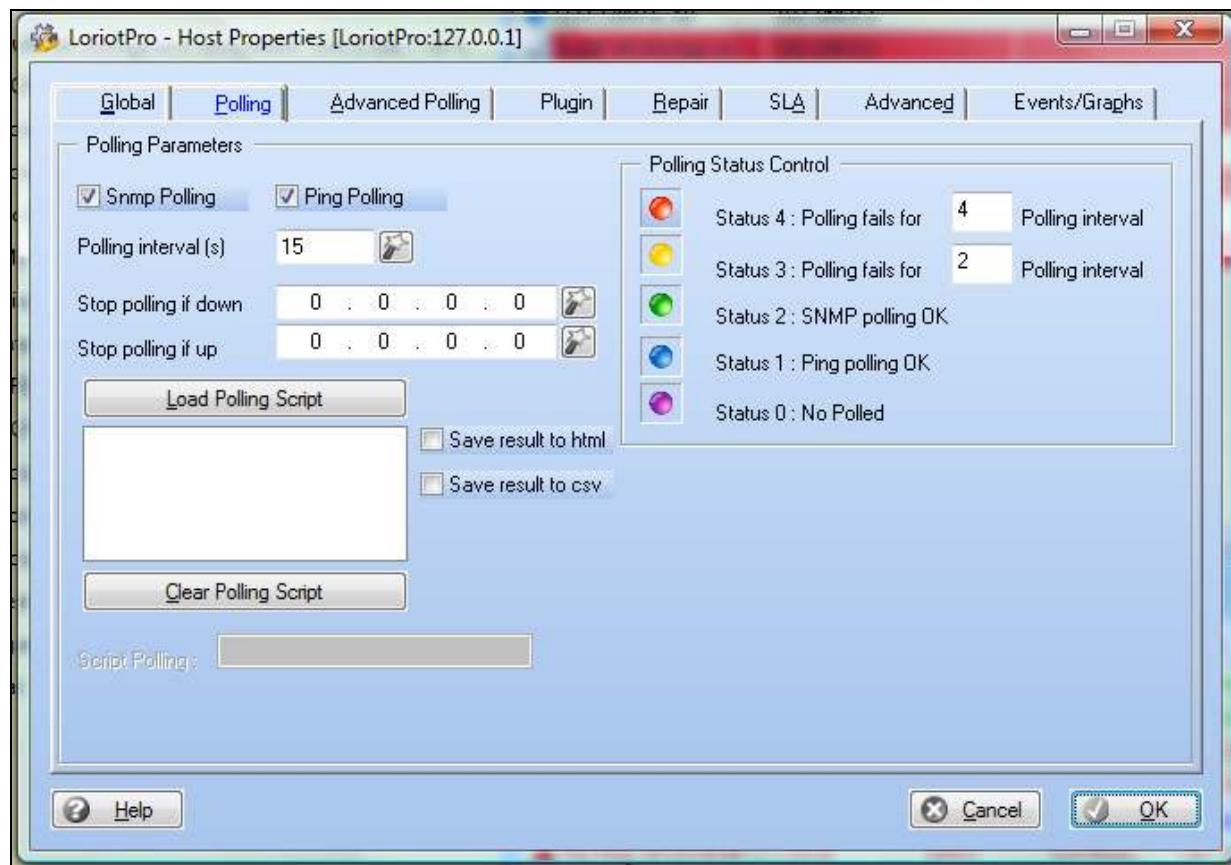


Dans ce cas, il y a une information dans le fichier.

```
1129993499;2;-1
1129993507;1;stop_polling
1129993507;2;stop_polling
1129993511;1;start_polling
1129993511;2;start_polling
1129993515;1;-1
```

#### 19.4.1.7 Arrêt d'un type de polling sur deux

Il est possible de reconfigurer le type de polling d'un équipement.



Un des deux types de polling est changé.

```
1129993675;1;-1
1129993675;2;-1
1129993677;2;stop
1129993691;1;-1
1129993707;1;-1
1129993723;1;-1
```

Ici le polling snmp est arrêté et le SLA aussi.

#### 19.4.2 SLA Conclusion

Les fichiers de SLA générés à partir du polling des équipements sont complexes car nous pouvons avoir un double polling conditionnel (ICMP et/ou SNMP). En effet, pour le moteur de polling, si le protocole SNMP répond, le protocole ICMP n'est pas utilisé, même si les deux types de polling sont validés dans les propriétés de l'équipement. Dans le cas d'un équipement, avec uniquement le protocole ICMP ou SNMP de validé, l'analyse du fichier est beaucoup plus simple. On peut, si on le désire, analyser uniquement les lignes avec le SNMP (2) prioritaire et considérer que si l'équipement passe en mode de polling ICMP (ping) (1) alors il y a rupture du SLA.

```
1130404775;1;start
1130404775;2;start
1130404782;2;203
```

```
1130404795;1;stop_polling ← le polling ICMP globale est stoppé
1130404795;2;stop_polling ← le polling SNMP globale est stoppé
1130404796;1;start_polling ← le polling ICMP globale est lancé
1130404796;2;start_polling ← le polling SNMP globale est lancé
1130404803;2;0 ← pas fiable (mais réponse de l'équipement)
1130404824;2;0
1130404845;2;0
1130404866;2;0
1130404907;1;start ← le SLA sur le polling ICMP est lancé
1130404907;2;start ← le SLA sur le polling SNMP est lancé
1130404920;2;250 ← pas fiable
1130404941;2;0
1130404962;2;0
1130405362;1;start_loriot
1130405362;2;start_loriot
1130405383;2;781
1130405404;1;stop_repair
1130405404;2;stop_repair
1130405460;1;start_repair
1130405460;2;start_repair
1130405460;1;782 ← pas fiable
1130405460;2;0
1130405481;2;15
1130405502;2;0
1130405524;2;-1 ← pas de réponse
1130405545;2;0
1130405576;2;0
1130405731;1;start_loriot ← le logiciel est lancé (ICMP)
1130405731;2;start_loriot ← le logiciel est lancé (SNMP)
1130405752;2;16 ← pas fiable
1130405763;1;stop ← le polling ICMP est arrêté
1130405773;2;0
1130405781;1;start ← le polling ICMP est starté
1130405794;2;16 ← pas fiable
1130405808;1;stop ← le polling ICMP est arrêté
1130405815;2;0
1130405821;2;stop_repair ← le polling est arrêté (snmp) car il y a une période de maintenance
1130405880;2;start_repair le polling est starté (snmp) après une période de maintenance
1130405880;2;0 ← pas fiable
1130405901;2;0
1130405925;2;16
← il y a eu un arrêt non standard
1130406241;2;start_loriot ← le logiciel est lancé (snmp)
1130406271;2;954 ← pas fiable
1130406291;1;stop_loriot ← le logiciel est arrêté (ICMP)
1130406291;2;stop_loriot ← le logiciel est arrêté (SNMP)
```

**Remarque :**

Derrière un ***start\_xxx*** la valeur fournie en millisecondes n'est pas fiable  
car tout dépend de la durée de l'arrêt.

## 20 Librairies externe LUA

Avec la fonction « lp.LoadLibrary », il est possible de charger des librairies écrites par des tiers. Si vous cherchez des librairies LUA sur internet vous en trouverez pour tous les usages. Il est aussi possible d'écrire ses propres librairies. Le projet « Lua\_Wizard\_LIB » fourni avec le package est là pour vous aider dans ce projet.

### Attention

L'usage des librairies externe n'est pas supporté par LUTEUS.

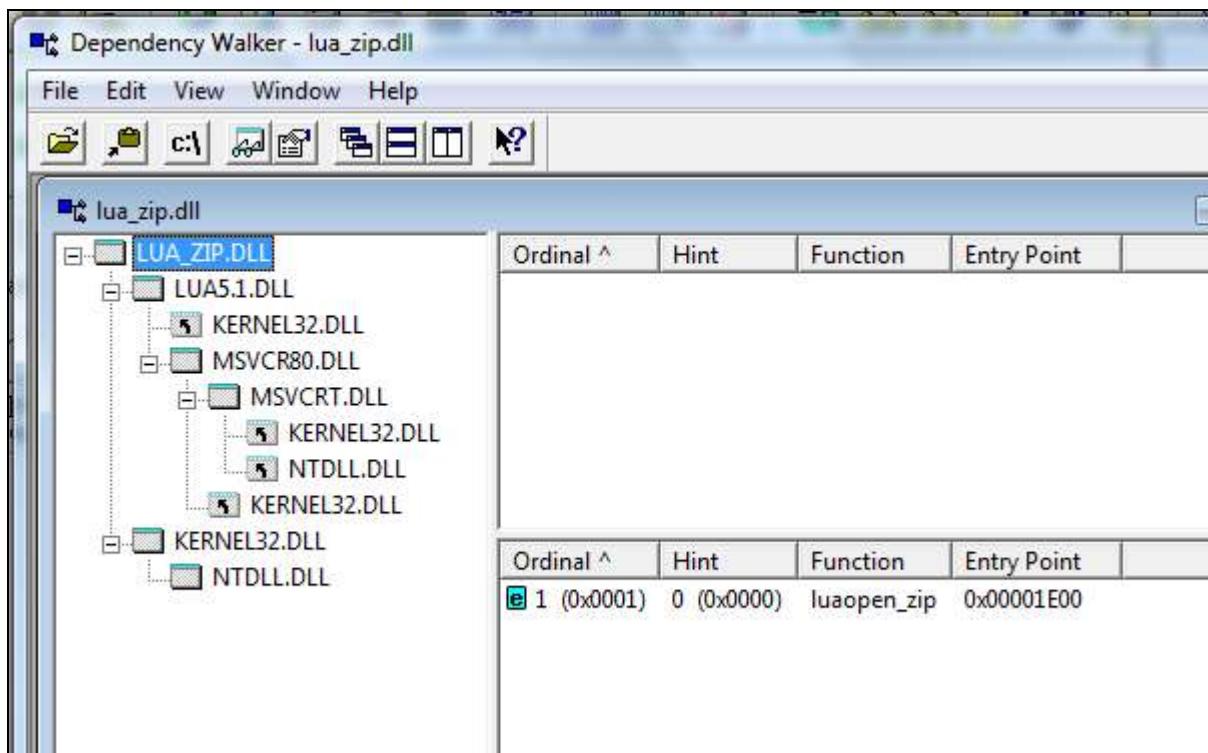
Pour être compatible, les librairies utilisées doivent être compilées en mode dll avec la version 5.1.x de LUA. La dll LUA appelée par la librairie doit être lua5.1.dll, il vous faut aussi connaître le nom de la fonction d'initialisation de la librairie. Vous pouvez utiliser l'utilitaire Dependency de Microsoft pour tester le chaînage des DLL.

Le site [http://luaforge.net/softwaremap/trove\\_list.php](http://luaforge.net/softwaremap/trove_list.php) contient quelques librairies utilisables à tester.

### 20.1 Exemple LUA\_ZIP

Voici par exemple comment intégrer le package luazip-1.2.3-win32-lua51.zip que vous pouvez trouver sur le site du projet Kepler <http://www.keplerproject.org/>.

1. Téléchargez le fichier luazip-1.2.3-win32-lua51.zip.
2. Le fichier luazip-1.2.3-win32-lua51.zip contient une DLL zip.dll ; nous renommons zip.dll en lua\_zip.dll pour ne pas effacer la dll zip.dll déjà présente dans le répertoire bin du logiciel.
3. Copiez la dll dans le répertoire bin.
4. Utilisez l'utilitaire Dependency pour vérifier qu'il ne vous manque rien.



Vous pouvez noter le nom de la fonction d'initialisation : luaopen\_zip

Les autres packages contiennent la documentation pour permettre l'utilisation de la librairie.

Extrait de la documentation de lua\_zip.

#### LuaZip License

LuaZip is free software: it can be used for both academic and commercial purposes at absolutely no cost. There are no royalties or GNU-like "copyleft" restrictions. LuaZip qualifies as Open Source software. Its licenses are compatible with GPL. LuaZip is not in the public domain and the Kepler Project keep its copyright. The legal details are below.

The spirit of the license is that you are free to use LuaZip for any purpose at no cost without having to ask us. The only requirement is that if you do use LuaZip, then you should give us credit by including the appropriate copyright notice somewhere in your product or its documentation.

The LuaZip library was designed and implemented by Danilo Tuler. Part of the code is a derived work from the Lua standard I/O library, copyrighted by Tecgraf, PUC-Rio.

Copyright © 2003-2007 The Kepler Project.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish,

distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

## **LuaZip**

Reading files inside zip files

LuaZip is a lightweight [Lua](#) 5.0 extension library that can be used to read files stored inside zip files. It uses [zziplib](#) to do all the hard work.

The API exposed to Lua is very simple and very similar to the usual file handling functions provided by the [IO](#) [Lua standard library](#). In fact, the API is so similar that parts of this manual are extractions from the Lua manual, copyrighted by Tecgraf, PUC-Rio.

LuaZip source is distributed as a pair of C source and header files. The distribution provides a `Makefile` prepared to compile the library and install it. The file `config` should be edited to suit the needs of the aimed platform.

### Installation

LuaZip follows the [package model](#) for Lua 5.1, therefore it should be "installed". Refer to [Compat-5.1 configuration](#) section about how to install the compiled binary properly.

### Reference

#### **zip.open (filename)**

This function opens a zip file and returns a new zip file handle. In case of error it returns nil and an error message. Unlike `io.open`, there is no `mode` parameter, as the only supported mode is "read".

#### **zip.openfile (filename [, extensions])**

This function opens a file and returns a file handle. In case of errors it returns nil and an error message. Unlike `io.open`, there is no `mode` parameter, as the only supported mode is "read".

This function implements a virtual file system based on optionally compressed files. Instead of simply looking for a file at a given path, this function goes recursively up through all path separators ("/") looking for zip file there. If it finds a zip file, this function uses the remaining path to open the asked file.

The optional parameter `extensions` allows the use of file extensions other than .zip during the lookup. It can be a string corresponding to the extension or an indexed table with the lookup sequence extensions.

#### **zfile:close ()**

This function closes a zfile opened by `zip.open`

#### **zfile:files ()**

Returns an iterator function that, each time it is called, returns a new table containing the following information:

- `filename`: the full path of a file

- `compressed_size`: the compressed size of the file in bytes
- `uncompressed_size`: the uncompressed size of the file in bytes

**`zfile:open (filename)`**

This function opens a file that is stored inside the zip file opened by `zip.open`.  
The filename may contain the full path of the file contained inside the zip. The directory separator must be '/'.

Unlike `f:open`, there is no `mode` parameter, as the only supported mode is "read".

**`file:read (format1, ...)`**

Reads the file `file`, according to the given formats, which specify what to read.  
For each format, the function returns a string with the characters read, or nil if it cannot read data with the specified format. When called without formats, it uses a default format that reads the entire next line (see below).

The available formats are:

- "`*a`": reads the whole file, starting at the current position. On end of file, it returns the empty string.
- "`*l`": reads the next line (skipping the end of line), return nil on end of file. This is the default format.
- `number`: reads a string with up to that number of characters, returning nil on end of file.

Unlike the standard read, the format "`*n`" is not supported.

**`file:seek ([whence] [, offset])`**

Sets and gets the file position, measured from the beginning of the file, to the position given by `offset` plus a base specified by the string `whence`, as follows:

- `set`: base is position 0 (beginning of the file);
- `cur`: base is current position;
- `end`: base is end of file;

In case of success, function `seek` returns the final file position, measured in bytes from the beginning of the file. If this function fails, it returns nil, plus a string error. The default value for `whence` is "`cur`", and for `offset` is 0. Therefore, the call `file:seek()` returns the current file position, without changing it; the call `file:seek("set")` sets the position to the beginning of the file (and returns 0); and the call `file:seek("end")` sets the position to the end of the file, and returns its size.

**`file:close ()`**

This function closes a file opened by `zfile:open`.

**`file:lines ()`**

Returns an iterator function that, each time it is called, returns a new line from the file. Therefore, the construction

```
for line in file:lines() do ... end
```

will iterate over all lines of the file.

## Example

Suppose we have the following file hierarchy:

```
/a
  /b
    c.zip
/a2
  b2.ext2
/a3.ext3
/luazip.zip
```

- `c.zip` contains the file '`d.txt`'

## LoriotPro V5.00 Extended Edition – Supervision et gestion SNMP par scripts LUA Introduction au System Expert et aux scripts

- b2.ext2 is a zip file containing the file 'c2/d2.txt'
- a3.ext3 is a zip file containing the file 'b3/c3/d3.txt'
- luazip.zip contains the files 'luazip.h', 'luazip.c', 'Makefile', 'README'

Below is a small sample code displaying the basic use of the library.

```
local zfile, err = zip.open('luazip.zip')

-- print the filenames of the files inside the zip
for file in zfile:files() do
    print(file.filename)
end

-- open README and print it
local f1, err = zfile:open('README')
local s1 = f1:read("*a")
print(s1)

f1:close()
zfile:close()

-- open d.txt inside c.zip
local d, err = zip.openfile('a/b/c/d.txt')
assert(d, err)
d:close()

-- open d2.txt inside b2.ext2
local d2, err = zip.openfile('a2/b2/c2/d2.txt', "ext2")
assert(d2, err)
d2:close()

-- open d3.txt inside a3.ext3
local d3, err = zip.openfile('a3/b3/c3/d3.txt', {"ext2", "ext3"})
assert(d3, err)
d3:close()
```

Pour utiliser cette librairie dans LoriotPro, utilisez la fonction **Ip.LoadLibrary** et non pas require.

Exemple avec LoriotPro:

```
lib,init=lp.LoadLibrary(lp.GetPath().."/lua_zip.dll","luaopen_zip");
if (lib) then
init();
zfile,err=zip.open (lp.GetPath().."/luazip-1.2.3-win32-lua51.zip");
    if zfile then
        for file in zfile:files() do
            lp.Print(file.filename,"\n")
        end
        zfile:close();
    end
end
lp_buffer ="ok";
```

The screenshot shows the LoriotPro Script Editor interface. The main window displays a Lua script with syntax highlighting. The script code is as follows:

```
5 -- lp_oid SNMP OID for this script "ifnumber"
6 -- lp_host default ip address for this script "127.0.0.1"
7 -- Output Values
8 lp_value = 0;
9 lp_buffer ="error";
10
11 lib,init=lp.LoadLibrary(lp.GetPath().."/lua_zip.dll","luaopen_zip");
12 if (lib) then
13     init();
14     zfile,err=zip.open (lp.GetPath().."/luazip-1.2.3-win32-lua51.zip");
15     if zfile then
16         for file in zfile:files() do
17             lp.Print(file.filename, "\n")
18         end
19         zfile:close();
20     end
21 end
22 lp_buffer ="ok";
23
```

Below the script editor is a table with two columns: "Value Name" and "Value in".

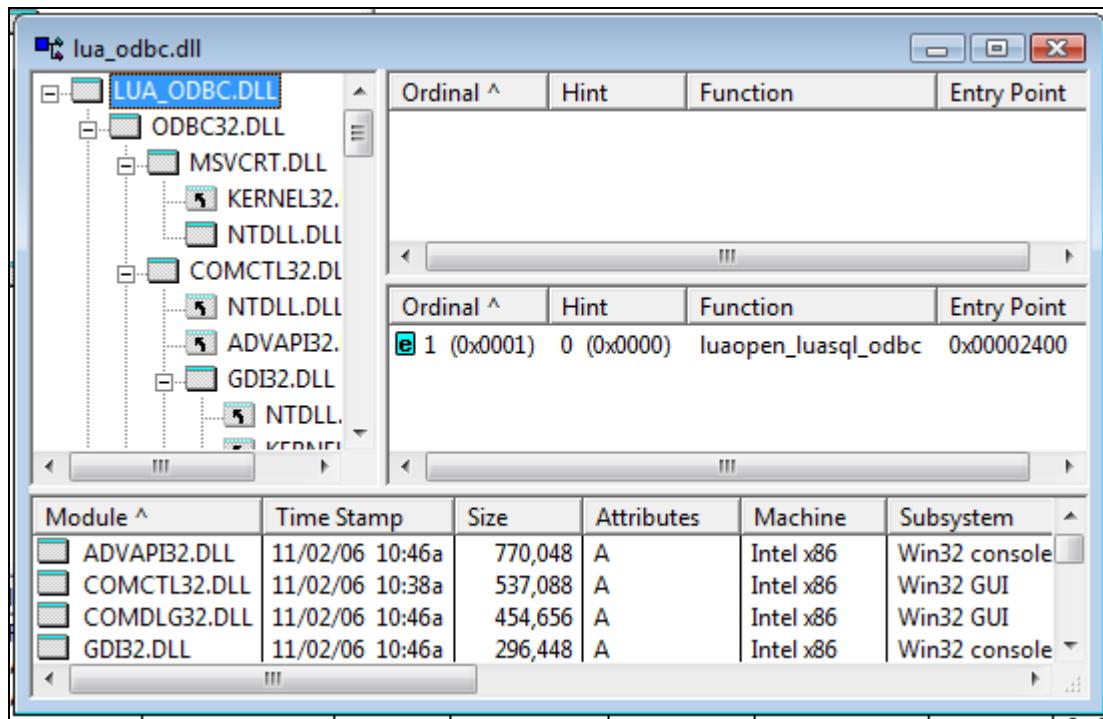
A secondary window titled "LoriotPro - Trace Windows" is open, showing a list of files under the "File" tab:

File
1 zip.dll
2

## 20.2 Exemple LUA\_ODBC

La librairie **lua\_odbc** (<http://www.keplerproject.org/luasql/>) a été testée avec succès pour accéder aux données de la base de données de LoriotPro.

Le fichier **luasql-2.1.1-odbc-win32-lua51.zip** contient la dll **odbc.dll** ; par mesure de sécurité renommez la dll en **lua\_odbc.dll** avant de la copier dans le répertoire **bin** du logiciel.



Exemple d'utilisation:

```
function ODBC_DisplayTable(conn,sql_request)

cur=conn:execute(sql_request);
if cur == nil then
lp.Print("SQL CCMDB_DisplayTable Cur Fail\n");
return 0
end;

row = cur:fetch ({}, "n"); -- the rows will be indexed by field names
if row==nil then
lp.Print("SQL CCMDB_DisplayTable fetch Fail\n");
cur:close();
return 0;
end
j=table.getn(row);

lp.Print("\n");

        if row then
            while row do
                line="";
                k=table.getn(row);
                lp.Print("[",k);
                for i=1,j do
                    line=line..row[i].."|\t";

lp.Print(">");
lp.Print(".");
lp.Print(row[i]);
lp.Print(" | \t");
                end
                lp.Print("]\n");
                --lp.Print(line," \n");
                row = cur:fetch (row,"n");-- reusing the table of results
            end
        end
    end
end
```

```
        end
    cur:close();
return 1;
end

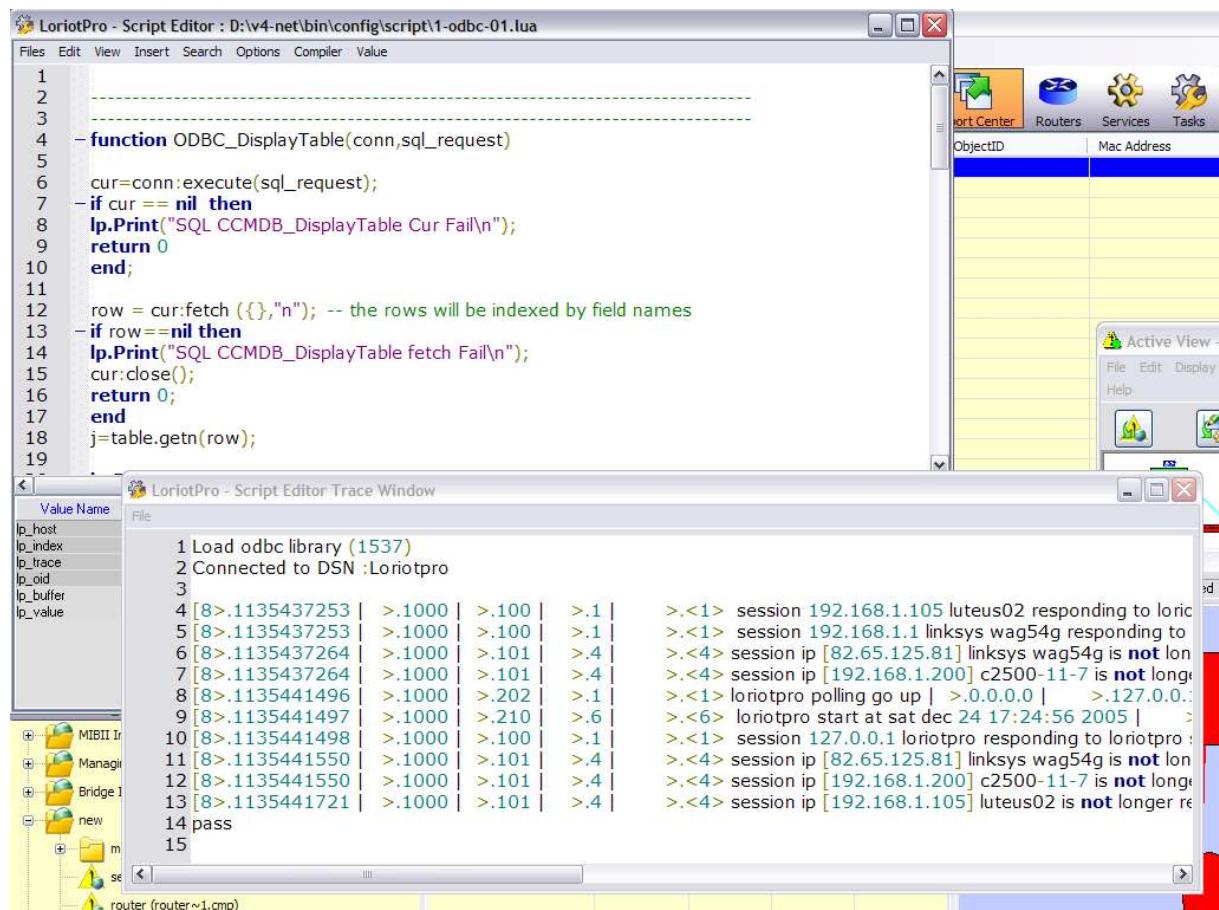
-----
-- main programme
-----
lp_buffer="ok";
lp_value=1;
if lp.IsDebugMode()==1 then
lib,init=lp.LoadLibrary(lp.GetPath().."/lua_odbcd.dll","luaopen_luasqlodbc");
else
lib,init=lp.LoadLibrary(lp.GetPath().."/lua_odbcd.dll","luaopen_luasqlodbc");
end

if lib==nil then error("odbc lib not loaded \n") end;

lp.Print("Load odbc library ("..lp.GetVersion()..")\n");
init();
env=luasql.odbc();

if env then
conn=env:connect("Loriotpro","root","");
if conn~=nil then
    lp.Print("Connected to DSN :","Loriotpro","\n");
    --ODBC_DisplayTable(conn,"Select * FROM loriot_events WHERE
l_reference=101 LIMIT 100;");
    ODBC_DisplayTable(conn,"Select * FROM loriot_events LIMIT 10;");
    conn:close();
else
    lp.Print("Error to connect DSN : Loriotpro\n");
end
env:close();
end

cur:close();
conn:close();
env:close();
lp.Print("pass\n");
```



## 20.3 Exemple LUA\_COM

L'utilisation de la librairie luacom <http://www.tecgraf.puc-rio.br/~rcerq/luacom/> est rendue possible uniquement en mode menu direct (ouverture de l'interface microsoft COM). Cette librairie est extrêmement puissante et permet de réaliser une interface entre LUA et COM. Malheureusement, elle est ancienne ; elle était disponible avec la version 5.0 de LUA uniquement. Vous pouvez cependant trouver le code source sur le site et procéder à une recompilation du code.

### Attention

L'usage de cette librairie ainsi que de toutes les librairies externes n'est pas supporté par LUTEUS.

### Description

LuaCOM is an add-on library to the [Lua language](#) that allows Lua programs to use and implement objects that follow Microsoft's *Component Object Model* (COM) specification and use the *ActiveX technology* for property access and method calls. LuaCOM is implemented as a C++ library and runs either stand-alone or embedded in

another application. LuaCOM is provided with its source code and the makefiles required to build it.

LuaCOM has been designed and implemented by [Vinicius Almendra](#) and [Renato Cerqueira](#), and is maintained by [Fabio Mascarenhas](#), at [TeCGraf](#), the Computer Graphics Technology Group of [PUC-Rio](#) (the Pontifical Catholic University of Rio de Janeiro in Brazil). TeCGraf is a laboratory of [the Department of Computer Science](#).

LuaCOM is available under the same [terms and conditions](#) as the Lua language, that is, it can be used at no cost for both academic and commercial purposes. Send your comments, bug reports and anything else to luacom % tecgraf.puc-rio.br

Copyright © 1998-2005 TeCGraf, PUC-Rio. All rights reserved.

### 20.3.1 Exemple d'utilisation de WMI

```
local lib,init= lp.LoadLibrary(lp.GetPath().."/luacom.dll","luaopen_luacom")

if lib then  init();
lp.Print("luacom loaded\n");
require("luacom")
--[[ 
Example Lua Object for using Windows Management and Instrumentation via LuaCom
This object is meant to be used as a class to create other objects.
Contributed by Michael Cumming
--]]

require("luacom")

cWMI = {
    New = function (self)
        o = {}
        setmetatable (o,self)
        self.__index = self
        return o
    end,
    Connect = function (self,computer,user,password)
        computer = computer or "."
        if not user then
            self.oWMIService = luacom.GetObject
        ("winmgmts:{impersonationLevel=Impersonate}!\\\" ..computer.. "\\root\\cimv2")
        else
            self.oWMIService = luacom.GetObject ("winmgmts:\\\" ..computer..
"\\root\\cimv2",user,password)
        end
        if not self.oWMIService then
            return nil,"Failed to connect to computer "..computer
        end
        --refresher
        self.oRefresher = luacom.CreateObject ("WbemScripting.SWbemRefresher")
        self.oRefresher.AutoReconnect = 1
    -- processor
```

## LoriotPro V5.00 Extended Edition – Supervision et gestion SNMP par scripts LUA

### Introduction au System Expert et aux scripts

```
self.refobjProcessor =
self.oRefresher:AddEnum(self.oWMIService,"Win32_PerfFormattedData_PerfOS_Processor") .ObjectSet

-- memory
self.refobjMemory =
self.oRefresher:AddEnum(self.oWMIService,"Win32_PerfFormattedData_PerfOS_Memory") .ObjectSet

--drive
self.refobjDisk =
self.oRefresher:AddEnum(self.oWMIService,"Win32_PerfFormattedData_PerfDisk_LogicalDisk") .ObjectSet

--network
self.refobjNetwork =
self.oRefresher:AddEnum(self.oWMIService,"Win32_PerfFormattedData_Tcpip_NetworkInterface") .ObjectSet
    self.oRefresher:Refresh ()

local cWMISettings = self.oWMIService:ExecQuery ("Select * from Win32_WMISetting")

for index,item in luacomE.pairs (cWMISettings) do
    self.ver = item:BuildVersion ()
end

return self.ver
end,

GetProcessorPercentTime = function (self)
    self.oRefresher:Refresh ()
    for index,item in luacomE.pairs (self.refobjProcessor) do
        if item:Name () == "_Total" then
            return item:PercentProcessorTime ()
        end
    end
    return 0
end,

GetFreeMemory = function (self)
    local x
    self.oRefresher:Refresh ()
    for index,item in luacomE.pairs (self.refobjMemory) do
        x = item:AvailableMBBytes ()
    end
    return x or 0
end,

GetFreeDiskSpace = function (self,drive)
    local x,y
    self.oRefresher:Refresh ()
    for index,item in luacomE.pairs (self.refobjDisk) do
        if item:Name () == drive then
            x = item:FreeMegaBytes ()
            y = item:PercentFreeSpace ()
            return x,y
        end
    end
    return 0,0
end,

GetNetworkInterfaceAll = function (self)
    local item,BpsSent,BpsRec,BpsTot,bpsBandwidth
    BpsSent,BpsRec,BpsTot,bpsBandwidth = 0,0,0,0
    for index,item in luacomE.pairs (self.refobjNetwork) do
        BpsRec = BpsRec + item:BytesReceivedPerSec()
        BpsSent = BpsSent + item:BytesSentPerSec()
```

## LoriotPro V5.00 Extended Edition – Supervision et gestion SNMP par scripts LUA

### Introduction au System Expert et aux scripts

```
BpsTot = BpsTot + item:BytesTotalPerSec()
bpsBandwidth = bpsBandwidth + item:CurrentBandwidth()
end
return BpsSent,BpsRec,BpsTot,bpsBandwidth
end,

CreateProcess = function (self,Process)
    local objProcess = self.oWMIService:Get("Win32_Process")
    return objProcess>Create (Process,nil,nil,nil)
end,

---[[ returns the following error codes
0 Successful completion
2 Access denied
3 Insufficient privilege
8 Unknown failure
9 Path not found
21 Invalid parameter ]]

TerminateProcess = function (self,Process)
    local colProcesses = self.oWMIService:ExecQuery("select * from Win32_Process
where Name=\\\"..Process..\\\"",nil,48)
    local i
    for index,item in luacomE.pairs (colProcesses) do
        i = (i or 0) + 1
        item:Terminate ()
    end
    return i
end,

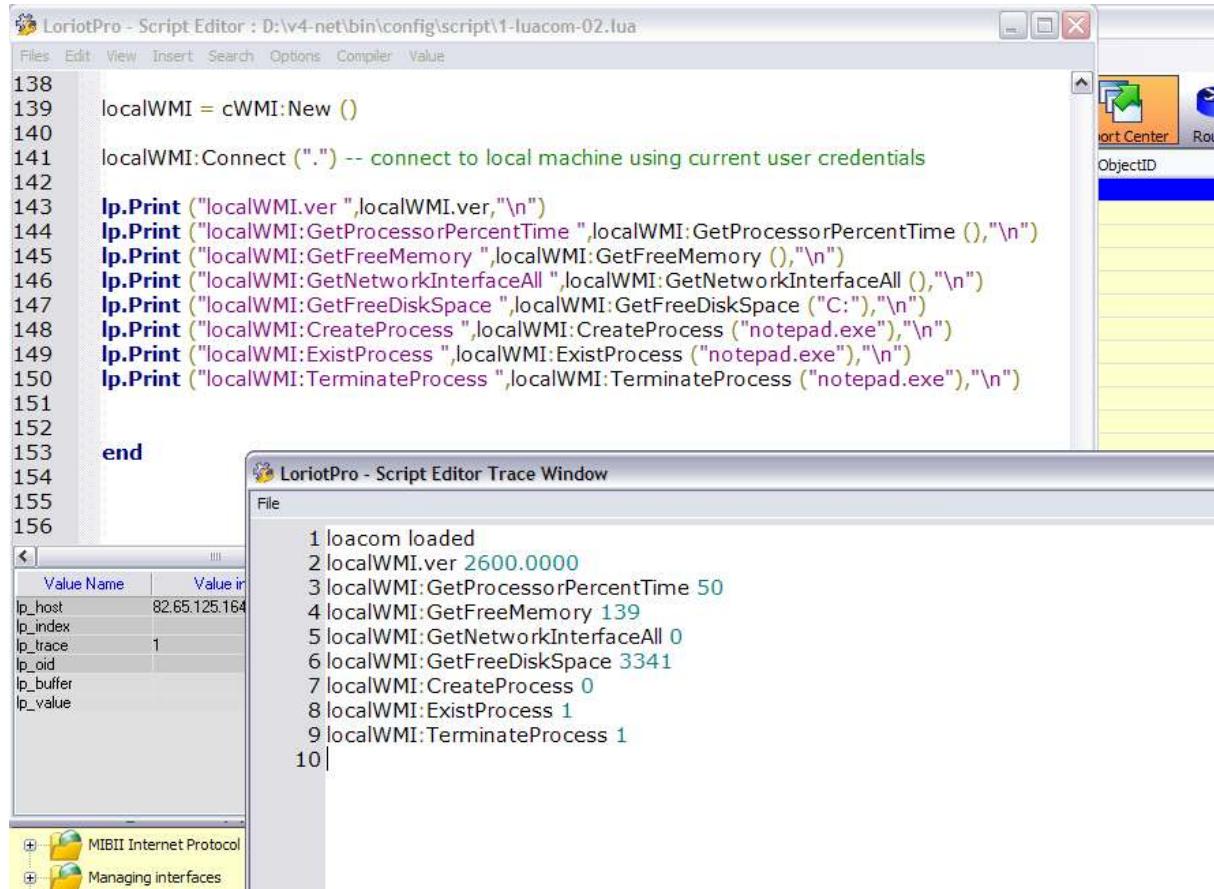
ExistProcess = function (self,Process)
    local colProcesses = self.oWMIService:ExecQuery("select * from Win32_Process
where Name=\\\"..Process..\\\"",nil,48)
    local i
    for index,item in luacomE.pairs (colProcesses) do
        i = (i or 0) + 1
    end
    return i
end
}

localWMI = cWMI:New ()

localWMI:Connect (".") -- connect to local machine using current user credentials

lp.Print ("localWMI.ver ",localWMI.ver,"\\n")
lp.Print ("localWMI:GetProcessorPercentTime ",localWMI:GetProcessorPercentTime
(),"\\n")
lp.Print ("localWMI:GetFreeMemory ",localWMI:GetFreeMemory (),"\n")
lp.Print ("localWMI:GetNetworkInterfaceAll ",localWMI:GetNetworkInterfaceAll (),"\n")
lp.Print ("localWMI:GetFreeDiskSpace ",localWMI:GetFreeDiskSpace ("C:"),"\n")
lp.Print ("localWMI>CreateProcess ",localWMI>CreateProcess ("notepad.exe"),"\n")
lp.Print ("localWMI:ExistProcess ",localWMI:ExistProcess ("notepad.exe"),"\n")
lp.Print ("localWMI:TerminateProcess ",localWMI:TerminateProcess
("notepad.exe"),"\n")

end
```



Exemple d'ouverture de word :

```

local lib,init= lp.LoadLibrary(lp.GetPath().."/lua_com.dll","luaopen_luacom")

if lib then init();
lp.Print("loacom loaded\n");
require("luacom")
word = luacom.CreateObject("Word.Application")

assert(word)

word.Visible = true

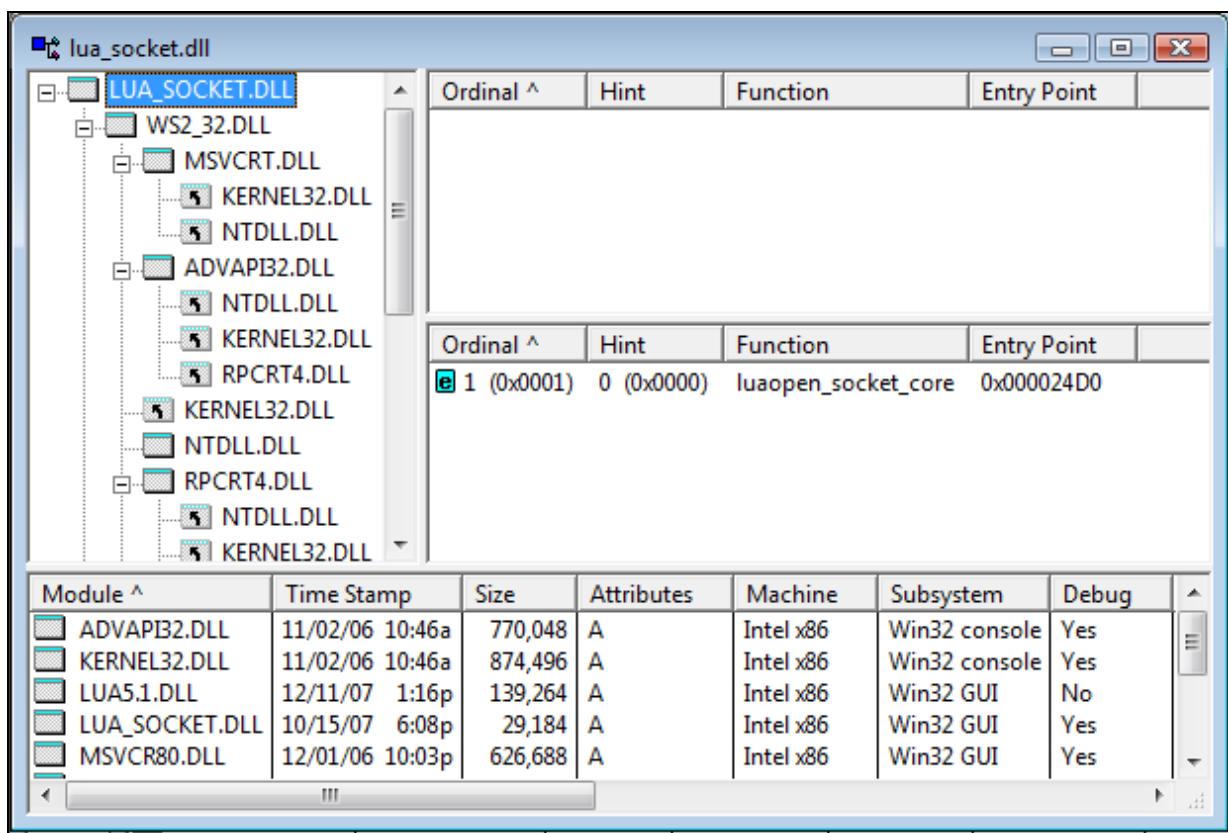
end

```

## 20.4 Exemple LUA\_SOCKET

Cette librairie (<http://www.cs.princeton.edu/~diego/professional/luasocket/>), un peu complexe, peut être utile. L'exemple suivant envoie un paquet syslog sur le port 516 à LoriotPro.

Le fichier ***luasocket-2.0.2-lua-5.1.2-Win32-vc8.zip*** contient la dll ***core.dll*** que nous vous recommandons de renommer en ***lua\_socket.dll*** et de copier dans le répertoire bin du logiciel.



La fonction d'initialisation du module est luaopen\_socket\_core.

Exemple :

```
-- main programme

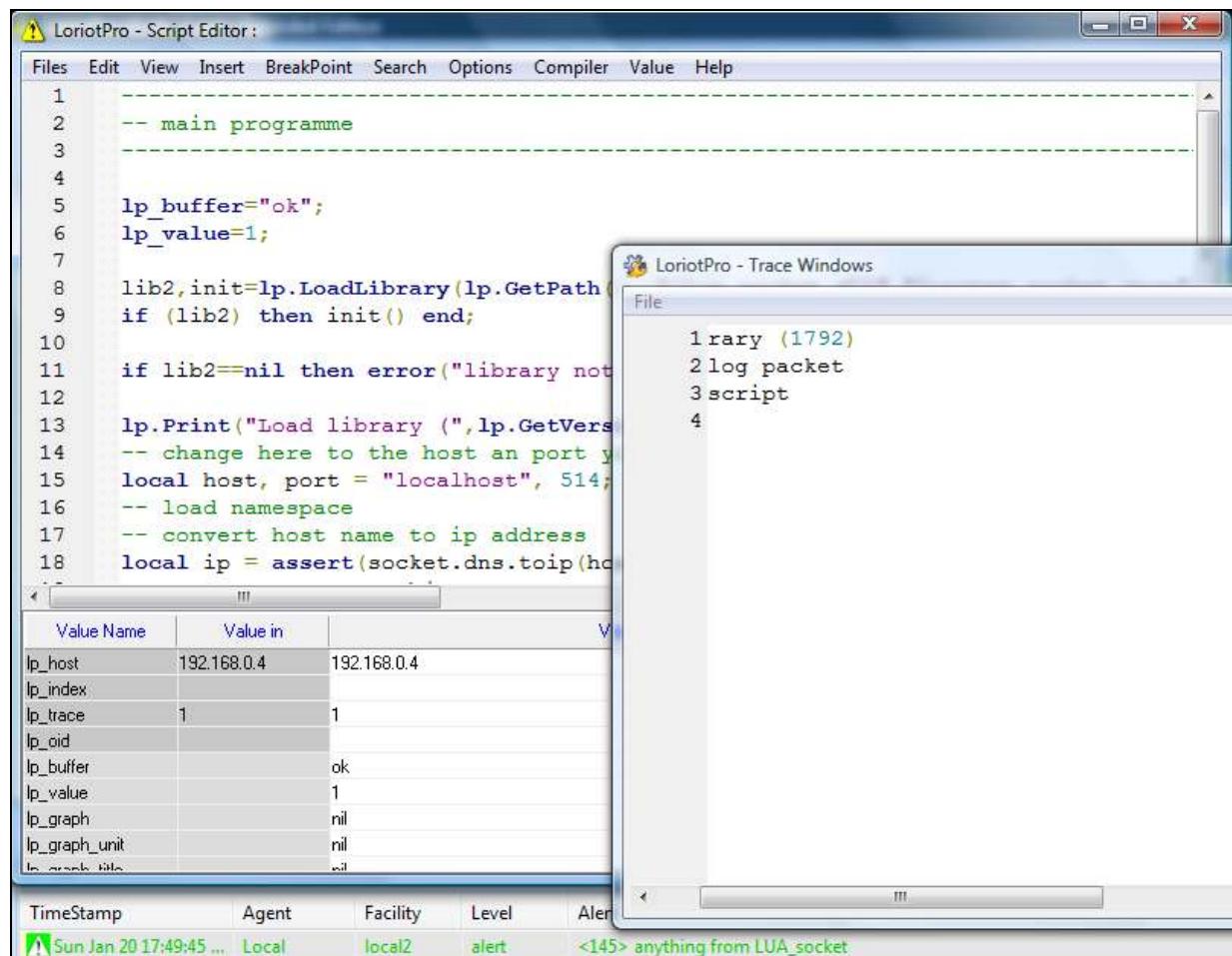
lp_buffer="ok";
lp_value=1;

lib2,init=lp.LoadLibrary(lp.GetPath().."/lua_socket.dll","luaopen_socket_core");
if (lib2) then init() end;

if lib2==nil then error("library not loaded \n") end;

lp.Print("Load library (",lp.GetVersion(),")\n");
-- change here to the host an port you want to contact
local host, port = "localhost", 514;
-- load namespace
-- convert host name to ip address
local ip = assert(socket.dns.toip(host))
-- create a new UDP object
local udp = assert(socket.udp())
-- contact daytime host
assert(udp:sendto("<145> anything from LUA_socket", ip, port))
-- retrieve the answer and print results
--lp.Print(assert(udp:receive()))
lp.Print("send syslog packet \n");

lp.Print(" End of script\n");
```



## 20.5 Projet Lua\_Wizard\_80

Le package Extended Edition contient un exemple et un projet Visual Studio 2005 pour vous aider dans la création de vos propres librairies LUA.

**lua\_wizard.dll**

```
// Lua_wizard.cpp : Defines the entry point for the DLL application.
//

#define WIN32_LEAN_AND_MEAN           // Exclude rarely-used stuff from Windows headers
#include <windows.h>

extern "C" {
    #include "../lib/lua.h"
    #include "../lib/lualib.h"
    #include "../lib/lauxlib.h"
    // #include "../lib/compat-5.1.h"
}

#define LUA_WIZARD_LIBNAME    "lpwz"

/* Pop-up a Windows message box with your choice of message and caption */
int lua_MsgBox(lua_State* L)
{
    const char* message = luaL_checkstring(L, 1);
    const char* caption = luaL_optstring(L, 2, "");
    int result = MessageBox(NULL, message, caption, MB_OK);
    lua_pushnumber(L, result);
    return 1;
}
```

```
}

static luaL_reg wizard_func[] = {
    {"MsgBox",        lua_MsgBox},
    {NULL,           NULL}
};

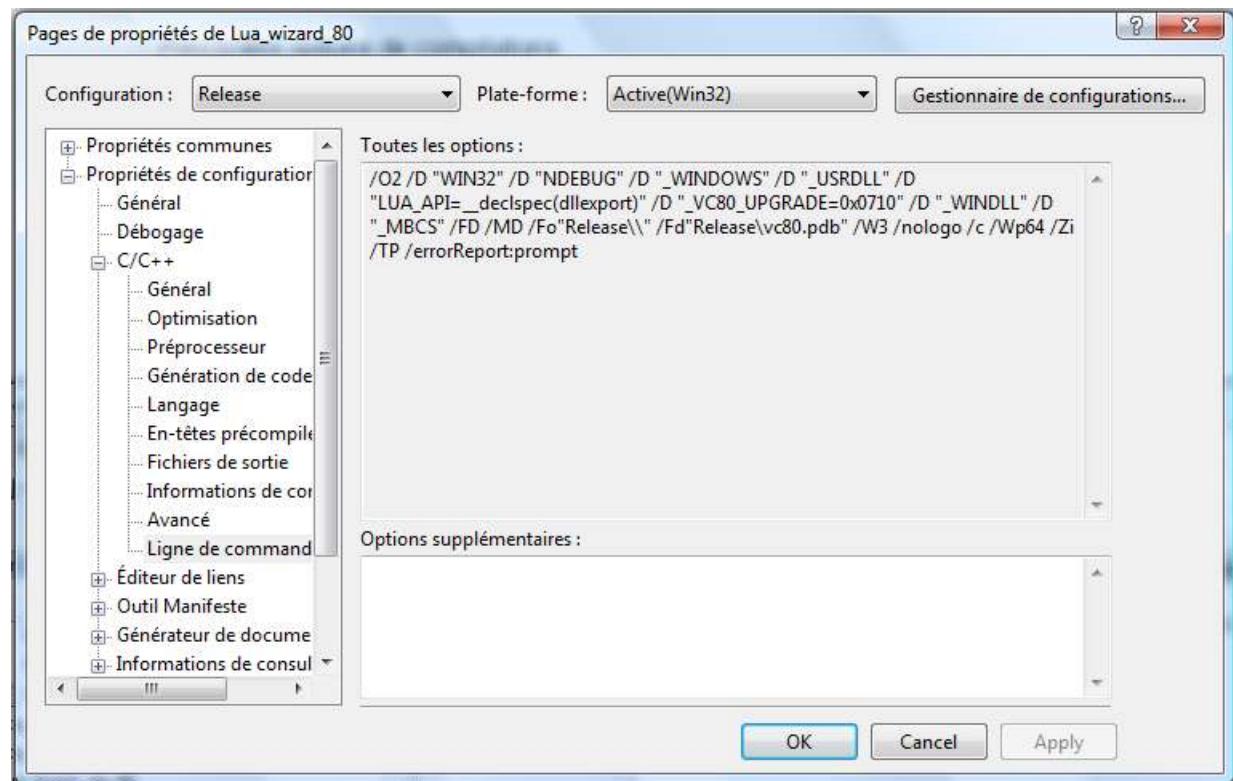
int libinit(lua_State* L)
{
    /* export functions (and leave namespace table on top of stack) */
    luaL_register(L, LUA_WIZARD_LIBNAME, wizard_func);
    return 0;
}
```

### lua\_wizard.def

```
LIBRARY      "Lua_WIZARD_LIB"
DESCRIPTION   'LoriotPro LUA 5.0.2 Plug-in MFC Windows Dynamic Link Library'

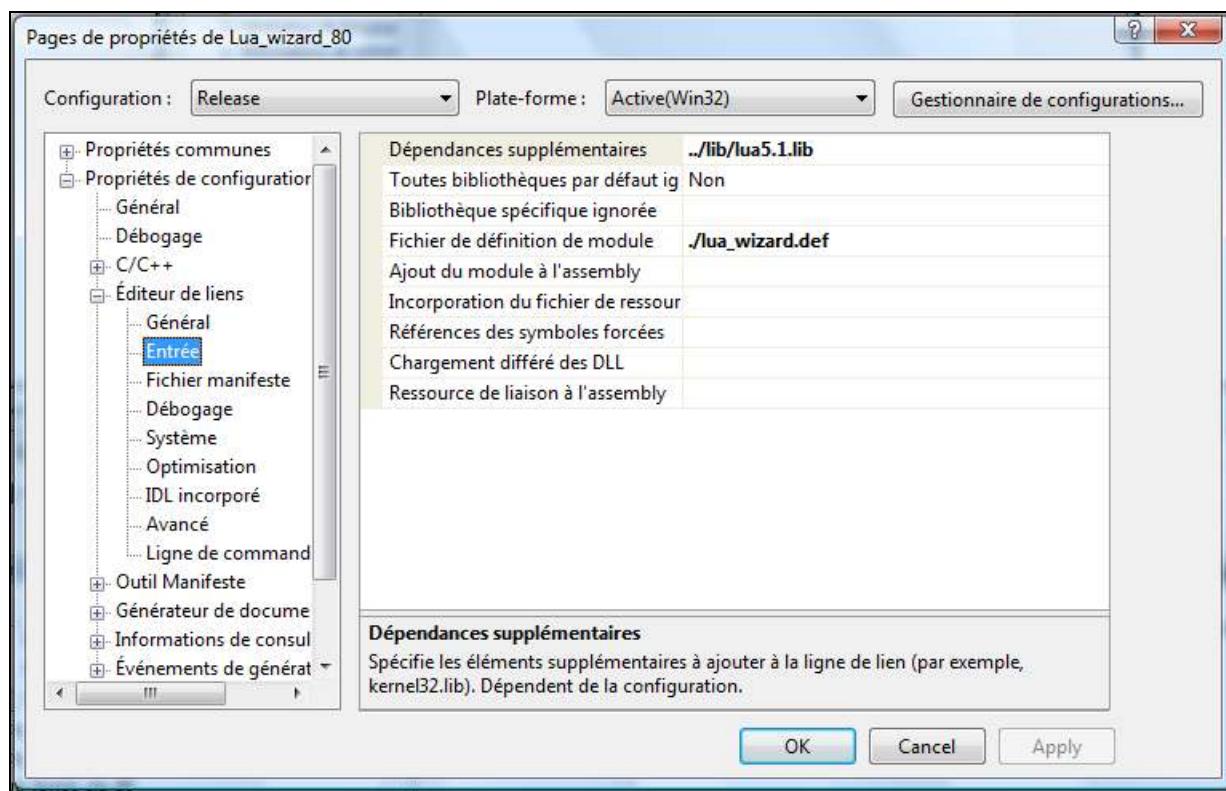
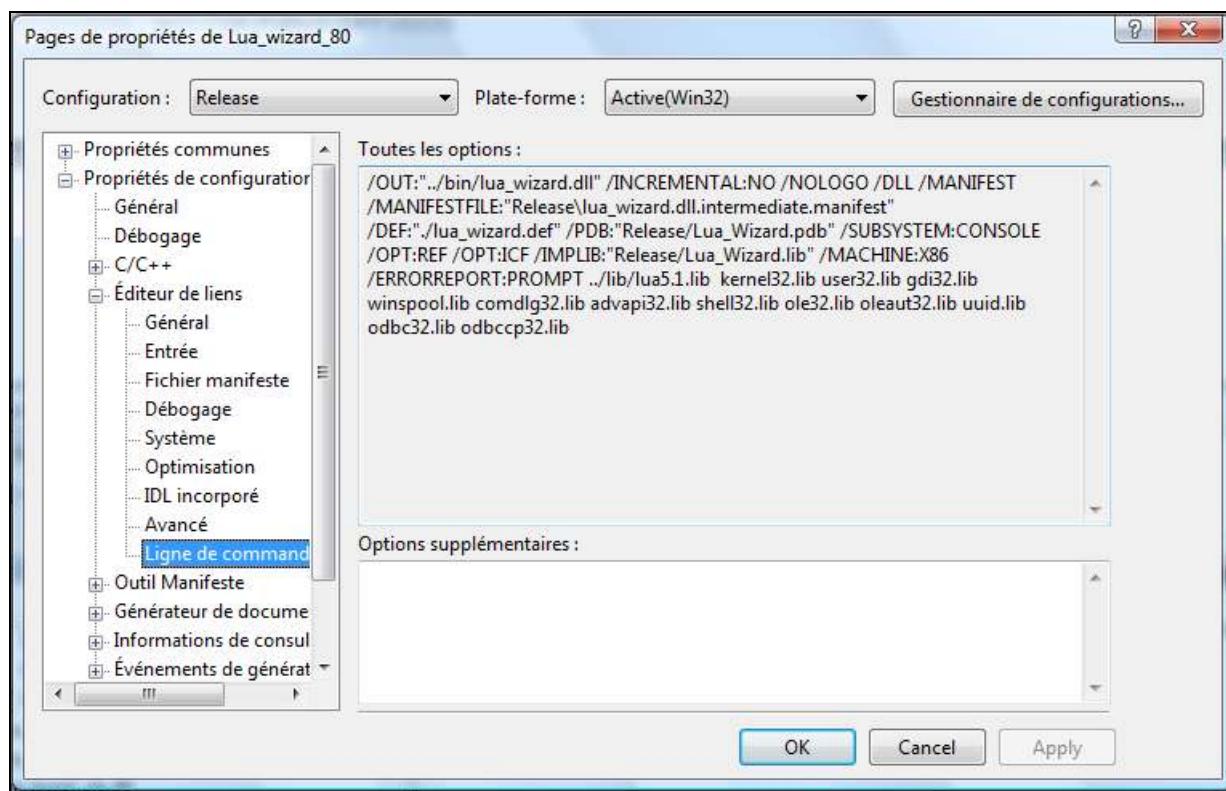
EXPORTS
libinit
```

### Principales options de compilation



# LoriotPro V5.00 Extended Edition – Supervision et gestion SNMP par scripts LUA

## Introduction au System Expert et aux scripts



## 21 Matrice de compatibilité des librairies LUA de Loriotpro

Certaines fonctions se prêtent difficilement à certains environnements. Ce tableau montre les restrictions d'usage. Il est possible de passer outre mais le résultat n'est pas assuré.

environnement	Object Snmp Virtuel	ActiveView Object graphique	ActiveView Mode direct (menu) Modale	Mode direct Menu principal Menu contextuel Reports Center Shortcut
Library				
Ip Librairie noyau	Oui fonctions non souhaitables Ip.LoadLibrary Ip.FreeLibrary	Oui fonctions non souhaitables Ip.LoadLibrary Ip.FreeLibrary	Oui	Oui
Ipav Librairie spécialisée pour les objets graphiques des ActiveViews	Non	Oui	Oui Un objet graphique doit être sélectionné.	Non
Lpw (lua_lp_wizard.dll) Librairie spécialisée pour lancer des wizards de sélection.	Non	Non	Oui	Oui
Librairie externe Lua_odbc Lua_socket Lua_ldap Etc...	Non	Non	Oui Si supporté	Oui Si supporté
Interface COM	Non	Non	Oui	Oui



## 21.1 License

Lua is free software: it can be used for both academic and commercial purposes at absolutely no cost. There are no royalties or GNU-like "copyleft" restrictions. Lua qualifies as [Open Source](#) software. Its licenses are compatible with [GPL](#). Lua is not in the public domain and [PUC-Rio](#) keeps its copyright. The legal details are below.

### [License for Lua 5.0 and future versions:](#)

Starting with Lua 5.0, Lua is licensed under the terms of the [MIT license](#) reproduced [below](#).

### [License for Lua 4.0 and previous versions:](#)

Until Lua 4.0, Lua used its [own license](#), which was very close to the [zlib license](#) and others, but not quite the same.

The spirit of both licenses is that you are free to use Lua for any purpose at no cost without having to ask us. The only requirement is that if you do use Lua, then you should give us credit by including the appropriate copyright notice somewhere in your product or its documentation. A nice, but optional, way to give us further credit is to include a [Lua logo](#) in a web page for your product.

The Lua language is entirely designed and implemented by a [team](#) at [PUC-Rio](#) in Brazil. The implementation is not derived from licensed software.

---

## 21.2 Lua 5.0 license

Copyright © 1994-2005 Lua.org, PUC-Rio.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND

NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

---

### **21.3 Lua 4.0 license**

Copyright © 1994-2002 Tecgraf, PUC-Rio.

Permission is hereby granted, without written agreement and without license or royalty fees, to use, copy, modify, translate, and distribute this software and its documentation (hereby called the "package") for any purpose, including commercial applications, subject to the following conditions:

- The above copyright notice and this permission notice shall appear in all copies or substantial portions of this package.
- The origin of this package must not be misrepresented; you must not claim that you wrote the original package. If you use this package in a product, an acknowledgment in the product documentation would be greatly appreciated (but it is not required).
- Altered source versions must be plainly marked as such, and must not be misrepresented as being the original package.

The authors specifically disclaim any warranties, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. The package provided hereunder is on an "as is" basis, and the authors have no obligation to provide maintenance, support, updates, enhancements, or modifications. In no event shall Tecgraf, PUC-Rio, or the authors be held liable to any party for direct, indirect, special, incidental, or consequential damages arising out of the use of this package and its documentation.

---

Last update: Tue Aug 23 14:02:10 BRST 2005

### **21.4 LoriotPro End User License Agreement**

\* LUTEUS designates the author of the software or its representative and/or its wholly owned subsidiaries, or its suppliers.

IMPORTANT : THIS SOFTWARE END USER LICENSE AGREEMENT ("EULA") IS A LEGAL AGREEMENT BETWEEN YOU AND LUTEUS. READ IT CAREFULLY BEFORE COMPLETING THE INSTALLATION PROCESS AND USING THE SOFTWARE. IT PROVIDES A LICENSE TO USE THE SOFTWARE AND CONTAINS WARRANTY INFORMATION AND LIABILITY DISCLAIMERS. BY INSTALLING AND USING THE SOFTWARE, YOU ARE CONFIRMING YOUR ACCEPTANCE OF THE SOFTWARE AND AGREEING TO BECOME BOUND BY THE TERMS OF THIS AGREEMENT. IF YOU DO NOT AGREE TO BE BOUND BY

---

THESE TERMS, THEN SELECT THE "CANCEL", "EXIT", "QUIT"  
OR "DO NOT ACCEPT THE LICENSE AND QUIT" BUTTON, DO NOT  
INSTALL THE SOFTWARE.

BY CLICKING ON THE "ACCEPT THE LICENSE" BUTTON YOU  
ACKNOWLEDGE THAT YOU HAVE READ THIS AGREEMENT,  
UNDERSTAND IT AND AGREE TO BE BOUND BY ITS TERMS AND  
CONDITIONS.

The software that includes binary, source code, sample programs, libraries and documentation (hereinafter the "Software") is licensed, not sold, to you by Luteus. ("Luteus") or its distributors for use only under the terms of this License, and Luteus reserves any rights not expressly granted to you. You own the media on which the Software is recorded or fixed, but Luteus retains ownership of the Software itself. All use of the Software is governed by the terms of the End User License Agreement which is provided below ("License Agreement"). The Software is solely for use by end users according to the terms of the License Agreement. Any use, reproduction or redistribution of the Software not in accordance with the terms of the License Agreement is expressly prohibited.

## 1. DEFINITIONS

- \* "Commercial Software" refers to the Software that is registered for using in commercial applications.
- \* "Educational Software" refers to the Software that is registered for using for educational purposes. Educational Software is identified, for use by students and faculty of educational institutions, only. Education Software may not be used for, or distributed to any party for, any commercial purpose.
- \* "Evaluation Software" refers to the Software that is auto registered for using during one period of 60 days, for evaluation only.

## 2. LICENSE

This EULA allows you to:

(a) Install and use the Commercial Software on a single computer; OR install and store the Software on a storage device, such as a network server, used only to install the Software on your other computers over an internal network, provided you have a license for each separate computer on which the Software is installed and run. A license for the Software may not be shared or used concurrently on different computers.

(b) Educational Software Only. If you have attained a license for the Education Version of the Software, then you may install and store the Software on a storage device, such as a network server, used only to install the Software on your other computers over an internal network for use by a total number of concurrent users not to exceed the number of user licenses you have been granted and provided the Software is installed and run from the computer. You agree to implement reasonable controls to ensure that your use of the Software does not exceeds the number of licenses you have been granted. You agree that Luteus may audit your use of the Software for compliance with the EULA at any time, upon reasonable notice.

- (c) Evaluation Software Only. If you have installed the Software for a 60 day evaluation period, you agree to implement reasonable controls to ensure that your use of the Software does not exceed the 60 day evaluation period time granted to you. You agree that Luteus may audit your use of the Software for compliance with the EULA at any time, upon reasonable notice.
- (d) Make one copy of the Software in machine-readable form solely for backup purposes. You must reproduce on any such copy all copyright notices and any other proprietary legends on the original copy of the Software.
- (e) Make copy of the Evaluation Software. You must reproduce on any such copy all copyright notices and any other proprietary legends on the original copy of the Software.

Under the terms of this license you may :

#### 2.1. Use the software.

- (a) The Software is "in use" when you run binary, executable, or plugin modules.
- (b) Develop any source code depended on the Software using the LoriotPro SDK and link your projects named plugin to the binary of the Software.
- (c) Use all the plugin sample code source to Develop new plugin for one LoriotPro usage.
- (c) Exploit the result of the work generated by the different module of the Software like plugin (created by you), graphs, HTML pages, reports etc...

#### 2.2. License Restrictions

- (a) Other than as set for Evaluation Software in Section 1, you may not make or distribute copies of the Software, or electronically transfer the Software from one computer to another or over a network.
- (b) Subject to the Grant of License here in above, you may not, in whole or in part, copy, photocopy, reproduce, translate, reverse engineer, derive source code, modify, disassemble, decompile, create derivative works based on the software (exclude the plugin sample code), or remove any proprietary notices or labels on the software without the prior consent, in writing, of Luteus.
- (c) You may not use the dsnmp\_\*.dll files, libraries or/and plugins in another Software without a written permission from Luteus.
- (d) You may not rent, lease, or sub-license the Software and transfer the Registered Software and related materials to other parties without the written permission of Luteus.
- (e) You may not distribute or publish any part of the Software (original or modified) unless you have received a written permission from Luteus.
- (f) You may not redistribute the Software as a part of a software development library without a written permission from Luteus.
- (g) You may not pass to any user registration number, which you have received from Luteus.
- (h) In the event that you fail to comply with this EULA, Luteus may terminate the license and you must destroy all copies of the Software.

##### 2.2.1. Restricted use

In all case LoriotPro is not designed to manage, supervise, control, instruments and/or equipment (like probes...) that have one report with human life.

No restricted samples :

- (a) LoriotPro is not designed or intended for use in on-line control of aircraft, air traffic, aircraft navigation or aircraft communication.
- (b) LoriotPro is not designed or intended for use in construction, operation or maintenance of any nuclear facility.
- (c) In all case, LoriotPro is not designed or intended for use with equipment that have one direct report with human life, like hospital equipment control, some probe, aircraft, nuclear facility, maritime control, military process, certain industrial process etc...

The usual no restricted function of LoriotPro is to help you to supervise your routers and networks through different polling process.

### 3. COPYRIGHT

The Software is owned by Luteus and is protected by International copyright laws and treaty provisions.

Some part of the software use public domain or commercial libraries. See the credit dialogue box includes in the software for more information. From the main menu call : 'menu>help>credits'. All the MIB files or derived works are Copyrighted by their author, see the MIB files headers for more information.

### 4. UPGRADES

If this copy of the Software is an upgrade from an earlier version of the Software, it is provided to you on a license exchange basis. You agree by your installation and use of this copy of the Software to voluntarily terminate your earlier EULA and that you will not continue to use the earlier version of the Software or transfer it to another person or entity. You may not pass to any user old registration number, which you have received from Luteus for the old version.

### 5. LIMITED WARRANTY/LIMITATION OF LIABILITY.

This Software is licensed "AS IS". If for any reason you are dissatisfied with the Software, remove it from your computer system and destroy all copies of it. These warranties are in lieu of any other warranties, expressed or implied, including the implied warranties of merchantability and fitness for a particular purpose. In no event will Luteus be liable to you for damages, including any loss of profits, lost savings, or other incidental or consequential damages arising out of your use of or inability to use the Software, even if Luteus has been advised of the possibility of such damages.

LUTEUS IS NOT OBLIGATED TO PROVIDE ANY UPDATES TO  
THE SOFTWARE.

### 6. INDEMNIFICATION.

By installing and using the Software, you hereby agree to save and hold harmless Luteus from any loss, direct or consequential damage, or claim incurred by you resulting from reliance upon the results obtained through the use of the Software.

The entire risk arising out of use or performance of the Software and Manual(s) remains with the User, however Luteus warrants up to and including thirty (30) days from the date

of your purchase of the Software that the media containing the Software shall be free from defects in material and workmanship. In the event that the media proves to be defective during that time period, and upon presentation to Luteus of proof of purchase of the defective Software, Luteus will at its option :

- 1) provide you with a product of equal or lesser value, or
- 2) refund your money.

Some country do not allow the exclusion or limitation of implied warranties or liability for incidental damages, so the above limitations may not apply to you.

Should you have any questions concerning this Agreement, Or if you desire to contact Luteus for any reason, please contact your local Luteus subsidiary or write to:

<http://www.loriotpro.com>

Copyright © 1999-2004 Luteus SARL, All rights reserved.

Contact : [sales@loriotpro.com](mailto:sales@loriotpro.com)

Luteus SARL  
462 rue Benjamin Delessert BP 83,  
77554 MOISSY CRAMAYEL, FRANCE.

